SQLBOOSTER: A TECHNOLOGY-ENHANCED LEARNING APPROACH FOR STUDENT MONITORING AND ADAPTIVE SQL LEARNING

R.W.A.D.U. Rajapakshe

2025

SQLBOOSTER: A TECHNOLOGY-ENHANCED LEARNING APPROACH FOR STUDENT MONITORING AND ADAPTIVE SQL LEARNING

R.W.A.D.U. Rajapakshe

2025

SQLBooster: A Technology-Enhanced Learning Approach for Student Monitoring and Adaptive SQL Learning

A dissertation submitted for the Degree of Master of Computer Science

R.W.A.D.U. Rajapakshe
University of Colombo School of Computing
2025

DECLARATION

Name of the student: R.W.A.D.U.Rajapakshe
Registration number: 2022/MCS/046
Name of the Degree Programme: Master of Computer Science
Project/Thesis title: SQLBooster: A Technology-Enhanced Learning Approach for Student Monitoring and Adaptive SQL Learning

1. The project/thesis is my original work and has not been submitted previously for a degree at this or any other University/Institute. To the best of my knowledge, it does not contain any material published or written by another person, except as acknowledged in the text.
2. I understand what plagiarism is, the various types of plagiarism, how to avoid it, what my resources are, who can help me if I am unsure about a research or plagiarism issue, as well as what the consequences are at University of Colombo School of Computing (UCSC) for plagiarism.
3. I understand that ignorance is not an excuse for plagiarism and that I am responsible for clarifying, asking questions and utilizing all available resources in order to educate myself and prevent myself from plagiarizing.
4. I am also aware of the dangers of using online plagiarism checkers and sites that offer essays for sale. I understand that if I use these resources, I am solely responsible for the consequences of my actions.
5. I assure that any work I submit with my name on it will reflect my own ideas and effort. I will properly cite all material that is not my own.
6. I understand that there is no acceptable excuse for committing plagiarism and that doing so is a violation of the Student Code of Conduct.

Signature of the Student	**Date (DD/MM/YYYY)**
Rajapakshe	30.06.2025

Certified by Supervisor(s)

This is to certify that this project/thesis is based on the work of the above-mentioned student under my/our supervision. The thesis has been prepared according to the format stipulated and is of an acceptable standard.

	Supervisor 1	**Supervisor 2**	**Supervisor 3**
Name	Mr. Rangana Jayashanka		
Signature	R. Jayashanka		
Date	30.06.2025		

I would like to dedicate this thesis to my family, whose unwavering love, sacrifices, and belief in me have shaped my journey.

To my supervisor, Mr. Rangana Jayashanka, for his invaluable guidance and support.

To my friends, for their constant encouragement and the emotional support provided along the way.

ACKNOWLEDGEMENTS

First, I would like to extend my deepest gratitude to my supervisor, Mr, Rangana Jayashanka, for always providing his guidance, patience, and unwavering support throughout this research. His commitment to academic excellence and diligent attention to detail have shaped this thesis in a significant manner. I would also like to thank the faculty and the staff members of UCSC for providing an environment that encouraged me towards learning and innovation.

My appreciation also goes to my colleagues and friends for providing both distractions when needed and encouragement when it seemed impossible to continue. I am also very grateful to everyone who participated in my surveys and evaluations.

Finally, I should also express my heartfelt thank you to my family for providing me with endless love, understanding, and support throughout my studies. Their belief in me has been a constant source of strength and motivation for me. It might not have been possible without you. Thank you for all that you have done.

ABSTRACT

This thesis presents SQLBooster which is a web-based learning management tool, that is designed to enhance SQL proficiency among the undergraduate students. The system addresses the limitations of traditional e-learning platforms, which often lack personalization and fail to monitor student engagement or psychological state. SQLBooster uses a multi-agent AI architecture, and uses technologies like Large Language Models, Retrieval-Augmented Generation, and FastAPI to enable a dynamic learning environment that is tailored according to the individual needs of the students.

A standout innovation of SQLBooster is its emotion-based tension detection model, which uses the focus details obtained from head-pose detection model and attentiveness details obtained from the attentiveness model to categorize the detected student emotions into four distinct levels of tension. This provides critical insights into the psychological states of the students, and helps the lecturers to identify whether external factors like stress may affect the performance.

SQLBooster also sets itself apart by offering comprehensive personalization features that are not collectively available in the existing tools like adaptive content delivery, personalized and progressive questioning, real-time feedback, and advanced partial grading for both DDL and DML SQL queries. It also provides the lecturers with details such as the common areas of difficulties among the students.

SQLBooster was developed using a pragmatism-guided design science methodology, ensuring both theoretical innovation and practical applicability. The tension detection model demonstrated high accuracy, and chi-squared analysis from pilot evaluations confirmed a statistically significant improvement in students' self-reported SQL confidence. Feedback from users was overwhelmingly positive, highlighting the system's impact on engagement and learning. These findings suggest that SQLBooster effectively supports both learners and educators by combining adaptive learning with real-time affective monitoring.

Keywords: personalization, partial grading, student monitoring, head-pose detection, emotion-based tension detection, attentiveness detection, learning analytics dashboard, feedback, SQLBooster, SQL education, learning management system.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

ABBREVIATIONS

Abbreviation	Definition
AI	Artificial Intelligence
BiGRU	Bidirectional Gated Recurrent Unit
CNN	Convolutional Neural Network
DCNN	Deep Convolutional Neural Network
DDL	Data Definition Language
DMA	Data Management Agent
DML	Data Manipulation Language
EEG	Electroencephalogram
FA	Factor Analysis
LLM	Large Language Model
LSA	Learning Strategy Agent
MCQ	Multiple Choice Question
ML	Machine Learning
PMA	Progress Management Agent
RAG	Retrieval-Augmented Generation
ReLU	Rectified Linear Unit
SQL	Structured Query Language
SVM	Support Vector Machine
UI	User Interface
VDCNN	Very Deep Convolutional Neural Network
3D	Three Dimensional

CHAPTER 1: INTRODUCTION

1.1. Motivation

1.1.1. Opening Section

Computing degree programs at universities play a vital role in preparing students with essential skills like programming, data analysis, cybersecurity, and Artificial Intelligence (AI). These programs are essential in meeting the increasing global demand for skilled technology professionals. An important subject within these programs is Database Management Systems (DBMS). DBMS teaches the students how to design, implement, and manage the databases that support modern applications and businesses (Saeed, 2015). Key areas covered in DBMS include data modeling, database design, Structured Query Language (SQL), and performance optimization.

Proficiency in SQL is a skill that is highly valued in the job market, especially since it enables database optimization, efficient data retrieval, and improved user experience ("The Importance of Writing Efficient SQL Code," 2023). However, many students find it challenging to master SQL through traditional lecture-based teaching methods, especially because most of the students listen passively without practicing what is taught in the lecture (Aupperlee, 2021). While the students have access to online resources, they often fail to provide a personalized learning experience which is tailored to their individual needs (Ali et al., 2021). Furthermore, the existing tools lack the ability to detect student tension based on emotion, which can affect learning performance. Sometimes, students struggle not because they lack knowledge, but because of stress. Current tools are limited in their ability to help the lecturers identify such issues and decide if a student may need extra support (Firoz, 2023).

This research introduces 'SQLBooster', which is an innovative web-based solution that is designed to enhance SQL learning through personalized learning paths, real-time feedback, partial grading (where students receive marks for partially correct SQL queries, without using an all-or-nothing approach), and student monitoring. The system uses advanced technologies such as Large Language Models (LLMs), Retrieval-Augmented Generation (RAG), LlamaIndex, and deep learning to deliver a highly interactive and adaptive learning experience tailored to the individual student needs. Initially, the students receive a set of general SQL questions, and their responses are analyzed to identify specific areas of weaknesses. Based on this analysis, SQLBooster recommends targeted tutorials, videos, and provides progressively challenging personalized questions. According to the answers the students provide for these questions, detailed and specific feedback is provided, highlighting exactly where errors occur and awarding partial marks for the correct segments of the answers. The tool also includes a chat-based assistant that provides explanations that are syllabus aligned. It can also provide links to additional resources if needed. To support comprehensive learning, SQLBooster integrates models for head pose detection, attentiveness detection, and emotion-based tension detection to monitor student engagement and identify signs of tension. Based on this data, the system can notify lecturers if a student may require additional support. Apart from the student learning analytics dashboard, there is a dedicated analytics dashboard for the lecturers to view common areas of difficulties among students, track student progress, and upload learning materials from which personalized content is generated.

This chapter presents the background and context of the study, followed by the statement of problem, rationale, research aims, objectives, and questions. It also outlines the scope and limitations, significance of the research, and the thesis structure.

1.1.2. Background

SQL, an essential component of computer science education, has evolved significantly since its introduction in the 1970s. It can be considered as a language which is used to manage relational databases and plays a fundamental role in tasks such as data manipulation, query optimization, and schema creation. Thus, it is an important skill needed in the modern computing curriculum ("What is SQL," 2024). By using data-driven technologies to shape modern business practices, SQL supports critical decision making, and thus assists to maintain competitiveness. (Hosen et al., 2024). Therefore, learning SQL is of utmost importance to both IT professionals and computing students.

Over the years, educational institutions have increasingly focused on adopting technology integrated approaches along with the traditional classroom instructions, to enhance the students' comprehension of technical skills such as SQL (Magalingam, 2022). However, most of the earlier attempts included computer-aided instruction systems that were designed to supplement lectures with static exercises and practice questions (Ruliah et al., 2019) (Kulik and Kulik, 1987). These tools laid the foundation for the subsequent developments but were limited in regards to providing interactive and personalized features (Titovskaia et al., 2019).

In the early 2000s, a shift towards more interactive learning environments could be observed, especially with the emergence of e-learning platforms and learning management systems (Khan and Mohamed, 2015). These platforms enabled access to a variety of learning resources beyond the traditional lectures, such as tutorials, discussion forums, interactive quizzes, and automated assessments that provide feedback indicating the correct answers. Nevertheless, despite this progress, recent studies show that these systems still struggle to provide tailored, personalized learning experiences (Permpool et al., 2019). Most of these LMS tools provide features such as standard SQL exercises and automated grading, but lack the adaptive mechanisms that are required to address the individual gaps effectively.

In the recent years, AI and Machine Learning (ML) have introduced opportunities to create advanced adaptive learning tools. Techniques such as LLMs, LlamaIndex, and RAG, can assist in creating systems that are capable of providing more dynamic and responsive learning experiences (Lewis et al., 2020) . As an example, RAG systems can be used to enhance the retrieval of relevant information from large data repositories, enabling responses that are highly contextually relevant and informed (Petroni et al., 2024). LLMs can assist to utilize vast amounts of text data, and this can facilitate the development of more flexible learning interactions. These can adapt in real-time to match each student's learning speed and style. These developments open possibilities to enhance the SQL education by providing more interactive and personalized learning experiences to cater to the specific needs of students.

However, despite the improvements of AI and ML, SQL learning tools remain underdeveloped. Existing tools often focus on automated grading and error detection but have limited ability to provide partial grading or adaptively adjust learning paths. Additionally, they often fail to provide comprehensive real-time feedback that is capable of providing deeper understanding. Furthermore, they are not capable of tracking and analyzing the detailed interactions of the students, and this limits the ability to understand the progress of the students during the SQL activities (Ouared et al., 2023). Moreover, while research like (Brkić et al., 2024) highlights efforts to improve automated programming assessment systems, these tools still struggle with effectively adapting to the complexities of SQL query evaluation. Additionally, most of the current tools do not consider non-academic factors like the stress levels of the students, which can significantly influence their performance. Without insights into such external factors, it becomes challenging to fully understand what affects the student responses during SQL activities. This indicates a critical need for enhanced student monitoring capabilities while students answer allocated questions. Furthermore, there is a need for further development in

SQL learning technologies, especially in areas such as providing personalized learning experiences and real-time feedback to offer continuous support tailored to the needs of the students.

The limitations in current SQL tools highlight the need for a solution that can provide adaptive and personalized learning environments, student monitoring, and the capability to assess students using partial grading. A tool like SQLBooster will address these gaps by helping to clearly identify students' weaknesses. By using techniques such as RAG, LLM, and LlamaIndex, SQLBooster will allow students to practice and deepen their knowledge in areas where improvement is needed. It will also assist them in gaining a better understanding of SQL concepts and in enhancing their skills. Furthermore, by integrating emotion-based tension detection capabilities, it will identify psychological barriers that may hinder student performance and will notify lecturers to provide the students with any support required.

1.2. Statement of the problem

Despite advancements in educational technology, existing SQL learning analytics tools still contain significant gaps in supporting personalized, adaptive, and emotionally responsive learning experiences. Many students struggle to develop SQL proficiency because existing tools do not sufficiently address the individual learning needs or tension they may be feeling. These limitations restrict the effective skill development and may lead to disengagement or underperformance, particularly in students who need targeted support. Addressing these challenges requires a unified system that integrates personalized learning, adaptive feedback, partial grading, and student monitoring.

1.2.1. Current State of Research

SQL learning tools have evolved to include several important features that support self-guided learning. These tools allow the students to practice SQL queries, receive immediate feedback on syntax or logical errors, and benefit from features such as automated grading (Permpool et al., 2019). Many platforms also include interactive quizzes and exercises designed to reinforce foundational SQL skills. These enable students to correct errors without relying entirely on instructor intervention.

More recent advancements in learning analytics have also enabled systems to monitor the student interactions, providing data that can be used to analyze learning behaviors (Ouared et al., 2023). Some monitoring systems in broader educational contexts use facial expression analysis, gesture recognition, and attentiveness tracking to improve academic integrity and detect disengagement (Shi et al., 2017). However, most struggle with high sensitivity and specificity in diverse environments (Hossen and Uddin, 2023; Indi et al., 2021).

1.2.2. Research Gap

While the current SQL learning tools provide necessary features such as automated grading and instant feedback, they lack the flexibility needed to address the individual learning needs of the students. Most systems present the same set of questions and feedback to all students, regardless of their performance level or specific areas of difficulty (Permpool et al., 2019). Even systems that attempt to personalize content often have limitations, such as the inability to progressively adjust the difficulty based on students' weaknesses. This prevents students from progressing at their own pace and deepening their understanding in weaker areas. Although learning analytics solutions as those presented by (Ouared et al., 2023) provide hints and error messages to assist the students in correcting their mistakes, they are not capable of providing advanced feedback necessary for adaptive learning. Furthermore, even though partial grading has been implemented recently, the tools still lack the ability to effectively grade SQL related to Data Definition Language (DDL) and more advanced SQL queries (Joseph et al., 2016).

Beyond academic evaluation, there is also a gap in how these systems address the non-academic factors that influence learning, such as student focus, engagement, and their emotion through student monitoring. While head pose detection, attentiveness tracking, and emotion recognition models do exist, the accuracy of these systems is often affected by environmental conditions, resulting in false positives or negatives in detecting engagement (Rodríguez-Arce et al., 2020). Currently, no SQL learning platform integrates head pose detection, attentiveness detection and emotion recognition to estimate student tension levels. Accurately understanding students' tension levels can provide critical insights into whether additional instructional support may be required.

Addressing these gaps is not only a matter of technological advancement but also of societal importance. SQL is a foundational skill in data-driven industries, and proficiency in it directly influences students' future employability. Without personalized support and emotional awareness, students who face learning barriers may fall behind. A comprehensive, adaptive learning tool that integrates academic assessment with student monitoring is essential to ensure that every learner receives the support they need to succeed.

1.3. Research Aims and Objectives

1.3.1. Research Aims

The aim of this study is to enhance the SQL knowledge of undergraduate students using Technology Enhanced Learning and Learning Analytics. In order to achieve this, a software system called SQLBooster is designed and implemented.

1.3.2. Research Objectives

The first objective of this study is to design and implement a web-based SQL learning environment that enhances the learning experience of undergraduate students through both multiple-choice questions (MCQs) and structured query exercises. The system aims to provide an intuitive and user-friendly interface that encourages active participation and practice.

The next objective is to integrate adaptive learning and personalized feedback mechanisms that dynamically respond to individual student performance. These mechanisms adjust the difficulty of questions in real time, customize content to address individual learning needs, and provide real-time explanations for errors. This includes both encouragement for correct responses and guidance for incorrect ones.

The third objective is to develop an automated grading system that evaluates SQL queries provided by students and provide partial marks for correct portions of their answers, for both Data Manipulation Language (DML) and DDL responses. This approach aims to offer a more supportive evaluation by helping students understand exactly where they made mistakes and how to improve.

The final objective is to monitor the students to ensure academic integrity and the impact of external factors such as stress. To achieve this, the system will integrate models for head pose estimation, attentiveness detection, and emotion-based tension detection.

1.3.3. Research Questions

This study is guided by the following research questions:

RQ 1. How can a web-based, user-friendly system impact the engagement and the SQL learning outcomes of the undergraduate students?

RQ 2. How can the use of adaptive learning paths and personalized feedback improve the proficiency in SQL by addressing the students' skill gaps?

RQ 3. How does automated, partial grading influence the understanding and improvement of the students with regards to both DML and DDL?

RQ 4. How effective are the student monitoring capabilities?

1.4. Significance

This research introduces SQLBooster, a tool designed to improve the engagement and proficiency of students in SQL by providing adaptive learning paths, personalized feedback, partial grading, and student monitoring. These features aim to overcome limitations in the current SQL education by addressing individual difficulties and diverse skill levels.

SQLBooster will use a problem-based learning (PBL) approach to engage the students in active problem-solving. This will help to enhance their understanding and critical thinking skills. Furthermore, the partial grading capabilities of SQLBooster will encourage students to focus on areas needing improvement, supporting a more individualized educational experience.

Proficiency in SQL is directly linked to job market readiness, yet many students struggle with complex concepts due to the lack of targeted guidance in traditional tools (Strahl, 2024). The absence of adaptive support and personalized feedback can result in persistent knowledge gaps, limiting long-term success in technical fields.

In addition to its core features, SQLBooster also include a basic learning analytics dashboard for both the students and the lecturers. This is not a primary focus of the research, but it will provide the students with an overall understanding of their standing. Furthermore, the lecturer dashboard will offer insights into the common areas of difficulty among the students. Ultimately, this can result in better educational outcomes.

The absence of dynamic feedback mechanisms in traditional settings often forces students to rely heavily on lecturer support. However, many students are reluctant to ask for help. This usually results in passive learning without much interaction (Aupperlee, 2021). By offering a more personalized and interactive learning experience, SQLBooster will help to fill these gaps. This will allow the students to develop a strong foundation in SQL.

The impact of SQLBooster extends beyond academia. By aligning with industry expectations, it equips students with robust query-writing skills and practical problem-solving abilities.

Additionally, the student monitoring feature will improve evaluation validity by indicating whether external factors such as tension may have affected the student performance. It also informs lecturers whether students are completing work independently or require support. This dual perspective helps bridge the gap between theoretical learning and practical proficiency.

In summary, SQLBooster contributes to both educational and industry outcomes by supporting practical skill development through personalized, feedback-rich, and emotionally aware learning tools.

1.5. Scope

1.5.1. Includes:

- SQLBooster is designed for the undergraduate students in Sri Lankan universities, and it will focus on enhancing their SQL learning through personalized learning, real-time feedback, monitoring and automated grading.
- The study will be limited to relational databases only, with SQL as the primary language.
- The learning experience will be dynamically adjusted based on the performance of each student by using LlamaIndex and OpenAI techniques.
- A combination of three models for head pose detection, attentiveness detection, and emotion-based tension detection is used to determine student tension level and determine whether additional support may be needed.
- This will contain a dashboard component to clearly provide student engagement metrics and to show insights about the performance and weaknesses of the students. This feature will be added to improve user experience but will not be a research component.

1.5.2. Excludes:

- The tool will not cover advanced SQL topics typically covered at the master's level or beyond.
- SQLBooster will not have multilingual support.
- SQLBooster will not allow the students to execute their queries and observe the results.
- SQLBooster will not include learning features such as shared workspaces or peer learning.
- Due to the project being developed within a limited timeframe and under strict budgetary constraints, it will be challenging to fully test and distribute the model.

1.6. Structure of the Thesis

This section intends to provide a roadmap of the thesis, briefly outlining the context and the purpose of each chapter to guide the reader.

Chapter Two: Literature Review presents a comprehensive review of relevant literature. It focuses on both theoretical and practical aspects of SQL education and highlights the key challenges in current SQL learning methods. Additionally, it will set the foundation needed to design SQLBooster.

Chapter Three: Methodology details the approach used to develop and implement SQLBooster. It outlines the design and development stages, and describes the technologies, techniques, and tools used to achieve the features such as adaptivity, feedback mechanisms, student monitoring, and personalized question pathways within the tool SQLBooster.

Chapter Four: Evaluation and Results discusses the evaluation process and the outcomes of implementing SQLBooster. It focuses on the impact of SQLBooster on the proficiency and engagement on SQL of the students. It will provide details about quantitative and qualitative data analysis, and will discuss how the tool can cause improvements in the student performance.

Chapter Five: Conclusion and Future Work summarizes the key findings of the research and reflects on the limitations of the current study. It also proposes directions for future enhancements and extensions of SQLBooster.

CHAPTER 2: LITERATURE REVIEW

2.1. Introduction

The aim of this literature review chapter is to provide an in-depth and comprehensive review of the existing literature that forms the basis for the research on ‘SQLBooster’, a tool that helps to improve SQL learning through adaptive and personalized methods. This will examine the advancement and limitations of the current SQL learning tools and methodologies, and will analyze the current status of automated grading, real-time feedback, student monitoring systems, and personalized learning paths, as well as how these can be achieved. This seeks to identify and address the educational challenges faced by the students when learning SQL.

The main purpose of this chapter is to provide a detailed analysis about the past and the current literature and highlight the significance of SQLBooster by presenting the gaps in the existing SQL educational tools. This will assist to identify both the theoretical foundations and empirical research related to SQL education, adaptive learning technologies, and feedback mechanisms.

The scope of the literature review consists of the evolution of SQL learning management tools from the foundational interactive SQL practice platforms to modern adaptive and personalized learning systems. This review will explore significant themes like automated grading systems, real-time feedback mechanisms, student monitoring, and adaptive learning paths. Furthermore, this will contain details regarding the learning analytics dashboards and systems that is capable of providing structured feedback designed to boost student engagement and performance. However, more advanced topics like procedural SQL extensions and multilingual support will not be considered in this analysis since it does not align with the main objectives of this research.

The body of this chapter will be organized in a thematic manner, and will be followed by a general chronological order to illustrate how educational technologies have evolved with time. Each of these themes will address the key developments, limitations, and potential advancements that can support the reasonings behind the SQLBooster project. Finally, this chapter will conclude with a summary of the important findings and by stating how SQLBooster can address the existing gaps in the tools.

2.2. Body

2.2.1. The Foundation of Theory

Adaptive and personalized SQL learning tools are built on various educational theories that emphasize individualized learning, timely feedback, and the use of interactive dashboards for improved student engagement. Here, various themes like student monitoring, adaptive learning systems, feedback mechanisms, automated grading, and visual learning dashboards will be explored.

Personalized and Adaptive Learning Systems:

Adaptive and personalized learning mainly focuses to tailor the educational experiences to the unique needs of the individual students (Taylor et al., 2021). These systems are capable of evaluating the initial knowledge levels of the students and can dynamically adjust the content with the progress. This helps to optimize the student engagement and the mastery of the complex topics (Daugherty et al., 2022). The theory focuses on the way of using data to create individualized learning paths, that is capable of responding to the student inputs and performance in real-time.

Over the years, these systems have evolved by transitioning from basic, non-personalized systems to more complex and adaptive solutions that are tailored according to the individual student needs (Nalintippayawong et al., 2017). With the advancement of educational technology, this progression highlights the importance of using data analytics and deep learning to build learning management tools that focus on personalized learning, since it helps the students stay engaged and learn more effectively through adaptable educational tools.

Feedback Mechanisms:

Feedback plays an important role in learning management systems because it helps to evaluate and guide the students as they learn. (Gupta, 2024) emphasizes that well-structured feedback can support self-directed learning by providing insights into the performance gaps and by suggesting improvements. (Nakamoto et al., 2023) states that timely feedback can engage the learners, improve their confidence, and can facilitate continuous improvement by guiding them with explanations and corrective actions. Furthermore, (Todorova, 2022) explains that personalized, real-time feedback can improve the student motivation and engagement. Thus, feedback helps the students to improve by helping them understand why mistakes occurred and how to improve.

As explained by (Reilly, 2018), personalized, explanatory feedback can help the students to self-regulate and adapt their learning strategies. This theoretical approach highlights that the feedback mechanisms should improve to provide real-time, personalized responses to meet individual student needs effectively.

Automated Grading:

Automated grading is an important aspect that makes the educational tools more efficient. It can allow fair and consistent grading and evaluation in large learning environments where manual grading would be too time-consuming or difficult (Venkatamuniyappa, 2018). Usually, algorithms are used to check the SQL answers provided by the students against the correct results and this helps to ensure that all the students are evaluated according to the same objective criteria. Since this allows to provide immediate feedback, the students can quickly identify and correct their mistakes (Matthews et al., 2012).

Partial grading is a key theoretical concept that can play a major role in areas where the responses of the students can contain elements of both correctness and error. This helps to identify the portions of the answer that are correct, even if the entire answer is not accurate. This can help students to understand their strengths and areas for improvement (Chandra et al., 2019). In the context of SQL learning, where complex queries can contain correct and incorrect elements in the same response, partial grading can help the students to truly understand the areas they should focus on more.

Student Monitoring:

Student monitoring systems play an important role in tailoring the educational experiences to meet the individual needs. According to the theory that direct observation of student behaviors and physiological responses helps to increase the ability to customize the teaching strategies, these systems provide important information that can assist in optimizing the learning outcomes. As an example, head pose estimation technology helps to determine whether the students are focused on their screens and this offers insights into their engagement levels (Wang and Xia, 2021).

The integration of physiological monitoring, like emotion, head-pose, and attentiveness detection, can help to further enrich these systems. These can help the instructors by providing insights into areas like the stress levels of the students, and can assist them to make decisions such as whether these factors might have affected the concentration and the academic performance of the students. Furthermore, access to this data can enable timely interventions to help students manage stress and improve their learning capacity (Rodríguez-Arce et al., 2020).

Additionally, the attentiveness monitoring technologies that analyses the eye gaze and facial expressions, can deliver insights into the student engagement. This can assist the lecturers in evaluating the effectiveness of their teaching methods and making necessary adjustments to optimize the learning outcomes (Barba Guamán and Valdiviezo-Diaz, 2022).

In summary, student monitoring systems not only support academic integrity but can also provide a comprehensive view of the emotional and cognitive states of the students. This can assist the lecturers to gain a deeper understanding of the potential factors that might be affecting the student performance, like stress or disengagement. Based on this information, it will be possible for the educators to implement effective strategies that support both the academic success and the mental health of the students.

Learning Dashboards and Visual Analytics:

The theoretical foundation of learning dashboards and visual analytics is mainly based on its ability to convert complex data into information that is easy to understand. Learning dashboards collect data and represent them through interactive visual tools. This helps to improve learning by allowing both the students and instructors to track performance and engagement in a clear and efficient way. (Park and Jo, n.d.) states that dashboards can improve the student motivation, by providing real-time, clear feedback. Visual representation also results in improved comprehension that will in turn help in decision making.

As explained in (Ramaswami et al., 2023), the presence of learning dashboards can result in active learning behaviors. It increases the engagement due to improved self-awareness. Furthermore, (Wang et al., 2023) describes that the use of prescriptive dashboards can support the varied learner needs because it can offer tailored guidance. Thus, as explained in (Todorova, 2022), by integrating learning dashboards, students can better adapt their strategies, making the learning process more personalized and effective.

2.2.2. The Empirical Research

The empirical research on the development and enhancement of SQL learning tools is broad and there are numerous studies that examine how various methods assist in student learning. This section will analyze the empirical findings related to real-time SQL monitoring, adaptive learning systems, automated grading mechanisms, real-time feedback, and visual analytics.

Personalized and Adaptive Learning Systems:

Empirical research on adaptive and personalized learning systems shows that personalization assists in improving the effectiveness in SQL education. The journey towards personalized learning in SQL education began with basic systems that provided elementary interactivity without personalization. SQLator (Sadiq et al., 2004), is an early example for a web-based interactive learning tool which was designed to allow students to write, execute, and evaluate SQL queries. This platform categorized the questions by difficulty, but was not able to adapt the questions based on the individual student progress or their specific weaknesses. The static nature of question presentation limited personalized learning experiences.

With time, more advanced systems, like the tool described in (Soler et al., 2006) emerged. This provided each student with an interactive set of SQL queries to solve. Students could input their answers and receive corrective feedback and had the opportunity to revise and resubmit their solutions to get updated feedback. This provided a more interactive experience because the students could engage in a problem-solving cycle. However, the questions remained static and this tool was not able to dynamically adjust the question difficulty based on real-time performance.

DBLearn (Nalintippayawong et al., 2017), was able to mark a significant step forward in adaptive learning, especially for database courses. Empirical studies showed that this greatly improved the student engagement by using adaptive learning methods tailored to individual student profiles. The ability of the system to monitor the student performance and adjust content delivery, made it possible to get better learning outcomes. Empirical evaluations highlighted that DBLearn improved student engagement due to its ability to provide a personalized experience. However, this still needed to improve how it predicts what students need to learn next and how it automatically creates questions that matches with the level of each student.

Research highlighted in (Taylor et al., 2021) explains the importance of systems that can analyze the previous student performance to guide the delivery of content. Empirical studies demonstrated that adaptive technologies can improve learning by creating individualized learning paths. This helps to optimize the engagement of students.

(Daugherty et al., 2022) further confirmed that adaptive learning technologies can increase the motivation of the students and that personalized learning can result in improved outcomes. However, empirical observations pointed out that the systems still struggle to provide personalized content that matches the growing skills of the students. These findings support the development focus of SQLBooster, which aims to incorporate adaptive pathways.

Feedback Mechanisms:

Feedback mechanisms play an important role in improving the learning experience of the students by assisting them to recognize their performance gaps and offering guidance for improvement. Initially, the early SQL learning tools provided only binary feedback. As an example, SQLator provided students with immediate feedback explaining whether their SQL query is correct or not (Sadiq et al., 2004). Even though this basic error recognition is important, this form of feedback is not sufficient to obtain the full understanding of the concepts.

Then, SQL learning tools were improved to consist of iterative submission and correction processes. As an example (Soler et al., 2006), represents a web-based tool used for teaching SQL, which allowed the students to iteratively submit answers and receive automated feedback. This supported a cycle of learning and improvement. Here, the feedback was syntax-focused and this was an improvement from the previous tools which only provided binary feedback.

(Reilly, 2018) emphasized the importance of feedback that supports active learning. The study described how providing feedback during the learning sessions was observed to improve the student engagement. This suggested an improvement in the student understanding of the concepts. However, this feedback mechanism was instructor-driven. Thus, it is clear that an automated system that can provide feedback throughout the learning process can assist in improving the learning of the students.

Further improvements could be seen in tools like MySQL Sandbox which was able to provide more structured feedback that was specific to error types, like missing columns or data types (Permpool et al., 2019). However, the feedback provided was not personalized according to the student needs.

(Fujita et al., 2019) provided additional insights into feedback-driven learning. Here, it described a learning support system that logged the student interactions to provide feedback on the common SQL mistakes. This enabled the instructors to monitor and understand the learning patterns in an effective way. However, the system did not offer adaptive feedback which was tailored according to the individual progress of students. Instead, this focused on providing general feedback for common errors.

(Nakamoto et al., 2023) stated that real-time feedback systems can improve learning outcomes by providing immediate, contextually relevant corrections. (Gupta, 2024) also highlighted that personalized feedback can improve the student motivation and engagement. Overall, even though empirical research shows a significant improvement from basic error notifications to more comprehensive feedback, the need for the availability of personalized and adaptive feedback remains as a critical area for development. This underlines the necessity for a tool like SQLBooster, which contains feedback mechanisms that are adaptive and can cater to the individual learning needs of students.

Automated Grading:

The development of automated grading in SQL education reflects a move from simple grading systems to more advanced partial evaluation models. Initially, in tools like SQLator, the grading was binary. It could only indicate whether a query was correct or not. It was not capable of checking for the partial correctness of queries and was primarily focused on SELECT statements.(Sadiq et al., 2004). Similar to above, the tools (Soler et al., 2006) and (Nalintippayawong et al., 2017) were also limited to grading basic SQL statements without the ability to provide partial credit. Thus, even though this provided immediate assessment, it lacked the depth that is required to guide the students in understanding their mistakes comprehensively.

The MySQL Sandbox (Nalintippayawong et al., 2017) significantly improved the scope of grading by supporting a wider range of SQL commands, including both DDL and complex DML statements. This system was capable of providing immediate feedback on executed queries, which extended beyond the simple SELECT statements. However, this tool was also not capable of providing partial correction.

Advancements in partial grading emerged with the model proposed by (Joseph et al., 2016). This approach allowed the system to evaluate partially correct answers and give the students credit for the components of their responses that were correct. This was mainly useful for complex SQL queries which consisted of multiple conditions or joins, where some elements might be correct even if the overall response was not. This allowed the students to learn from their mistakes and improve in their struggling areas.

(Panni and Hoque, 2020) expanded on these advancements by using algorithms to evaluate individual components of SQL queries, like SELECT, FROM, WHERE, and nested queries. Their model assigned scores by comparing the syntactic similarity of answers of the students to a reference query. Empirical results showed that this allowed students to pinpoint the correct and incorrect elements in their answers. (Venkatamuniyappa, 2018) explained that the use of partial grading mechanisms can result in more effective learning because this not only provide marks for the correct points, but will also encourage the students to focus on their weak areas.

It can be observed that automated grading has evolved from evaluating basic SELECT queries to support a wide range of SQL commands. Furthermore, grading has been improved from binary grading to partial grading. However, most of the systems focuses only on the partial grading of DML operations. Despite this limitation, partial grading models have significantly improved the quality of the grading feedback, making it more comprehensive and informative.

These advancements lay the foundation for the goal of SQLBooster, which is to extend the support to partial grading of even the DDL and complex queries.

Student Monitoring:

Student monitoring not only focuses on maintaining academic integrity but also focuses on understanding the emotional and psychological states of the students. Recent studies show that emotional states can impact the learning processes and that they can affect the students' motivation, focus, memory, and the overall academic performance. Areas like emotion detection, head pose detection, and attentiveness detection play a major role in improving the student engagement and comprehension during the academic activities. By analyzing these aspects, lecturers will be able to tailor their teaching methods to better suit the needs of students. With that understanding it will be possible to improve the learning outcomes and the overall educational experiences of the students.

Emotion Detection:

Emotion detection technologies have advanced from its initial focus on basic sentiment analysis to complex physiological monitoring. Initially, the technology focused on categorizing the sentiments as positive, negative, or neutral. However, with time, the need for the ability to grasp more emotional nuances became apparent. This was especially true for educational and interactive settings. In order to go beyond simple sentiment detection, today's advancements use more sophisticated deep learning and ML techniques. Using these techniques, it is possible to improve the capability to interpret complex emotional states. That is, this can assist to provide deeper insights into human emotions (Kusal et al., 2022). This can ultimately help to improve the engagement of the students.

(Rodríguez-Arce et al., 2020) analyzes the use of data from heart rate, skin temperature, and galvanic skin response signals to identify the stress related features in students. It used a variety of sensors to collect physiological data like Arduino board and several low-cost sensors such as the e-Health V2 sensor shield, for real-time data acquisition. These sensors were attached to students in non-invasive ways, such as by using adhesive tape to secure the GSR sensors to the fingers and a pulse oximeter to the thumb. This allowed to continuously monitor the physiological responses during various academic tasks. It explained that the use of these physiological indicators can make it possible to achieve high accuracy in predicting the stress and anxiety levels.

(Nishchal et al., 2020) explains how the cheating activities can be detected by analysing the posture of the body by using CCTV footage in classroom. It explains that it is possible to use sentiment analysis to decide if a student has an intention to cheat in the exams. By analysing the frequency of the detected emotions like anxiety or nervousness, this system provided evidence to support the hypothesis of the cheating intent.

(Alkilani and Nusir, 2022) explains about an emotion tracking system which analyses and detects the emotions of the students using CK+ face images dataset. This integrated Factor Analysis (FA) and Very Deep Convolutional Neural Network (VDCNN) classifier to ensure precise emotion recognition and to demonstrate superior performance in accurately identifying the various emotional states.

(Ozdamli et al., 2022) provides details about research that was conducted to assess the effectiveness of a facial recognition system that was designed to detect the student emotions and cheating during online exams. This made use of advanced computer vision algorithms and deep learning techniques to analyse the facial expressions in real-time. This was tested in a controlled environment where the students participated in online sessions. Here, the results highlighted that the system could accurately detect various emotional states. By detecting the emotional states, it was possible to improve the student engagement and the integrity during exams.

(Prabhakar et al., 2023) looked into the effectiveness of deep learning models in detecting stress indicators during online examinations. It made use of MobileNet and ResNet models which were trained on image datasets that include stress indicators like grinding teeth and scratching the head. It states that there was a notable accuracy in this approach. It also stated that stress can impact the exam performances of the students.

(Avdimiotis et al., 2024) explains about a structured experiment conducted using twenty-six postgraduate students to analyse how stress impacts emotions during examinations. Here, they have mainly used Electroencephalogram (EEG) technology to achieve this. Here, the data revealed that stress levels were higher during the written and oral exams compared to the rest phases. It also stated that stress directly affects the engagement and the focus of students. This highlights the need for strategies to manage the stress effectively to improve academic performance.

It is clear that understanding the emotions of the students can significantly improve their learning. By including an emotion detection system in SQLBooster, it will allow the instructors to monitor the emotions like the stress levels of the students when answering questions. This insight will help the lecturers to accurately determine whether the students are confident in solving SQL problems independently or if they need additional support.

Thus, the lecturers will be able to tailor their teaching methods to better meet the individual student needs.

Head pose Detection:

Head pose detection technologies can influence the way we understand and monitor student engagement and integrity during exams. These can be mainly useful in areas like e-learning. This section highlights the existing literature related to head pose detection.

(Ebisawa, 2008) describes a head pose detection method that was developed using a single camera to improve the safety by observing driver behaviour. The aim here was to prevent drivers from failing to keep their eyes on the road. This analysed the 3D positions of pupils and nostrils to improve the horizontal detection range. It accurately measured the head positions within a range of up to 45 degrees left or right horizontally, and up to 40 degrees up or down vertically. Then this method was applied using two cameras. When this was done, it could measure even wider, up to 60 degrees horizontally, showing that it is more precise and flexible compared to older methods that used two cameras from the beginning.

(Murphy-Chutorian and Trivedi, 2009) explains that head pose estimation plays a major role in applications that require understanding the orientation and movement of a human head and that it can reveal the focus of attention or intent of a person. It highlights the challenges and techniques in accurately estimating the orientation of a human head from digital images. It states that the accurate detection is complicated by factors like camera distortion, lighting conditions, and facial obstructions like glasses or hats. It also highlights the need for advanced solutions in practical applications for head pose monitoring.

(Samir et al., 2021) introduces a system that uses human pose estimation to detect cheating during exams by tracking the head postures and hand movements of the students. This system monitors for abnormal head angles, which could mean that a student is looking away from their work area. This system was capable of having advancements in areas related to monitoring and ensuring the integrity of examination environments. This system explained here was a scalable solution which is applicable to both online and on-site examination settings.

(Indi et al., 2021) used an innovative approach to monitor the online exams by using state-of-the-art ML techniques. It analysed head pose and eye gaze of exam takers and identified when a visual focus of attention (VFOA) of a student deviate from the test material on the screen that could indicate cheating behaviours. This only required the test-taker to have a webcam and internet connection. This system was highly accessible and scalable, and provided a solution to uphold the integrity in online assessments.

(Wang and Xia, 2021) presented an attention monitoring system that made use of head posture estimation to measure the student focus in educational settings, mainly during laboratory experiments. This system was based on a Raspberry Pi equipped with a camera and made use of ERT and EPnP algorithms to determine the head angles of yaw and pitch. Using this, it was possible to check if a student's gaze remains within the screen area. Using this approach, it was possible to monitor the student engagement in real-time and was able to provide the instructors with visual data on student attention. Using this approach, it was possible to improve the effectiveness of experimental teaching by ensuring that students remain focused on their tasks.

(Chai et al., 2023) used advanced methodologies for driver head pose detection using data collected from naturalistic driving scenarios. This monitoring system was capable of considering lower resolution and less controlled camera settings. This approach used ML techniques to classify the head poses of the drivers into Looking Forward, Looking Left, and Looking Right. The aim here was to improve the accuracy of head pose estimation in challenging real-world conditions.

By integrating head pose detection into SQLBooster, it will allow the tool to monitor the student engagement in real-time. This will help to improve the learning experience of the students and to further understand whether the students need any help with that particular SQL area.

Attentiveness Detection:

Attentiveness detection plays a major role in educational research. This has evolved from basic observational methods to computational approaches and contributes significantly to understand and improve the student engagement and learning outcomes.

(Negron and Graves, 2017) explains a system called the Classroom Attentiveness Classification Tool (ClassACT), which was designed to monitor the student attentiveness. It mainly made use of sensors integrated within tablets and PCs to collect and process data, in order to differentiate between attentive and inattentive behaviours effectively. This tool was capable of identifying academic dishonesty during assessments and assisted in improving educational engagement and upholding academic integrity.

(Canedo et al., 2018) developed an autonomous computer vision system to non-invasively monitor the classroom attentiveness. The aim of it was to identify off-task behaviours and to improve lecture dynamics by providing feedback to both instructors and learners. Here, the system integrated advanced face detection, facial feature extraction, and pose estimation technologies to assess the student attentiveness in real time. The research used OpenPose for body pose estimation and this enhanced the ability of the system to interpret physical cues related to attentiveness, such as orientation towards the teacher or engagement in note-taking activities. This allowed to continuously assess the engagement of the students and provided valuable feedback to the instructors on the effectiveness of their teaching methods and student interaction.

(Tankala et al., 2021) describes a multipurpose AI-driven attentiveness monitoring system that was developed to address the challenges of remote education and exam proctoring during the COVID-19. Here, visual cues of attentiveness in online settings were continuously tracked to ensure academic integrity and student engagement. Here, the system monitored the eye gaze, head movements, and mouth movements using a webcam-based AI model, to compute a real-time trust score to reflect the focus level of each student. The system showed results that proved that the model could effectively distinguish on-task students from those who were distracted or cheating. This study showed how multi-faceted visual monitoring (eyes, head, mouth) can provide instructors with an automated tool to measure student attentiveness and uphold exam integrity in virtual learning environments.

(Potluri et al., 2023) presented an approach for online proctoring through the development of an Attentive System. The purpose here was to monitor the student attentiveness and detect mischievous behaviours during online examinations. The system integrated a series of computer vision techniques and included face detection, head pose estimation, and face spoofing checks to make sure that there will be rigorous examination monitoring. The system used the Attentive-Net alongside Inception-Resnet for face recognition and pose estimation and evaluated the orientation of the examinee and attention in real-time. The empirical findings displayed that this system helps to increase the accuracy of detecting inattentiveness or fraudulent activities. This had an improved accuracy of 0.87 in distinguishing between genuine and deceitful actions.

(Elbawab and Henriques, 2023) developed a webcam-based ML approach to monitor student attentiveness in e-learning. Here, video data was collected from seven students and extracted 11 features from each student's face. This assisted to capture the physical indicators of attention such as eye aspect ratio for blink/drowsiness, yawn frequency, and head pose. The system used the human observer ratings of attentiveness as the ground truth and then these features were used to train and test decision trees, random forest, Support

Vector Machine (SVM), and XGBoost. This helped to automatically classify the attention levels. These empirical findings of this study validate that a low-cost, webcam-only system can effectively improve the student engagement remotely and provide instructors with attentiveness reports to improve the quality of online lectures.

(Hossen and Uddin, 2023) presented an AI-driven system which is capable of monitoring the student attentiveness in online classes using an extreme gradient boosting (XGBoost) classifier. The main aim here was to handle the challenge of evaluating student engagement and attention during virtual lessons. It made use of facial recognition for identity verification, face and hand tracking, mobile phone use detection, and body pose estimation to obtain a comprehensive picture of the behaviour of each student during class. XGBoost model was capable of achieving an accuracy of about 99.75% in distinguishing attentive vs. inattentive states on test data. It demonstrated robust attention detection capabilities and could generate detailed attentiveness reports via a web interface. The main contribution of this was a highly accurate, multi-modal approach to automatically detect and report the student attention levels.

Using an attentiveness detection algorithm can assist SQLBooster to improve its ability to monitor student engagement in real-time during the SQL learning sessions. With this, the lecturers will be able to gather important data like the attentiveness of the students when they are using SQLBooster.

Learning Dashboards and Visual Analytics:

Learning dashboards can improve both the student engagement and instructional oversight. (Fujita et al., 2019) explained that the dashboards displaying performance metrics such as login frequency and query success rates allow the instructors to monitor the students and tailor their support accordingly. However, this was mainly designed for instructors, and thus lacked real-time, actionable feedback for students.

(Park and Jo, n.d.) indicated that dashboards with real-time data and personalized progress trackers can help the students to monitor their learning and stay motivated. Similarly, (Ramaswami et al., 2023) found that dashboards with intuitive visual components, like progress graphs and interactive displays, increased the student engagement and encouraged continuous self-assessment. Furthermore, (Wang et al., 2023) highlighted that dashboards delivering personalized feedback can improve the academic performance and improvement of students. Thus, having a dashboard contributes to enhance both the learning experience and academic performance of the students. SQLBooster aims to use similar dashboard features to improve active student engagement and self-assessment.

2.2.3. Discussion of Previous Methodologies

This section will examine the methodologies of the existing literature, focusing on adaptive learning, feedback mechanisms, student monitoring, and automated grading.

Personalized and Adaptive Learning Systems:

Adaptive learning methodologies have evolved significantly starting from foundational tools like SQLator (Sadiq et al., 2004). This was one of the first web-based tools that were used to teach SQL and these tools focused on implementing methodologies that allowed the students to execute and evaluate SQL queries. This consisted of an equivalence engine that provided correct/incorrect feedback but lacked adaptive content customization. The tool was built using Microsoft COM components and static HTML, limiting its ability to adjust to a student's progress.

Methodologies in later researches, such as of DBLearn (Nalintippayawong et al., 2017), included adaptive engines to customize content and monitor student progress, marking a major move toward personalized learning. This system used student profiles to track the learning preferences and previous performance and used this information to create tailored learning paths and to adjust question difficulty. DBLearn included automatic question generation based on student interactions, allowing students to practice at their own pace through an adaptive content delivery system. However, it did not adjust question difficulty in real-time, keeping the complexity of questions static.

The methodologies used in (Taylor et al., 2021) and (Daugherty et al., 2022) expanded on adaptive learning by focusing on real-time analysis of student input to adjust content delivery. According to (Eapen and V S, 2023), even though there are significant challenges, LLMs can be used for personalization and customization and can assist in enhancing user interaction and satisfaction across various applications, including educational tools and learning systems. However, these current adaptive systems often have limited use of advanced methods like LlamaIndex, and LLMs. These techniques can make content delivery more personalized by interpreting the student responses and generating educational material based on natural language queries (Dingmin et al., 2024). Thus, techniques such as LlamaIndex and LLMs can be used to offer more effective adaptations to individual learning paths, significantly enhancing the personalization of content delivery.

Feedback Mechanism:

Initial methodologies for feedback in SQL learning tools offered basic binary responses. As an example SQLator (Sadiq et al., 2004) used equivalence checking to indicate whether a query was correct but did not provide further explanations. (Soler et al., 2006) introduced an iterative submission approach, allowing students to receive syntax-focused feedback and resubmit their queries. These early automated processes did not offer any personalized feedback.

(Reilly, 2018) highlighted instructor-driven methodologies that provided detailed and active feedback. They were able to promote direct interaction but demanded considerable manual effort. Automated feedback systems like MySQL Sandbox (Permpool et al., 2019) offered structured, error-specific feedback through predefined parsing techniques. However, these methods did not adapt dynamically to the learning needs of students.

Recent feedback methodologies have integrated advanced techniques like LLM and RAG. RAG frameworks can pull from extensive data repositories to deliver comprehensive feedback that is tailored according to the student inputs. These can help to bridge the gaps in traditional approaches (Hu and Lu, 2024). Moreover, LLMs can further enhance these systems by enabling even more finely tuned personalization options, using its vast training data to generate responses that are deeply aligned with the individual user needs (Raiaan et al., 2024). The use of these techniques can improve feedback and personalization, and can address the limitations of earlier systems.

Automated Grading:

Automated grading methodologies have significantly improved from basic techniques to those capable of partial credit evaluation. Early models, such as SQLator (Sadiq et al., 2004) was only capable of providing binary feedback, indicating whether the SQL query was correct. These focused on simple SELECT statements without acknowledging partially correct answers. Methodological advancements were seen in tools like the one described by (Joseph et al., 2016), which introduced component-based grading for SQL queries. This approach enabled systems to evaluate different parts of a query, such as SELECT, WHERE, and JOIN clauses. This approach allowed to provide partial credits and helped in providing more detailed and fair assessment to complex queries.

Methodologies, as of those in DBLearn (Nalintippayawong et al., 2017), expanded the range of supported SQL types to include both DML and DDL for automated grading. However, this was not able to provide partial grading.

Further advancements could be seen in the methodologies explained in (Venkatamuniyappa, 2018) and (Panni and Hoque, 2020). Here, the focus was on providing partial evaluation to enhance the grading of SELECT queries. These methods provided a more detailed analysis of query correctness by evaluating individual components, allowing for more precise feedback. Therefore, while significant progress has been made, there is still a gap in providing comprehensive partial grading for more complex DML and DDL queries.

Student Monitoring:

When reviewing the previous methodologies on student monitoring, it is clear that there are significant advancements in the fields of emotion detection, head pose estimation, and attentiveness (Alkilani and Nusir, 2022; Hossen and Uddin, 2023; Murphy-Chutorian and Trivedi, 2009; Wang and Xia, 2021). Methodologies in these have the ability to analyse various aspects of student behaviour. Building on these advancements, it is important to examine how these methodologies have been applied and integrated under the section emotion detection, head pose detection, and attentiveness detection. With this understanding it will be possible to understand the potential for further research.

Emotion Detection:

Recent advancements in emotion detection have evolved into sophisticated automated systems that uses physiological signals and deep learning algorithms. With the help of deep learning techniques, it is possible to understand the emotional responses in a more accurate way. Having a clear understanding of the existing methodologies in emotion detection systems can pave the way for better approaches that can in turn help to make improvements in the educational technology.

(Schmidt et al., 2019) outlined a methodology that used wearable sensors for affect recognition. It focused on capturing physiological and inertial parameters to detect emotional states. This methodology used the capabilities of wearable devices, which were integrated with sensors that monitor changes related to the cardiac system or electrodermal activity. Here, the data that can reflect the emotional states of the users was collected continuously through measurable physiological responses. The methodology consisted of stages such as data collection, pre-processing, feature extraction, and classification. It used a variety of ML techniques to analyse the data that was obtained from the sensors and with this, it was possible to detect various emotional states. Even though this study advanced the field by using wearable sensors to detect emotional states, the methodology still required users to wear specialized devices. This limited its practical applicability in everyday settings.

(Rodríguez-Arce et al., 2020) involved an approach that uses a combination of physiological data collection and ML algorithms to detect stress and anxiety in academic environments. This used an Arduino board with low-cost sensors to measure 21 physiological features across five signals from 21 students. The collected data underwent pre-processing, feature extraction, and classification stages. It used the State-Trait Anxiety Inventory (STAI) to assess the anxiety levels of the students and correlated the physiological indicators with self-reported emotional states. Even though this study achieved high accuracy in recognizing stress and anxiety through physiological features, this approach was limited because of its reliance on wearable sensors. Furthermore, it did not incorporate any head pose cues to measure the student attentiveness.

(Alkilani and Nusir, 2022) presented an Emotion Tracking System (ETS) that used facial expression recognition with ML to assess the impact of the emotional states of the students on their academic performance. This system made use of a VDCNN classifier and FA for dimensionality reduction and this helped to increase the accuracy and efficiency of emotion detection. Here, the process began with pre-processing using the Non-Local Means technique to improve the clarity of facial images and reduce noise. Then FA was applied to reduce the dimensionality of the data and this helped to maintain a meaningful representation of facial expressions. Next, VDCNN classifier was used to analyse these processed images to identify the emotional expressions. This approach allowed the system to provide actionable insights into the emotional well-being of the students during exams. This study was also able to achieve high accuracy in classifying student facial expressions. However, this solely focused on facial emotion detection without integrating head pose cues.

(Ozdamli et al., 2022) outlined a methodology that uses a combination of computer vision algorithms and deep learning techniques to improve the online exam integrity and emotional analysis. This system used Convolutional Neural Networks (CNNs) for real-time facial emotion recognition and to detect cheating during online exams. It analysed the facial expressions to identify various emotional states and assessed for behaviours that might indicate cheating such as gaze diversion. This methodology also included continuous monitoring through video feeds. This helped to maintain the credibility of online assessments. It also supported emotional monitoring to understand the stress levels. Thus, this helped in both improving the educational outcomes and exam security. However, this was still limited by drawbacks such as the inability to automatically detect subtle head movements.

(Prabhakar et al., 2023) focused on detecting stress levels from facial expressions and actions of students during examinations. This approach used advanced deep learning models like MobileNet and ResNet to detect stress from video and image data captured during exams. Here, initially data was collected from videos and images and then they were segmented into frames to analyse stress indicators like teeth grinding. The study used both pre-trained models and custom training sessions to fine-tune the models for stress detection. This approach allowed detailed monitoring and assessment of stress levels. However, this focused only on stress-related cues.

(Avdimiotis et al., 2024) explained about an experimental protocol incorporating the Emotiv Epoc+ EEG device to monitor the emotional states of the students during examinations. This used a modified Trier Social Stress Test (TSST) to measure the impact of stress on academic performance. The EEG device captured data on various emotions like stress, engagement, excitement, interest, focus, and relaxation throughout the examination. Then this data was analysed using network analysis techniques to understand the relationship between the emotions and their influence on the performance of students. Here, it showed that stress can both be a motivator and a hindrance in academic environments. Even though this study was able to achieve a detailed understanding of the interaction between stress and student engagement, it was still limited because it had to rely on EEG measurements. It also lacked the integration of visual cues such as head pose or attentiveness detection.

This review of the methodologies of the current emotion detection technologies depicts the potential to improve student monitoring systems. By having an understanding of the existing methodologies, SQLBooster can offer a novel approach to improve the student engagement by understanding their emotional states.

Head pose Detection:

Head pose estimation technologies have significantly evolved from initial techniques such as multi-class linear SVMs and Random Forest classifiers (Murphy-Chutorian and Trivedi, 2009). These methods mainly relied on a clear relationship between facial feature positions and head pose and could provide accurate estimations without needing any complex steps. Recent developments have transitioned to deep learning techniques such as Deep Neural

Networks (DNNs). These are capable of processing a broader range of pose variations and can operate effectively in diverse environments (Asperti and Filippini, 2023).

(Ebisawa, 2008) explained about a single-camera system which is equipped with specialized ring-like lighting. It aimed to improve the pupil and nostril visibility. The system processed images to detect differential brightness in pupils. This helped in pinpointing their precise location in real-time. By calculating the 3D positions of the pupils and nostrils using geometric constraints on their distances, this method used a pinhole camera model to derive directional vectors from these facial features. These vectors helped to determine the orientation of the head by analysing the planes formed between the pupils and each nostril. This helped in obtaining accurate head pose estimation. However, this focused only on head orientation tracking. It did did not integrate any detection of the emotional states of the users.

(Indi et al., 2021) used an FSA-Net deep learning model to estimate the head pose from a single image. This model outputs a 3D vector containing yaw, pitch, and roll angles by mapping an image to these pose angles. This was done by using a function that minimizes mean absolute error. Here, FSA-Net used an Soft Stagewise Regression Network architecture. This was capable of performing a hierarchical classification and regression. This network combined feature maps from multiple stages through element-wise multiplication and used pooling to preserve spatial information without loss. Additionally, the network generated attention maps for each stage using scoring functions. This helped in the accurate and robust estimation of head poses. However, this did not integrate any analysis of the emotional states of the students.

(Samir et al., 2021) made use of Tensorflow-PoseNet model to analyse the video footage of students during exams. This system was designed to detect cheating by continuously validating the head posture and the hand movements of the students. The system applied multiple-pose estimation techniques. This allowed to track several students within a single frame. The Tensorflow-PoseNet model processed the camera images to identify key points on human figures such as nose, eyes, and ears. The system evaluated these key points to determine if the head angle deviates from a predefined normal range. This deviation, if significant, can suggest potential cheating behaviour. Even though this study achieved high-accuracy detection of cheating behaviours, this was still limited because it concentrated only on the on the physical cues of cheating. It did not integrate any facial emotion analysis.

(Chai et al., 2023) focused on analysing the attention of the drivers when driving by examining their head pose using naturalistic driving data. Here, the video data was captured in less controlled, real-world settings. Due to the lower resolution and non-ideal camera angles, this posed unique challenges compared to laboratory conditions. The methodology used a Bidirectional Gated Recurrent Unit (BiGRU) alongside other techniques to classify the head pose into the categories looking forward, left, or right. Here, key facial points were extracted using the RetinaFace detector. The analysing was done by using the BiGRU framework to account for temporal dynamics and to improve detection accuracy under varied naturalistic conditions. This study achieved enhanced accuracy and range in head pose estimation. However, this methodology was specifically focused on tracking the head orientation of the driver and did not consider the emotional states of him.
Having an understanding of the previous methodologies helps to determine how deep learning techniques can be adopted to implement a head pose detection model for SQLBooster. Integrating such a model to SQLBooster can help to improve the learning experiences.

Attentiveness Detection:

Attentiveness detection have transitioned into systems that uses physiological signals and algorithms to monitor and improve the student engagement. In order to provide real-time insights into student attentiveness, this evolving field uses a variety of ML techniques and sensory data analysis methods such as eye tracking, head pose estimation, and facial expression analysis.
(Negron and Graves, 2017) explains a tool called Classroom Attentiveness Classification Tool (ClassACT) which was designed to monitor and classify the student behaviour during the educational activities as either attentive or inattentive. ClassACT used built-in sensors within devices such as tablets or PCs to collect various data points like camera images, microphone sounds, and system interactions. Here, the data collector gathered the environmental and user inputs continuously. Then an image processor was used to analyse the visual changes and a sound processor was used to assess the audio levels. Finally, the system used a data aggregator to compile these inputs into a structured format that feeds into classifiers such as the Multilayer Perceptron, SVM, and Proximal Support Vector Machine. Then, these classifiers were able to determine the likelihood of attentive versus inattentive behaviour based on the aggregated data. While ClassACT effectively distinguishes between attentive and inattentive behaviour using device sensors, it does not

consider the contextual meaning of the student expressions. That is, there is a possibility that the system will misclassify engagement due to its reliance on binary attentiveness classification without considering the emotional states or head pose adjustments.

(Tankala et al., 2021) described a system that uses an AI-based model to monitor student attentiveness during online exams. This was achieved by analysing live video feeds captured through high-resolution webcams. In order to assess the focus and detect potential cheating behaviours, the system tracked the head and mouth movements. It calculated a trust score based on the deviations from the expected behaviours. During off-task behaviours, such as when the students were distracted, the trust score was dropped. This score provided with a provides a quantifiable measure of each student's engagement. However, this method relied on a limited set of facial motion cues and does not consider emotional indicators of engagement.

(Potluri et al., 2023) demonstrated an online proctoring system which could assess the attentiveness and detect misbehaviour during exams. This system combined face detection, multiple person detection, face spoofing, and head pose estimation using Attentive-Net. Initially, a live video of the examinee was captured to analyse their activities. For face detection, system used Attentive-Net combined with a face net algorithm to extract facial features and landmarks. Then it evaluated the alignment of faces using an Affine Transformation rotation matrix. It used a shallow CNN called Liveness net to initiate face spoofing detection for aligned faces. In order to estimate the head pose, the system used the SolvePnp equation and this helped to flag the suspicious behaviours effectively. However, this proctoring solution only focused on physical cues. It did not incorporate any analysis of the emotional state of the students. This shows a lack of integration between emotion recognition and head pose estimation that SQLBooster aims to address.

(Hossen and Uddin, 2023) described an attention monitoring system that uses an AI-based approach using the XGBoost classifier. It was used to evaluate the attentiveness in online classes. The system used various techniques such as facial recognition, hand tracking, mobile phone detection, and pose estimation, to continuously analyse the video feed of students. The system could identify various behavioural cues that indicate attentiveness. Once these behaviours are detected and the relevant data is extracted, it was then used to train the XGBoost classifier. Then the model evaluated the data to determine the attentiveness levels of students during online classes. The effectiveness of the system was highlighted by its ability to provide detailed, anonymous reports on student attentiveness. However, this only focused on the observable physical behaviours and omitted the direct emotion recognition. Thus, the system could misinterpret the engagement of a student when the student is physically present but mentally distracted.

(Elbawab and Henriques, 2023) focused on evaluating the student attentiveness through a combination of emotional and non-emotional cues captured via a webcam. This study used various ML algorithms, such as decision trees, random forests, SVM, and XGBoost to analyse the video data collected from students during e-learning sessions. The video data was processed to extract features like eye aspect ratio, yawn aspect ratio, head pose, and emotional states. Then these were used to train the ML models. Here, the aim was to develop a classifier that can accurately evaluate the levels of attentiveness by correlating physiological and behavioural indicators with observed attention levels. While this study integrates both emotional and non-emotional features for attentiveness detection, it does not coordinate head pose with emotional states. The model treats these features separately and does not assess whether the detected emotion is relevant to the learning context.

With the understanding of existing methodologies, SQLBooster will be able to decide on the best approach it can take to develop an attention detection model for SQLBooster that addresses the limitations in the current tools.

Building on the detailed exploration of existing methodologies in emotion detection, head pose detection, and attentiveness detection, SQLBooster aims to use advanced AI models to integrate these advancements to address the limitations observed in previous systems.

2.2.4. The Research Gap

While there is considerable development in adaptive learning systems, student monitoring systems, automated grading, and feedback mechanisms, there are still some gaps that SQLBooster try to address. The Table 1 provides a brief comparison of selected tools, to demonstrate the features and the limitations of the existing research. While not exhaustive, this selection highlights the key areas where SQLBooster will make contributions. Table 1 will also show that there is no tool that provides all the features adaptive questioning, dynamic progressive difficulty, partial grading, real-time personalized feedback, and student monitoring. The dashboard component is not discussed here as a gap, as it is not a research focus but is included to improve user experience.

Table 1. Comparison with existing literature

Research Paper	Adaptive Questioning	Dynamic Progressive Difficulty	Partial Grading of SQL	Real-Time Personalized Feedback	Student Monitoring (Emotion, head pose, attentiveness)	Measure student tension based on emotion they feel like happiness, etc.
DBLearn (Nalintippayawong et al., 2017)	✓	✗	✗	Partial – Not Detailed	✗	✗
SQLator (Sadiq et al., 2004)	✗	✗	✗	Partial – Binary and not Detailed	✗	✗
SQL Learning Tool using MySQL Sandbox (Permpool et al., 2019)	✗	✗	✗	Partial – Binary and not Detailed	✗	✗
XData System (Joseph et al., 2016)	✗	✗	Partial – Only for SELECT queries	✓ (Mainly for incorrect queries)	✗	✗
Coping with examination stress (Avdimiotis et al., 2024)	✗	✗	✗	✗	Partial – Only measure emotion	✗
Research and design of an attention monitoring system based on head posture estimation (Wang and Xia, 2021)	✗	✗	✗	✗	Partial – Does not measure emotion.	✗
Attention monitoring of students during online classes using XGBoost classifier (Hossen and Uddin, 2023)	✗	✗	✗	✗	Partial – Does not consider emotion	✗
SQLBooster (This Solution)	✓	✓	✓	✓	✓	✓

The following further explains the research gaps that SQLBooster aims to address under the areas personalized and adaptive learning systems, feedback mechanisms, automated grading, and student monitoring.

Personalized and Adaptive Learning Systems:

The adaptive learning systems have evolved over time from basic, static tools which can only provide uniform content to more advanced platforms that can tailor the learning experiences based on the interactions of the students. While there are tools such as DBLearn (Nalintippayawong et al., 2017) which have made progress in adaptive learning by customizing content based on student profiles and performance, they are still not capable of providing questions that increase in complexity progressively based on the mastery levels of the students. Current systems, as highlighted by (Taylor et al., 2021), can customize the learning paths and offer immediate feedback through real-time adjustments. However, they are limited in their ability to align the complexity of the questions with the mastery level of the students and adapt content dynamically as the learner progresses. With the help of various technologies such as RAG, LlamaIndex and LLM, SQLBooster aims to address these gaps by not only personalizing the content but also by adjusting the complexity of the questions dynamically to challenge and improve the student learning progressively.

Feedback Mechanisms:

Feedback has evolved from binary responses (Sadiq et al., 2004) to more detailed, error-specific feedback (Permpool et al., 2019). However, the feedback provided by most of the tools are not comprehensive, adaptive, and is not tailored according to the performance of the individual students. (Gupta, 2024) emphasizes the importance of systems that are capable of boosting the confidence of students when they get the response correct and offering clear, constructive guidance when they get something wrong. SQLBooster aims to address these gaps by delivering more comprehensive and contextually relevant feedback to not only explain errors but also to guide the students on how to improve.

Automated Grading:

Existing grading tools have evolved from just being able to mark the responses as correct or wrong, to being able to provide partial marks. However, even though it is possible to grade more comprehensive queries (Atchariyachanvanich et al., 2017), partial grading is largely restricted to SELECT statements. Since partial grading is not available for SQL types such as DDL and advanced DML queries, SQLBooster aims to bridge this gap. This will help the students to pinpoint their areas of weaknesses and obtain a deeper understanding of the SQL concepts.

Student Monitoring:

Recent advancements in student monitoring have improved areas such as emotion detection, head pose estimation, and attentiveness detection. However, several limitations can be seen in the existing research that can impact the accuracy and effectiveness of these systems in real-world applications.

Most of the existing systems lack an integrated approach. That is, most of them focus on one or two aspects of student monitoring and does not consider all the aspects emotion, head pose, and attentiveness. This lack of consideration can lead to inaccuracies. As an example, many studies have explored emotion recognition using facial expressions but does not consider the inaccuracies that might occur due to various changes in head pose (Alkilani and Nusir, 2022; Bhardwaj et al., 2024; Ozdamli et al., 2022).

Another limitation in emotion detection research is context awareness. That is, most of the systems classify facial expressions into emotions but does not consider when that emotion is displayed. This can result in false interpretations. As an example, the surprised face of a student might be due to a noise in the room rather than the learning content. Without proper context, the system can incorrectly label that surprise as related to the lesson (Alkilani and Nusir, 2022).

Head pose estimation also plays a major role in determining whether a student is paying attention. Most of the current models face accuracy issues under the real-world conditions (Murphy-Chutorian and Trivedi, 2009). Factors such as the student partially covering their face, can make the head pose calculations inaccurate. These factors can cause the systems to misinterpret the head direction.

Another gap in the current research is the inability of the existing tools to measure the student tension or stress by using emotional understanding. Most of the existing systems that attempt to detect stress rely on physiological signals such as heart rate or skin conductance. These models do not consider how different emotions can contribute to stress.

There are tools that can identify stress directly as a singular emotional state that can impact the exam performance. However, they are not capable of mapping emotions like happiness, surprise, or sadness into specific tension levels (Prabhakar et al., 2023).

Attentiveness detection often relies on a narrow set of indicators such as eye tracking or basic facial landmarks such as whether the eyes are open or not. While useful, these overlook the other behavioral signs such as engagement or disengagement. As an example, a student might keep their eyes on the screen but they maybe fatigued. These can be identified by indicators such as the duration of eyes closed. Most of the time, these indicators are ignored.

SQLBooster aims to bridge this gap by integrating head pose estimation, attentiveness detection, and emotion-based tension detection to enhance the accuracy of student engagement assessment. By incorporating head pose analysis, the system can distinguish between emotions that arise from interaction with learning content and those triggered by external distractions. Furthermore, the attentiveness detection model considers indicators such as eye-closure duration, offering a more comprehensive understanding of the student's level of engagement.

Thus, as a summary, the field of SQL education has significantly benefited because of the advancements in adaptive learning, feedback mechanisms, automated grading, and student monitoring. However, an integrated system with improved personalization, comprehensive real-time feedback, student monitoring and partial grading across all SQL types is still lacking. SQLBooster will bridge these gaps by addressing these limitations.

2.3. Conclusion

This literature review has explored various foundational theories, empirical research, existing gaps, and discussions on methodologies in SQL education tools. The literature is mainly focused on the themes adaptive learning systems, feedback mechanisms, automated grading, student monitoring, and learning dashboards. The development from basic tools like SQLator to more advanced adaptive systems such as DBLearn shows that there is a significant progress in the SQL learning technologies. However, these advancements have limitations, especially in areas such as student monitoring, delivering progressively challenging and personalized content, providing feedback tailored to individual student learning paths, and offering comprehensive partial grading beyond the basic queries.

These limitations highlight the necessity for a solution like SQLBooster. SQLBooster aims to bridge these gaps by integrating adaptive technology, real-time feedback mechanisms, student monitoring strategies, and advanced partial grading using technologies such as deep learning, LlamaIndex, RAG and LLM. With a tool such as this, it will be possible to improve the student outcomes by providing personalized and adaptive learning experiences with student monitoring. Ultimately, this development can contribute to more effective educational practices.

CHAPTER 3: METHODOLOGY

3.1. Introduction

The methodology chapter outlines the research approach and the design used to achieve the objective of the study which is to improve the SQL writing skills among the undergraduate computing students. Traditional learning methods rely on static content and linear progression mostly. This limits their ability to address the individual learning needs (Elshani and Pireva Nuci, 2021). In response, this study uses advanced educational technologies to create a dynamic and responsive learning platform. This helps to ensure a more personalized and adaptive SQL learning experience. The methodologies applied in SQLBooster are designed to overcome the core challenges in SQL education, making learning more interactive, feedback-driven, and tailored to the proficiency levels of the students.

SQLBooster uses a multi-agent AI architecture powered by LLMs, specifically OpenAI's GPT-4o, and is orchestrated through the LlamaIndex framework. The system is designed to generate personalized questions, deliver real-time and adaptive feedback, perform partial grading, and track student progress. Deep learning techniques are also used for student monitoring. This methodology chapter further explores components such as the hierarchical agent framework, dual-memory systems, RAG, and deep learning. The subsequent sections provide details about the research philosophy, research approach, and the proposed solution design and implementation.

3.2. Body

3.2.1. Research Philosophy

This research aligns closely with the pragmatist philosophy, and supports the idea that concepts and theories prove their value through practical success (Kaushik and Christine, 2019). Pragmatism is mainly suitable for studies like this, because the objective is to develop solutions that address the real-world problems. It recognizes that there can be multiple valid approaches to learning, understanding, and problem-solving, and provides a framework that addresses the learning needs of the students. Pragmatism focuses on practical outcomes, and this aligns with the development of SQLBooster. This is because the effectiveness is measured by its ability to improve the SQL skills of the students. The development of SQLBooster begins with an initial evaluation to include essential features based on student feedback. This initial feedback phase is important to tailor the tool to meet the educational needs effectively.

Since the development of this is done based on the feedback obtained from the students and the lecturers, and since the tool will be personalized to suit each student, this reflects the fundamental principles of pragmatism, in its design and function. By using a pragmatist approach, SQLBooster positions itself at the intersection of theory and practice. Here, the aim is to create a tool that is grounded in methodical research and delivers improvements in SQL education. Building on the pragmatist foundation, the next section details the specific research approaches taken to evaluate and validate the effectiveness of SQLBooster.

3.2.2. Research Approach

SQLBooster follows the design science methodology (Brocke et al., 2020) as it provides a structured approach for creating and improving the technological solutions to tackle SQL related educational challenges. This method guides the systematic development of SQLBooster through the phases of design, development, and evaluation.

Initially, to identify the areas where undergraduate students struggle most in, and to evaluate the potential areas of improvement in SQL education, a questionnaire was distributed to the students in several universities. This questionnaire contained both closed-ended and open-ended questions, and thus could be used for both qualitative and quantitative analysis. The closed-ended questions aimed to gather data on areas students find challenging and the desired features. The open-ended questions allowed the students to express their suggestions in more detail to further understand the problems in the current SQL education methods. This dual approach allowed the collection of a broad spectrum of data, ranging from statistical frequencies to personal narratives. This method helped to ensure that the tool was designed with a clear understanding of the needs and challenges of the students. The questionnaire was anonymous and this encouraged the honest feedback from participants.

After developing SQLBooster based on the feedback obtained, the tool was presented to the students to gather initial reactions and feedback. This helped to get an understanding about the standing of the user interface, functionality, and the overall appeal of the tool. Students were asked questions such as their likelihood to use SQLBooster, their interest in its various features, and their overall impressions of the tool. This feedback helped to ensure that SQLBooster not only addresses the educational gaps identified but also aligns with user preferences and requirements. Therefore, a key component of the design science approach here was the formative evaluation.

SQLBooster aligns with the Research Onion model, since it provides a systematic approach to conduct research. This model guided the selection of appropriate research strategies, methods, and data collection techniques, and helped to ensure that each layer of the research design contributed to an effective study (Tengli, 2020).

By integrating the Design Science Methodology, formative evaluation, and mixed research methods (Doyle et al., 2009) within the framework of the Research Onion model, SQLBooster will be able to effectively improve the SQL learning experiences and the outcomes of the undergraduate students.

3.2.3. Proposed Solution Design and Implementation

In this section, design and implementation of SQLBooster will be presented under two main parts to ensure clarity. Part 1 is called 'Personalization and Feedback Mechanisms'. This section will cover the functionalities such as the generation of personalized questions, personalized content, adaptive feedback mechanisms, partial grading, and the integration of the learning analytics dashboards. Part 2 is called 'Student Monitoring'. This section will focus on the monitoring aspects of the solution through models for head pose detection, attentiveness detection, and emotion-based tension detection. These components will be integrated in a sequential process. The system will first check for focus using head pose detection, then verify attentiveness, and finally analyze emotion, mapping it to tension levels to enable informed decisions about whether additional support is needed.

Part 1: Personalization and Feedback Mechanisms

Architectural Overview

SQLBooster is designed to create a highly interactive and adaptive learning environment which is capable of transforming the way that SQL is taught and learned. The architecture of this tool is structured so that it can deliver knowledge as well as adapt dynamically to the learning pace and style of each student. By understanding the architecture and the technologies behind SQLBooster, it will be possible for us to get an understanding on how to achieve these educational goals.

The architecture of SQLBooster uses powerful frontend and backend technologies, to create a user-friendly interface with efficient data processing capabilities. The frontend of the system uses Next.js which is a React framework that supports features like server-side rendering. This helps to ensure a fast and smooth user experience. Furthermore, Tailwind CSS and Radix UI are used to improve the aesthetic appeal and the accessibility of the interface. The WebSocket technology ensures that the communication between the client and the server is instantaneous.

On the backend, SQLBooster uses FastAPI, which is a high-performance web framework. This is known for its speed, ease of use, and robustness. To manage the data MongoDB database is used, especially because of its high performance and flexibility. This integration ensures that the backend can manage complex queries and operations that are necessary for the learning platform. FastAPI-Cache is used to enhance performance through efficient in-memory caching. This facilitates rapid data processing and retrieval.

The architecture of SQLBooster is a modular, hierarchical multi-agent system which is designed to optimize the SQL education through distributed intelligence and real-time coordination. The diagram below shows SQLBooster's high-level architecture and illustrates the specific components and their interactions within the system.

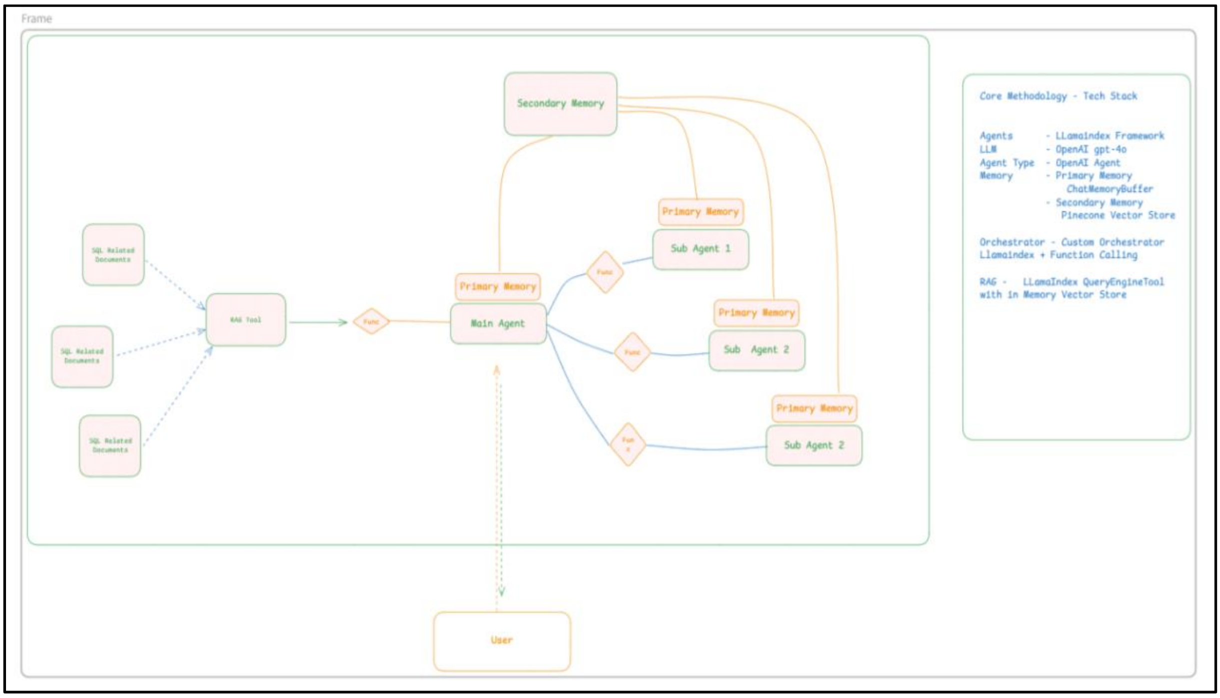


Figure 1. High Level Architecture of personalization component

As shown in Figure 1, the architecture of SQLBooster is enclosed within a green frame labeled "Frame". The student monitoring aspect, discussed separately in Part 2, is not depicted in this figure. Here, the components are connected using color-coded lines and nodes, which represent the flow of data and functional interactions between various components.

At the core of this architecture lies the **Main Agent**, which serves as the orchestrator of the entire system. Represented as a green rectangle in the center of Figure 1, it is built using a custom orchestrator within the LlamaIndex framework. This agent orchestrates the operations of the system by assigning tasks to Sub-Agents. These tasks are assigned by using function calling mechanisms and these are represented using the orange diamonds which are labelled as

"Func". The main agent also manages the memory synchronization and ensures that all of the components in the system operate in a reliable and efficient way.

Surrounding the Main Agent are three **Sub-Agents**, each designed to fulfill a specific role in the personalized learning process. These are shown using a green rectangle and are labelled as Sub Agent 1 (Learning Strategy Agent), Sub Agent 2 (Data Management Agent), and Sub Agent 3 (Progress Management Agent). These play a major role in tailoring the learning processes to match the individual needs. Each Sub-Agent has a primary memory which stores session-specific data. These primary memories enable seamless collaboration between the Sub-Agents and the Main Agent, supporting adaptive and context-aware learning experiences.

The **User Interface** which is crucial for the user interactions is represented by an orange rectangle labeled "User" at the bottom of the diagram. This directly connects to the main agent as shown via the dashed green arrow. This symbolizes the real-time interaction that is facilitated by the WebSocket technology. This connection plays a major role in providing instantaneous responses and adjustments based on student inputs.

The **Memory System** in SQLBooster consists of both short-term and long-term storage components. The secondary memory of the memory system is shown as a green rectangle at the top of the diagram. This is also referred to as 'Pinecone Vector Store' and serves as a long-term storage solution. It interconnects with all the agents and this is shown in Figure 1 via orange curved lines. The Primary Memory (implemented as ChatMemoryBuffer) of the system provide short-term caching which is crucial for storing active session data such as current quiz responses. This is essential for quick data retrieval and processing during active learning sessions. Thus, this allows for low-latency interactions that are important for an interactive learning environment.

The left section of the diagram highlights the **RAG Tool** and is shown as a green rectangle. It links to the "SQL-related Documents" via dashed blue arrows, a connection that enables SQLBooster's RAG capabilities. The tool uses LlamaIndex's QueryEngineTool to generate personalized learning content dynamically from the stored SQL documents. By integrating external SQL related materials into the learning process, the RAG system expands the LLM's knowledge base. It retrieves contextually relevant content, such as JOIN examples and helps to make sure that the instructions are accurate and customized for the learner.

Overall, the SQLBooster architecture follows a microservices inspired design that emphasizes modularity and scalability. This is consistent with the principles of agent-based systems (Wooldridge and R. Jennings, 1995). Inter-agent communication is facilitated through function-calling mechanisms, as indicated by the "Func" nodes in Figure 1. For instance, the Main Agent can use a function to request a personalized SQL query from the Learning Strategy Agent (LSA). After processing the request, the LSA will send the customized query back to the Main Agent for further actions. This process helps to ensure effective task delegation and to facilitate continuous feedback loops. These are essential for the real-time adaptability of the system.

By coordinating these components, SQLBooster creates a responsive and intelligent environment for SQL education which is capable of delivering dynamic instruction, adaptive feedback, and contextualized support for learners in real time.

Agent Design and Capabilities

SQLBooster consists of three specialized Sub-Agents. Each of these agents are enhanced by the capabilities of GPT-4o and are integrated through the LlamaIndex framework. These agents are designed to manage distinct areas of the learning process and helps to enable a dynamic and personalized educational environment.

The first of these, the **Learning Strategy Agent (LSA)**, plays a critical role in personalizing the learning experience. The primary role of this includes generating SQL questions that are tailored to the individual student and providing immediate, contextual feedback. This agent mainly makes use of GPT-4o to dynamically produce SQL questions that are context aware. These questions are designed so that they progressively increase in difficulty based on the performance metrics of the students stored in the secondary memory. Furthermore, the LSA uses a RAG approach to provide detailed explanations and contextual feedback. As an example, it is capable of simplifying explanations of complex SQL subqueries for beginners. This can improve the understanding of the students. Additionally, the LSA is responsible for path optimization. It is capable of intelligently sequencing the lessons to ensure progressive skill development and mastery in SQL.

Data Management Agent (DMA) ensures the integrity and accessibility of data within SQLBooster. It validates the quiz results and other data inputs using Pydantic schemas and stores them efficiently in a MongoDB database. Furthermore, the DMA tracks the key performance metrics such as the accuracy. This is crucial to assess the student progress and to adjust the learning paths dynamically. Additionally, this agent can manage session data effectively. That is, it helps in ensuring rapid retrieval and smooth user experiences. This involves caching the session and the interaction data in the primary memory to ensure quick access during active learning sessions.

The third sub-agent, the **Progress Management Agent (PMA)**, oversees the curriculum progression. This is crucial to guide the students through their educational journey in SQL. This agent tracks the educational milestones of the students and maps the student progress to specific competencies. It can adjust the learning paths in a dynamic way by using a tool calling mechanism to make sure that learners are consistently challenged at the appropriate level of difficulty. Furthermore, PMA can manage level transitions. This helps to ensure that the students have mastered the prerequisite skills before advancing.

These sub-agents operate collaboratively under the coordination of the Main Agent, as illustrated in Figure 1. Their interactions are facilitated by function-calling operations (represented by the "Func" nodes) and are supported by the shared use of memory and RAG components. This architecture enables SQLBooster to deliver a seamless and adaptive learning experience grounded in intelligent agent collaboration.

Design and Implementation of Features

This section presents the design and implementation details of the core features within SQLBooster.

The **personalized learning functionality** in SQLBooster is powered by a multi-agent AI architecture built on OpenAI's GPT-4o. At the core of this system is a main agent that coordinates with specialized sub-agents, each responsible for handling specific aspects of the learning process, as described earlier. These agents work together to dynamically generate personalized SQL questions within a conversational interface. Additionally, RAG is used to further improve the personalization by using external SQL related documents into the learning process. This allows the LSA to tailor the questions and content dynamically based on the individual student needs. SQLBooster uses the memory system to efficiently store and retrieve student performance data and educational content, enabling personalized experiences.

To complement this personalized questioning system, there is a comprehensive guidebook and the capability for the lecturers to add their own instructional documents. This allows the students to ask any doubts that they have related to the content and improve their knowledge. Furthermore, if the students need further assistance or external references, SQLBooster can recommend and provide links to suitable additional resources. To orchestrate these LlamaIndex framework is used. This enhances the capability of the system to deliver rich, customized educational experiences. By following this approach, SQLBooster can provide a dynamic and interactive platform where personalization extends beyond mere questioning.

The **question categorization process** is also guided by GPT-4o and integrated into a structured curriculum, which is segmented into progressive levels. These levels are tailored to gradually develop SQL proficiency from fundamental concepts to advanced techniques. The LSA dynamically categorizes questions according to a student’s performance history, current understanding, and curriculum alignment. By analyzing interaction data, the system determines the student's strengths and weaknesses and delivers questions that are both relevant and suitably challenging for their level. This ensures continuous engagement and supports learning at an appropriate pace.

SQLBooster is also capable of **feedback delivery** and that is deeply intertwined with its personalization capabilities. When the students interact with the system, the LSA uses the information that is retrieved by the RAG tool to provide feedback that is relevant to the context. It will not only correct the mistakes but will also explain the concepts tailored to the current understanding levels of the students. This approach ensures that the feedback is not generic and that it is customized. It also directly addresses the immediate learning needs of the students and promotes a deeper understanding of SQL.

The system's **interactive user interface** is designed to ensure an engaging and seamless experience. Built using modern web technologies such as Next.js and asynchronous JavaScript, the interface allows for real-time communication through WebSocket integration. This architecture enables dynamic updates, such as displaying new questions and feedback instantly, without requiring page reloads. As a result, user actions generate near instant responses from the backend AI systems, maintaining the system's responsiveness and enhancing usability.

The **student dashboard** offers personalized insights into learning progress. It tracks various performance metrics and analyzes them to provide individualized feedback. Data from student interactions are stored in a MongoDB database and processed using AI-powered algorithms to evaluate performance trends. This allows the system to identify areas of strength and weakness and adjust future learning content accordingly. The adaptive learning insights offered through the dashboard align the difficulty and type of content with the student's evolving performance.

In parallel, the **lecturer dashboard** is designed to provide instructors with comprehensive analytics regarding student learning patterns, so that they can tailor the instruction material effectively. It allows the lecturers get an understanding about the common challenges faced by the class. It also helps the lecturers identify topics that will require additional instruction. SQLBooster also allow the lecturers to upload additional materials by the user-friendly interface to directly influence the curriculum. These materials are processed by the backend and incorporated into the student learning paths via the LlamaIndex framework.

SQLBooster also contains an **autonomous grading system** that evaluates student submissions intelligently and efficiently. Central to this grading system is the use of partial grading techniques, which uses the capabilities of OpenAI's GPT-4o to analyze the SQL queries submitted by the students. The system evaluates these submissions not only for correctness but also for syntactic accuracy, logical structure, and the adherence to the SQL best practices. For syntax correctness, which accounts for 40% of the total score, the system checks whether the query adheres to the standard SQL syntax. Queries with correct syntax receive full marks, while those containing errors are marked proportionally based on the severity and number of issues. Logical accuracy also contributes to 40% of the score and is determined by how effectively the query retrieves the correct result. Fully accurate queries earn full marks, while partially correct outputs are evaluated based on the percentage of correct data retrieved. Execution efficiency, weighted at 20%, evaluates the efficiency of the SQL query, considering factors such as the minimization of sub-queries, and optimal use of SQL functions. Queries that demonstrate best practices in query optimization will receive full marks. Partial marks will be given to less optimal queries. This grading approach helps to ensure that partial credits are awarded for partially correct answers.

Each level in SQLBooster consists of ten SQL questions. To pass an attempt, a student should at least receive 70% of the marks. This standard makes sure that the students have grasped the material adequately before moving to more complex topics. This method uses both performance metrics and improvements over time, and this allows the system to adapt the difficulty of future questions based on the performance history.

Even though the above features were described individually for clarity, they are interconnected through the user interface of SQLBooster. Figure 2 to Figure 11 represents the User Interface (UI) of SQLBooster.

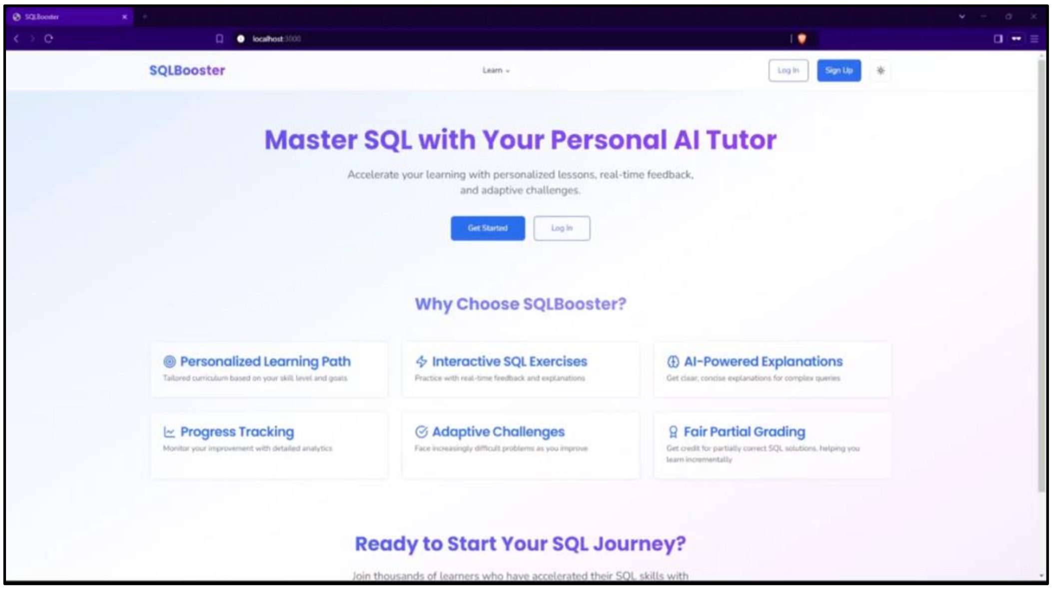


Figure 2. Initial UI - Light Theme

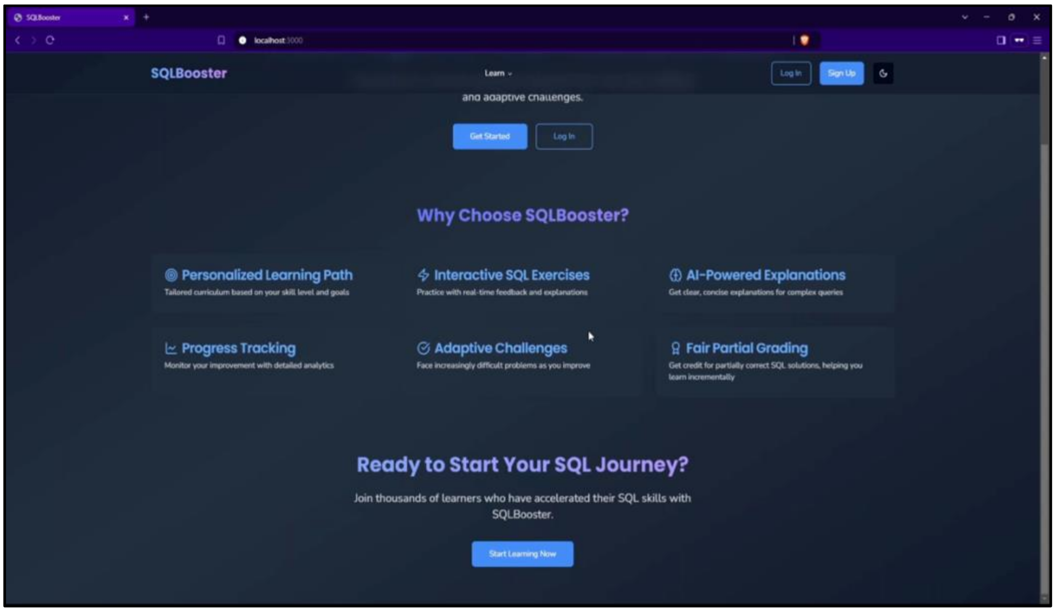


Figure 3. Initial UI - Dark theme

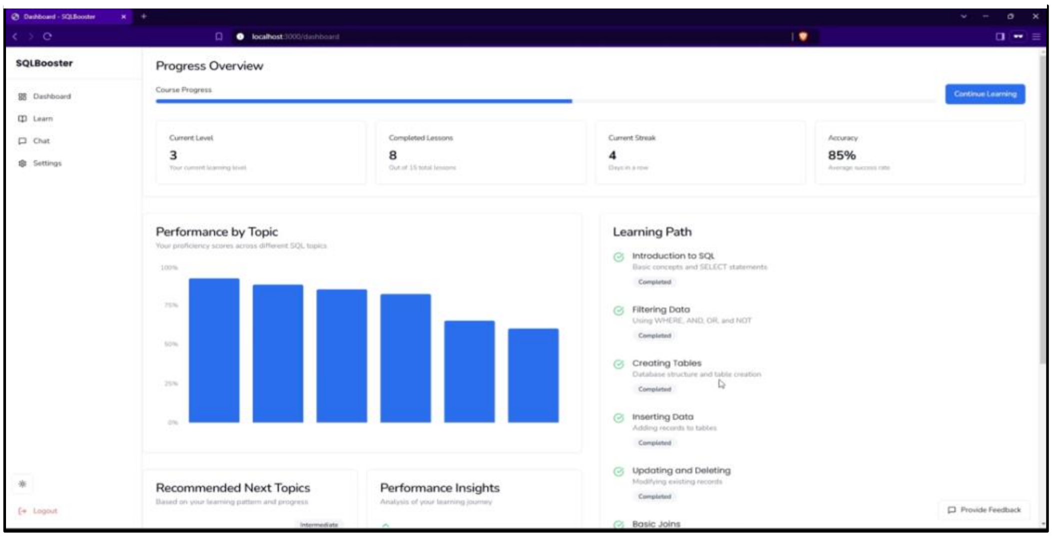


Figure 4. Student Learning Analytics Dashboard

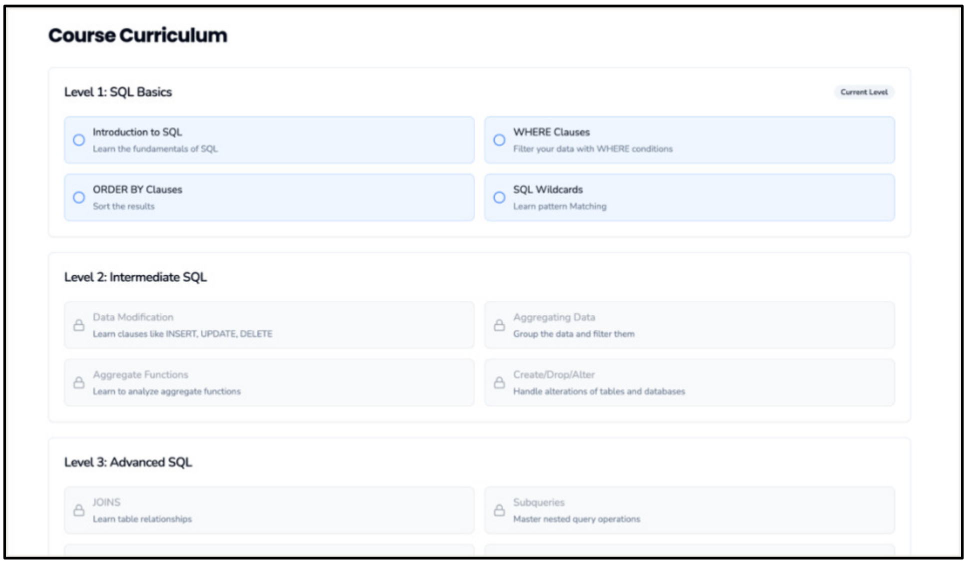


Figure 5. Course Curriculum

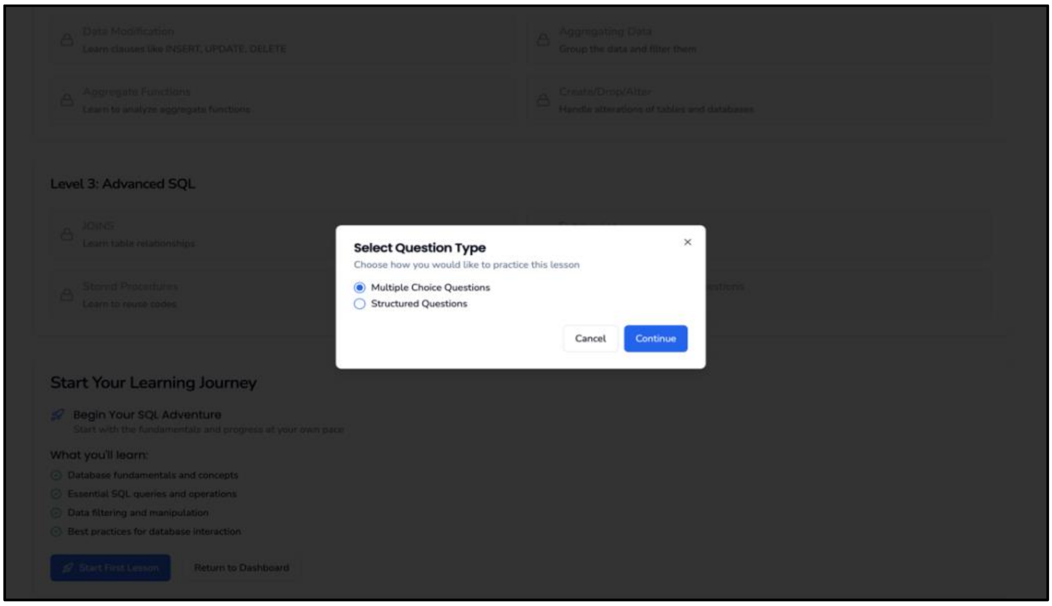


Figure 6. Select question type for question generation

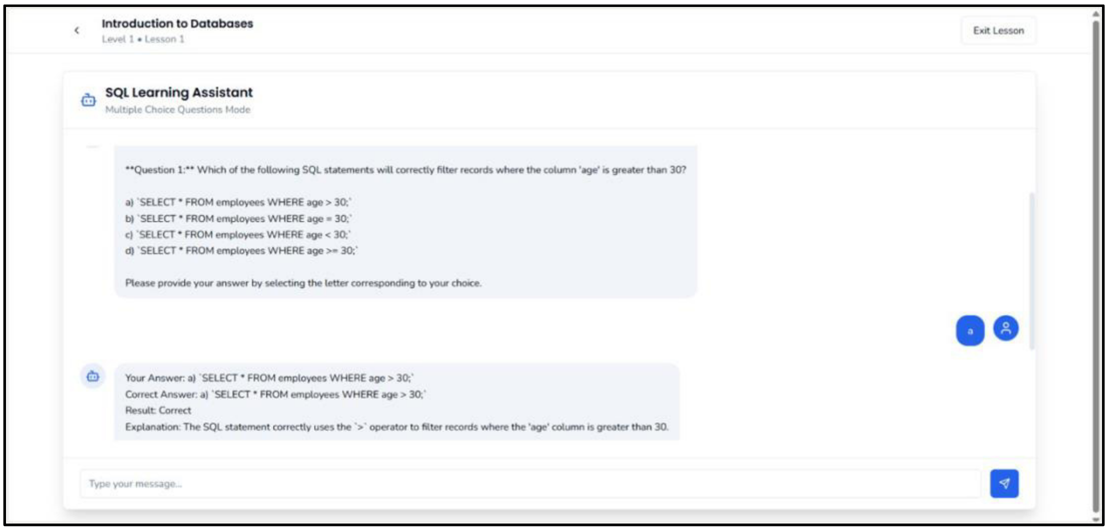


Figure 7. Personalized MCQ questions and Feedback

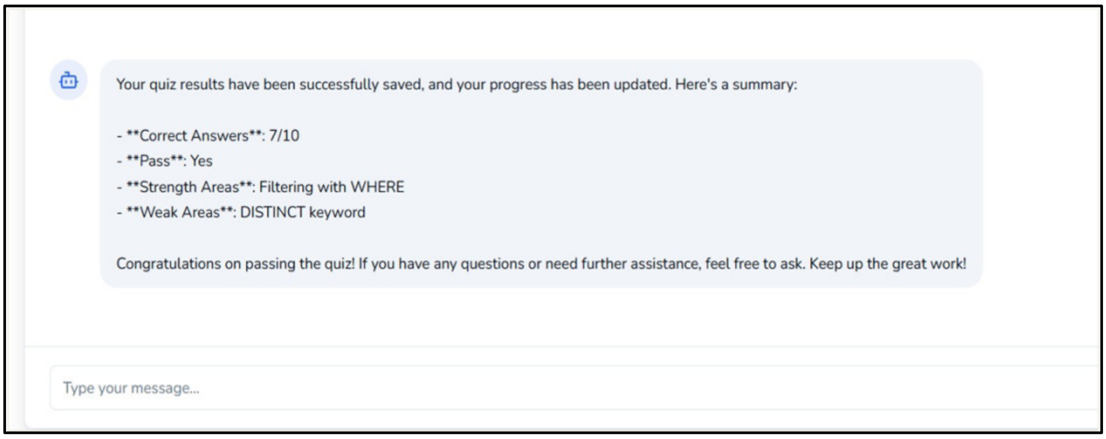


Figure 8. Final level grading

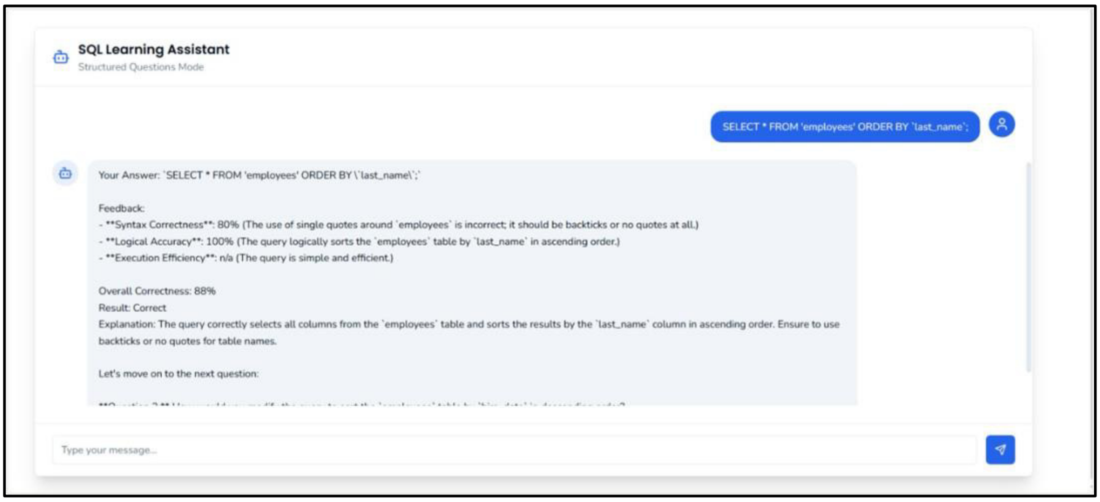


Figure 9. Tailored feedback and partial grading for structured questions

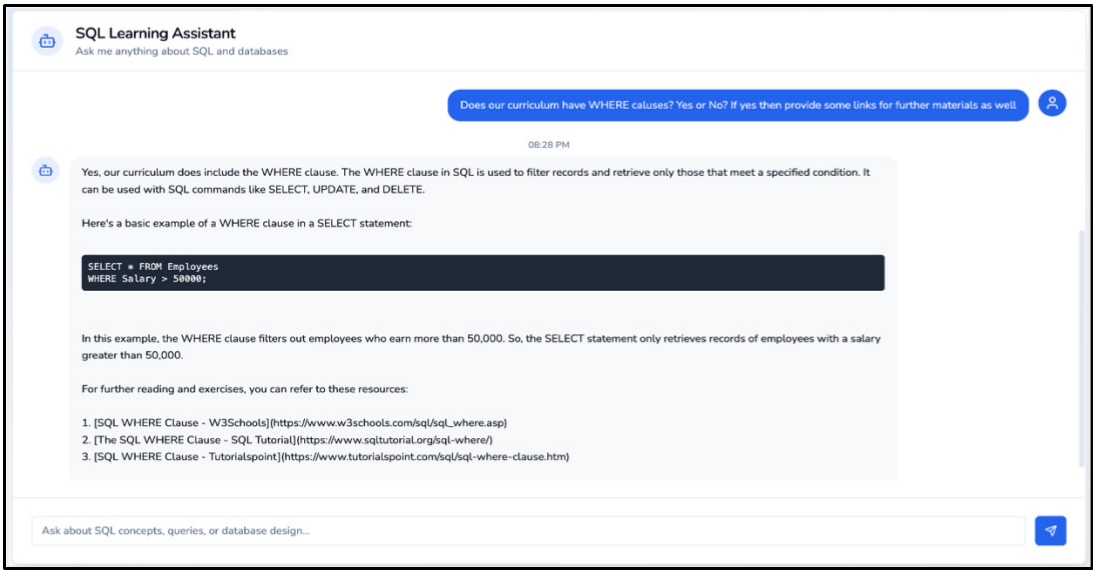


Figure 10. Curriculum based doubt clarification

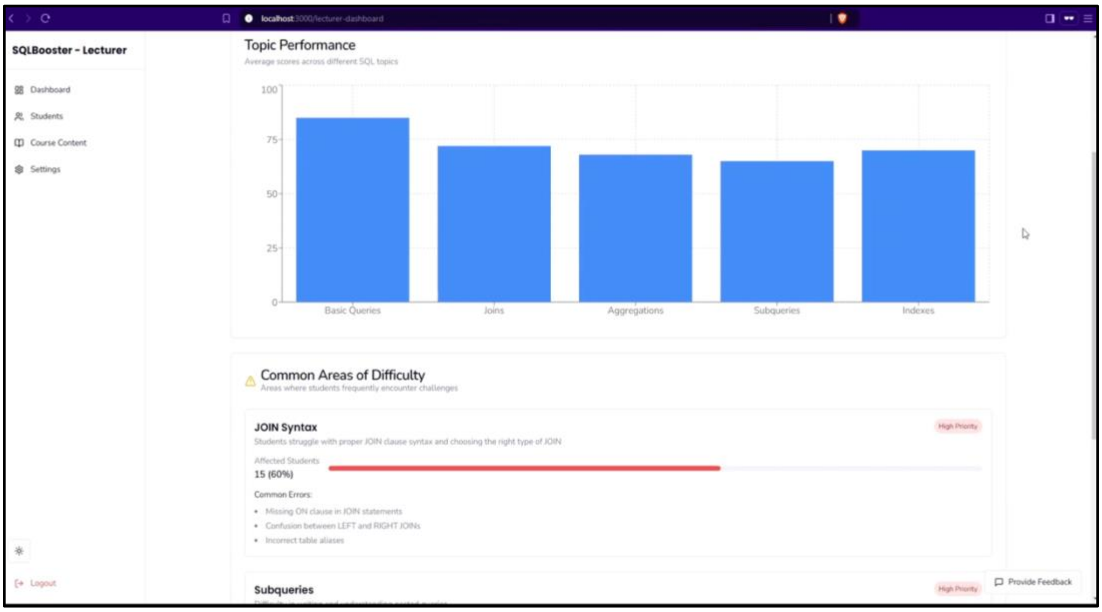


Figure 11. Lecturer Dashboard

Here, I did not include UI screenshots for standard pages like login, sign up, feedback sharing, and settings because these are common to most applications and is not unique to SQLBooster.

Curriculum Structure

The curriculum of SQLBooster is mainly categorized into three progressive difficulty levels. Each of these levels consist of four lessons. These are designed in a way to adapt dynamically to the individual learning progress through agent-driven personalization.

Level 1, titled SQL Fundamentals, introduces students to the foundational concepts of SQL. The first lesson focuses on basic SQL operations, including the SELECT statement, use of DISTINCT, aliases, and selecting specific columns. The second lesson introduces filtering techniques using the WHERE clause, covering logical operators such as AND, OR, and NOT. Lesson three covers sorting query results using the ORDER BY clause, while the fourth lesson explores the use of SQL wildcards for flexible pattern matching.

Level 2, or Intermediate SQL, builds upon the foundational knowledge by covering more complex data manipulation and aggregation operations. Lesson five addresses data modification commands such as INSERT, UPDATE, and DELETE. Lesson six introduces data aggregation using GROUP BY and filtering aggregated results with HAVING. The seventh lesson provides an in-depth understanding of aggregate functions including MIN, MAX, and COUNT. Lesson eight shifts focus to structural operations, including creating, altering, and dropping tables and databases.

Level 3, labeled Advanced SQL, targets more complex SQL operations. Lesson nine explores various types of joins, including FULL JOIN, LEFT JOIN, RIGHT JOIN, and INNER JOIN, to enable multi-table data retrieval. Lesson ten introduces subqueries, guiding students through the use of nested queries for complex logic. Lesson eleven covers stored procedures. From lesson twelve onwards, students engage with assignment type practice questions that integrate knowledge acquired across previous lessons.

Initially, when a new student starts, they will begin with Lesson 1. This is not personalized as there is not yet enough student data to tailor the content. However, when the student progress and more performance data is collected, the personalization will start. More weight will be given to questions from the current lesson, and a few questions from previous lessons where the student showed weaknesses will also be added. After Lesson 12, students receive a personalized set of questions from the curriculum.

Technology Selection Rationale

When examining the technologies considered in SQLBooster, it is important to validate the selection of them. The technologies were decided based on areas such as cost-effectiveness, and performance.

In selecting a large language model, several options were evaluated. Initially, I considered to use OpenAI's o1 which is known for its advanced reasoning capabilities. However, the cost of this was high and it had slower response times ("OpenAI Pricing," n.d.). This made it less suitable for SQLBooster. Then, I evaluated GPT-4 Turbo. It had good performance but it was beyond my budget constraints. Then I explored Claude 3 Opus. This had higher analytical reasoning and extended context handling. However, the cost was higher than my budgetary constraints. Finally I decided to use **OpenAI's GPT-4o** due to its high performance, multimodal capabilities, and cost-effectiveness ("Context," n.d.).

For the backend framework, the decision was between Flask, Django, and FastAPI. Flask was initially considered due to its simplicity and wide adoption in web development. However, since Flask has a synchronous nature, this could lead to slower response times under heavy loads. Since SQLBooster has real-time interaction requirements, this made Flask less suitable (Rajandran, 2024). Then I considered using either Django or FastAPI. Django offers a comprehensive framework with built-in features like an ORM, authentication, and an admin panel. Thus, Django is a well-suited option for full-stack applications. On the other hand, FastAPI is optimized to build high-performance APIs. It supports asynchronous programming and automatic interactive documentation. Even though Django has introduced some async capabilities, it is still not fully asynchronous. Due to SQLBooster's real-time interaction needs, it was decided to use **FastAPI** due to its support for async operations and WebSockets (Mantosh, 2023).

The RAG tasks in SQLBooster required an efficient vector search mechanism. Initially, I considered using Elasticsearch because of its powerful full-text search capabilities. However, this was complex and resource-intensive ("altexsoft," n.d.). Then I decided to use either Redis Search or LlamaIndex. Redis search has fast performance and cost-effectiveness, however its vector search capabilities are relatively new ("Redis," n.d.). Finally, **LlamaIndex** was selected because it specializes in vector search and this is crucial for the RAG tasks in SQLBooster. Furthermore, this option had a straightforward implementation, and was cost-efficient.

When selecting a database technology, MySQL, OracleSQL, and MongoDB were examined. Both MySQL and OracleSQL had robust features. However, their schema rigidity and higher operational costs made me consider MongoDB (MongoDB and Oracle Compared, n.d.;

"MongoDB," n.d.). Thus, I selected **MongoDB** due to its schema flexibility, scalability, and cost-effectiveness.

For the front-end technology, the initial candidates were React and Angular due to their widespread use and large developer communities. However, Angular was too heavy for the needs of SQLBooster. React alone needed additional configurations for routing and server-side rendering. Thus, I considered using **Next.js**. Due to the built-in features like server-side rendering, static site generation, and automatic code splitting available in Next.js, I decided to use this for building the frontend (Liyanage, 2024; Sahoo, 2024).

In conclusion, the technologies used in SQLBooster were carefully selected based on performance benchmarks, budget constraints, development simplicity, and alignment with the system's functional requirements. Each technology was chosen after a comparative analysis to ensure that the overall architecture remained efficient and suitable for personalized SQL education.

Part 2: Student Monitoring

In this section, the training of the head pose detection model, attentiveness detection model, and emotion detection model is outlined. Additionally, this includes a step-by-step guide on how these models work, details about emotion-based tension detection, and a high-level architecture diagram that outlines the overall system for student monitoring. Furthermore, this section contains an innovative approach that details how head pose detection and attentiveness detection is combined with emotion detection to effectively identify tension.

Training the Models

Head pose Model Training

Initially, training of the model started with the selection of a dataset from a repository that consisted of a wide array of images capturing various head movements (Gourier and Crowley, 2004; "Head Pose Image Database," n.d.). To ensure diversity and coverage of the dataset, personally collected data was also used. This combination of data sources aimed to provide a comprehensive and representative image set for training and testing of the head pose model. From these images, 2207 images were selected to cover a wide range of head positions and movements. The selection was done in an intentional way by ensuring that it would cover a range of possibilities and establish a fair and suitable training dataset. Here, the aim was to reduce bias by making sure there was diversity in demographic representation and head pose types. Additionally, a separate set of 262 images was selected to test the performance of the model.

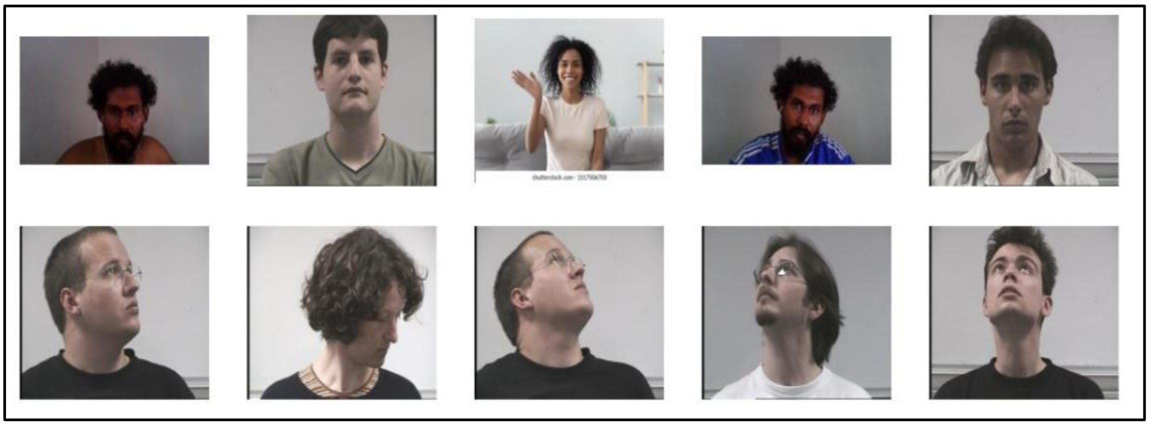


Figure 12. Head Pose Training Images

After preparing the dataset, various models were experimented to analyze the head movements. The aim here was to select the most suitable model for SQLBooster. As an initial step, I developed and trained a custom CNN model specifically for head pose classification. However, the model achieved relatively lower accuracy. As a result, I decided to explore the use of well-established pre-trained models to determine whether they could provide a stronger baseline for this task. Several models such as ResNet50, VGG16, MobileNetV2, and DenseNet121 were considered alongside the custom CNN. Each of these models were trained using a dataset for 50 epochs. This was done to ensure that they had sufficient opportunity to learn from the data. The following table summarizes the performance of each model. It examines how precisely and consistently each model could identify the head positions with and without focusing on specific movements, and their overall accuracy. This comparative analysis helped identify the models that were more effective and provided a foundation for selecting the best-suited model for further enhancement.

Table 2. Initial performance analysis of deep learning models for head movement analysis

Method	**No Focus Precision**	**No Focus Recall**	**No Focus F1-Score**	**Focus Precision**	**Focus Recall**	**Focus F1-Score**	**Overall Accuracy**
ResNet50	0.80	0.83	0.81	0.77	0.80	0.78	0.79
VGG16	0.84	0.86	0.85	0.83	0.85	0.84	0.85
MobileNetV2	0.88	0.89	0.88	0.87	0.88	0.87	0.88
Custom CNN	0.69	0.71	0.70	0.67	0.69	0.68	0.69
DenseNet121	0.91	0.91	0.91	0.90	0.91	0.90	0.91

From this analysis, it was clear that several pre-trained models significantly outperformed the custom CNN model. Among them, DenseNet121 achieved the highest overall accuracy of 91%. Even though this had a high performance, further experiments were conducted for DenseNet121 along with ResNet50, VGG16, and MobileNetV2 to further increase their performance. The aim here was to improve the performance metrics of all these models, including DenseNet121, and to determine if any of the other deep learning models could match or exceed the improved performance levels of DenseNet121.

The first experimental approach focused on adapting the models for a binary classification task. To achieve this, the architecture of each model was modified by simplifying the output layer to suit the binary classification requirements. Specifically, the original top classification layer of each model was removed to allow the addition of custom layers tailored for distinguishing between the two categories ‘focus’ and ‘not focus’. This approach used the feature extraction capabilities of the base models while customizing the final classification process. A Flatten layer was added to convert the multidimensional output of the convolutional layers into a one-dimensional array suitable for dense layer processing. Finally, a Dense output layer with two neurons was introduced, using a softmax activation function to produce probabilistic outputs corresponding to the two classes. This configuration enabled each model to make clear and interpretable predictions, which is appropriate for binary classification tasks. For DenseNet121, the above-mentioned modifications were implemented as shown in Figure *13*.

```
model = Sequential()
model.add(DenseNet121(include_top=False, input_shape=(224, 224, 3)))  #
DenseNet121 model without the top layer
model.add(Flatten())  # Flatten layer to connect the Convolutional layer and
Dense layer
model.add(Dense(2, activation='softmax'))
```

Figure 13: DenseNet121 - Experiment for binary classification

This experimental setup allowed for the evaluation of DenseNet121's performance in a binary context with minimal changes to its base architecture. The same modifications were applied to ResNet50, VGG16, and MobileNetV2 to ensure consistent conditions across all models.

In the second experiment, more advanced architectural modifications were introduced to further improve the classification accuracy. These enhancements involved the use of pre-trained weights and the addition of layers designed to refine feature extraction while minimizing the risk of overfitting. Each model was initialized with weights pre-trained on the ImageNet dataset, which is widely recognized for its diversity and effectiveness in general-purpose image classification tasks. This pre-training served as a strong foundation for visual feature recognition, allowing the models to bypass the initial stages of generic learning and focus on identifying task-specific patterns relevant to binary classification. To preserve the benefits of these learned features, all pre-trained layers were set to non-trainable during subsequent training. After the pre-trained layers, a Flatten layer was added to convert the multi-dimensional outputs into a one-dimensional array, preparing the data for dense layer processing. A Dense layer with 512 neurons and a Rectified Linear Unit (ReLU) activation function was included to help the model learn complex, non-linear features. To address overfitting, a Dropout layer with a rate of 0.5 was incorporated, which randomly disables a portion of the input units during each training iteration. This step encourages the model to learn more generalizable and reliable patterns. Finally, a Dense output layer with two neurons and a softmax activation function was used to produce probabilistic outputs corresponding to the two binary classes. This complete architecture enabled clear and interpretable predictions suitable for binary classification tasks.For DenseNet121, the above-mentioned changes were implemented as shown in Figure 14.

```
base_model = DenseNet121(input_shape=(224, 224, 3),  # Shape of our images
                         include_top=False,
                         weights='imagenet')

for layer in base_model.layers:
    layer.trainable = False

# Flatten the output layer to 1 dimension
x = layers.Flatten()(base_model.output)

# Add a fully connected layer with 512 hidden units and ReLU activation
x = layers.Dense(512, activation='relu')(x)

# Add a dropout rate of 0.5
x = layers.Dropout(0.5)(x)

# Add a final sigmoid layer for classification
x = layers.Dense(2, activation='softmax')(x)

model = tf.keras.models.Model(base_model.input, x)
```

Figure 14. DenseNet121 - Experiment for advanced binary classification

Similarly, the above-mentioned experimental approach was carried out for the other models ResNet50, VGG16, and MobileNetV2 as well. The same modifications were applied to them as well, so that it will be possible to test all the models under uniformly structured conditions. In both experiment 1 and experiment 2, each model was trained for 50 epochs. The enhanced DenseNet121 continued to outperform the other models and was therefore selected as the final head pose classification model used in SQLBooster. Further details about the results will be presented in the evaluation chapter.

Attentiveness Model Training

Initially, the training of the attentiveness detection model began by gathering images with either closed eyes or opened eyes. This dataset included images from publicly available sources as well as those collected personally. The dataset was diverse and represented different eye conditions in an effective manner.

To train the model, 1234 images were carefully chosen so that it will cover a range of closed and opened eye scenarios. This selection was done by considering areas such as various eye shapes and sizes to ensure that the model works well across diverse populations. In addition to above, a separate group of 218 images was selected to test the performance of the model. These images were used to check how well the model can identify new and unseen images.

Figure 15. Closed and Opened eye set for training attentiveness model

Next, it was needed to select a suitable model that would provide high accuracy when trained on this data. To achieve this, a custom Convolutional Neural Network (CNN) model was developed and evaluated for its performance. The architecture of this model was designed to efficiently process input images and extract meaningful features that would help in classifying whether the eyes were open or closed. The design choices for the CNN were aimed at achieving optimal feature extraction, reliable classification, and reduced risk of overfitting.

The model began with a Conv2D layer, which applied a set of filters to the input image to generate feature maps. These maps captured basic visual features such as edges and textures, which are important for distinguishing between open and closed eyes. This was followed by a MaxPooling2D layer that reduced the spatial dimensions of the feature maps, lowering computational complexity while retaining essential information. A second Conv2D layer was then added to extract more complex patterns, followed by another MaxPooling2D layer to continue simplifying the data while preserving critical features. To deepen the network's learning capacity, a third Conv2D layer was included, which enabled the model to capture finer details necessary for accurate eye state classification. A third MaxPooling2D layer followed, further reducing the dimensionality and preparing the data for the classification phase.

To mitigate overfitting, a Dropout layer was added next, which randomly set a portion of input units to zero during training. This regularization technique helped ensure the model did not become too reliant on specific features and encouraged more generalized learning. The output of the convolution and pooling layers was then passed through a Flatten layer, which transformed the multi-dimensional feature maps into a one-dimensional array. This allowed the model to transition from the feature extraction phase to the classification phase.

Following this, a Dense (fully connected) layer was used to interpret the extracted features and make a prediction about the eye state. To further support generalization and reduce overfitting, a second Dropout layer was included after this dense layer. Finally, the model concluded with an output Dense layer containing two units, corresponding to the two possible classes: open and closed eyes. A softmax activation function was used in this layer to convert the raw outputs into probabilities, providing a clear indication of the predicted class.

Figure 16 explains the structure and flow of the custom CNN model described above, using an example. Each layer is shown sequentially from the input layer at the top to the output layer at the bottom. This visual representation shows how the data transforms through each phase of the model. It also shows how the reduction in spatial dimensions and the focus on feature extraction and classification happens.

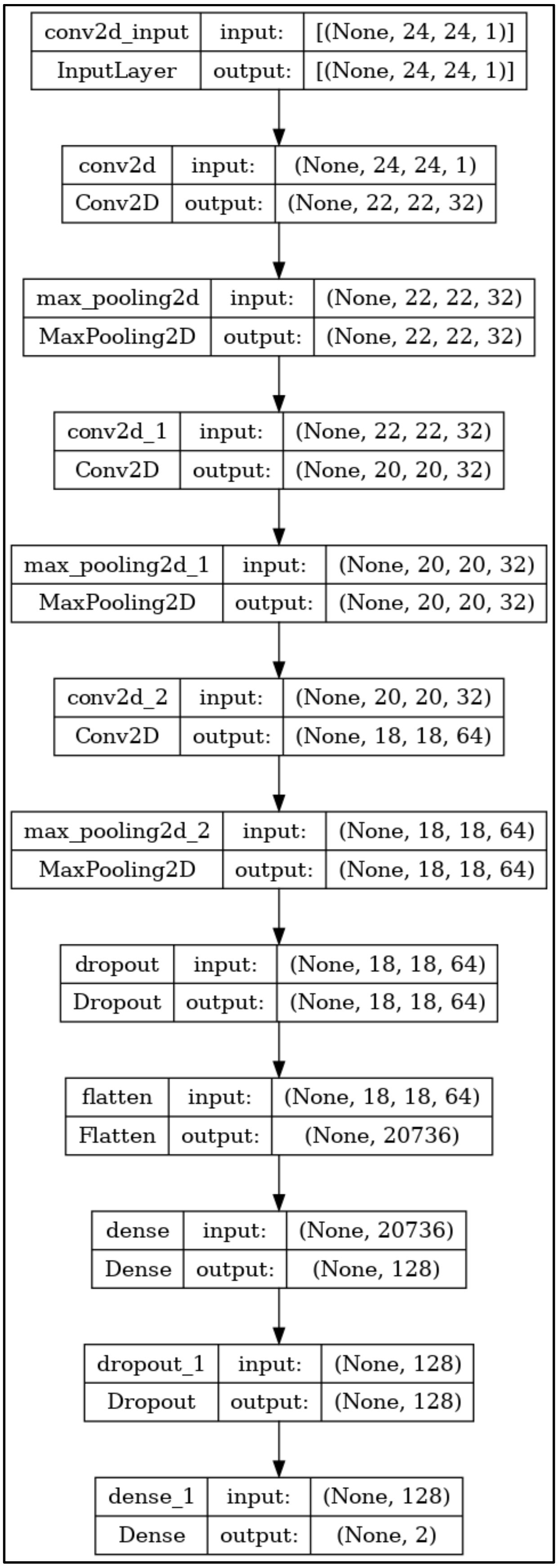


Figure 16. Custom CNN model

This custom model showed high accuracy during testing. Since the eye region occupies a relatively small area and the classification task of distinguishing between open and closed eyes is straightforward, the use of large-scale pre-trained models is not necessary. Therefore, the custom CNN architecture was considered sufficient and selected as the final model for attentiveness detection. Detailed results about the performance of the model will be provided in the Evaluation chapter.

Emotion Detection Model Training

Here, the development of the emotion detection model started with the selection of the 'fer2013.csv' dataset, which was used to train and evaluate various models for accurate emotion prediction ("Facial Expression Recognition Challenge data," 2019). This included a variety of facial expressions that were categorized into the seven emotion categories Anger, Disgust, Fear, Happiness, Sadness, Surprise, and Neutral. These emotion labels were shown in the dataset by using numerical values 0-6.

A primary challenge encountered during this stage was the imbalance among these emotion classes. To address this issue, data augmentation techniques were applied to the underrepresented classes in order to equalize their presence in the dataset. This involved enhancing the minority classes through image transformations such as padding, cropping, and brightness adjustments. These steps helped ensure that the model was exposed to a balanced representation of all emotion categories. The dataset was then divided into a training and validation set, with 90% allocated for training and 10% for validation. The decision to use a 90% training split, rather than the more commonly used 80%, was justified by the large dataset size of 62,923 images, which still provided a substantial validation set of 6293 images for effective model evaluation.

Then, it was required to select a model with high accuracy to detect the emotion. In order to this, the Deep Convolutional Neural Network (DCNN) model was considered first. The architecture of DCNN included layers such as Conv2D to extract detailed features from images, MaxPooling to reduce the spatial dimensions, Dropout layers to reduce the risk of overfitting, and BatchNormalization to keep the outputs of the network consistent.

The training of the model was conducted over 100 epochs with a batch size of 32. It used techniques such as Early Stopping to avoid overfitting and Reduce Learning Rate on Plateau to lower the learning rate when the model stopped getting better. This makes it easier to fine-tune and improve the performance of the model. Real-time data augmentation was implemented by

using an ImageDataGenerator to mix up the images during training. This helps the model to learn from a broader range of scenarios.
Later, the model was trained for an extended 150 epochs, which led to further improvements in accuracy. Further details about the results obtained will be explained in the evaluation chapter. Even though models such as simple CNN, VGGNet, and Mini-ResEmoteNet were considered, their accuracy was less when compared with this DCNN model. Thus, it was decided to use DCNN as the deep learning model to predict the emotion.

Emotion-based Tension Detection

Emotion to Tension Mapping

The DCNN model was only capable of predicting the emotion class. It was not capable of detecting the tension of the students based on the emotion. Thus, a novel approach was used to map emotional states to tension levels. This makes it possible to gain an understanding of how each emotional category can impact the cognitive and emotional responses of students. Each emotional state is linked to a specific level of tension, which influences how students interact with educational content.
There are mainly four tension categories when considering emotion to tension mapping. The first category is **Positive: Normal/Balanced**, which includes the emotions of happiness and neutrality. Emotions in this category is seen in optimal learning environments. When the students experience happiness, there is a high possibility that they will be motivated and engaged. This helps to enhance the cognitive functions such as memory and problem-solving. A neutral state suggests that the student has a calm and composed demeanour. This allows the students to focus better and process the information in an efficient way without any emotional distraction. In scenarios such as examinations or interactive learning sessions, these emotions can lead to higher performance levels and thus can result in more effective learning outcomes.
The second category is **Positive: Lower Tension**, which is associated with the emotion of surprise. When positively experienced, surprise can stimulate curiosity and increase a student's interest in the learning material. It often arises from new discoveries or unexpected positive feedback, which can make the learning experience more engaging. Lecturers can use this by incorporating novel instructional techniques or surprising elements into their teaching to maintain high levels of student engagement.
The third category is **Negative: Lower Tension**, which includes the emotion of sadness. Even though sadness is a negative emotion, the impact of it on tension is typically lower when compared with other negative emotions. It can slow down the cognitive processes and can

reduce the motivation of students. However, it doesn't usually cause significant disruption in cognitive capabilities. Students might withdraw somewhat from participation, but will still be able to follow along and understand the educational material. Recognizing negative lower tension can help the lecturers provide support or adjustments to learning approaches, such as offering additional help or more time to complete tasks. This can also suggest that it might be good to consult the student.

The final category is **Negative: Higher Tension**, which encompasses emotions such as anger, fear, and disgust. These emotions are usually associated with high levels of stress and anxiety. This can severely impact the decision-making and memory functions. High tension can divert the student energy from cognitive to survival mechanisms and thus, it can impact learning in a negative way. If negative high tension is observed, lecturers can make immediate interventions, such as trying to calm the student and consider about the student's mental state.

Theoretical and Neuroscientific justification for Emotion to Tension Mapping

The rationale for mapping specific emotions to their respective tension categories is grounded in both psychological theory and neuroscientific evidence.

Happiness is closely related to increased levels of neurotransmitters such as dopamine and serotonin, which plays a major role in mood regulation and cognitive functions (Baixauli, 2017). High dopamine levels result in increased motivation and reward-seeking behaviour (Bromberg-Martin et al., 2010). Serotonin contributes to feelings of well-being and focus. These neurochemical changes can improve the learning and processing abilities (Cowen and Sherwood, 2013). These can lead to better memory and problem-solving skills. In educational settings, happiness can result in increased student engagement, motivation, and academic achievement. Therefore, happiness is categorized under the Positive: Normal/Balanced tension category.

Neutral emotional states are usually characterized by calmness and composure. They provide a stable platform for learning by minimizing emotional distractions. This emotional neutrality makes better concentration and efficient information processing possible because cognitive resources are not diverted by emotional responses. In educational settings, having a neutral emotional environment can result in increased student engagement and academic achievement (Cohen et al., 2016). Hence, the neutral emotion is also classified under the Positive: Normal/Balanced tension category.

Surprise can result in cognitive and physiological responses that vary based on whether the surprise is positive or negative. Positive surprise, such as unexpected good news, heightens the alertness for a moment but often results in quick emotional regulation. This leads to excitement or curiosity rather than prolonged tension. This response results in cognitive flexibility and engagement and thus, can be considered to be in a positive lower-tension state. When negative surprises are considered, these may cause due to reasons like sudden bad news. Thus, this will result in an immediate stress response, activating heightened physiological arousal, increased heart rate, and muscle tension. Thus, this may result in a higher-tension state due to the body's fight-or-flight reaction (Noordewier et al., 2016). Thus, this type of surprise aligns more closely with Negative: Higher Tension.

Sadness is typically associated with reduced physiological arousal, decreased activity levels, and introspection. This results in a reflective state that helps the individuals process emotions and adapt to challenges (Robinson and Demaree, 2009). Thus, this can result in negative lower tension. However, under extreme conditions, sadness can become overwhelming. This can lead to physiological stress responses such as increased blood pressure and agitation. Prolonged sadness can even cause heightened distress, restlessness, and irritability (Wu et al., 2017). Thus, in this condition sadness may affect in negative higher tension.

Both **anger** and **fear** are high-arousal emotions and can impact both the cognitive and physiological responses. Both emotions activate the body's fight-or-flight response, resulting in elevated heart rate, increased blood pressure, and muscle tension. These physiological changes redirect mental energy away from cognitive tasks and toward immediate survival-oriented responses. Anger, in particular, can impair rational thinking and decision-making due to reduced activity in the prefrontal cortex, which is responsible for self-regulation and higher-order reasoning (Armfield, 2006; Blair, 2012). Thus, both anger and fear are classified under Negative: Higher Tension.

Disgust, although physiologically distinct from fear and anger, is a strong negative emotion that triggers both cognitive and physiological reactions. It heightens the attention and memory for unpleasant stimuli and this increases the mental strain. Physiologically, it causes facial expressions like nose-wrinkling and sensations of nausea. It does not increase heart rate like fear or anger, but its strong instinctive reactions can create discomfort and tension. Therefore, disgust is also mapped to the Negative: Higher Tension category (Chapman and Anderson, 2012).

Handling Ambiguous and Context Dependent Emotions

Some emotional categories, such as surprise and sadness, require contextual interpretation rather than fixed classification. While surprise is generally mapped to Positive: Lower Tension, this is only valid if the surprise is positive. If it is a negative surprise, it results in negative higher tension. Similarly, even though I have mapped sadness as a negative lower tension, under extreme conditions this can result in negative higher tension. The emotion-based tension detection model can handle this scenario in the following way.

The emotion detection model does not analyze the entire video of the assignment period as a whole. It will process the video frame by frame by treating each frame as an individual image for analysis. As a result, different frames within a session can have various emotions detected. Usually, being surprised is short-lived compared to other emotional states. When the model detects surprise, even though it is mapped as a positive lower tension, it cannot be immediately considered as true because this is only valid if the surprise is positive. Therefore, the other frames should be examined to determine whether most emotions are happy or neutral, or if they lean more towards negative emotions like anger, disgust, or fear. If the majority of frames show negative emotions, we can say that the detected surprise is negative. Then this will result in a negative higher tension in students.

When sadness is considered, even though under extreme conditions tension can be changed to a negative higher tension, in the context of SQLBooster, the aim is to check if any actions need to be taken by the instructors. Since both the scenarios will mean that the instructor intervention might be needed, this will not be an issue in SQLBooster.

Step by Step Evaluation of Separate Models

In this section, the focus is on understanding the function of head-pose detection model, attentiveness detection model, and emotion-based tension detection model separately.

Head Pose Detection Model

The head pose detection functionality in SQLBooster is designed to evaluate student focus by analyzing the head movements during a video recorded session. The system follows a structured pipeline to assess whether the student maintains focus throughout the session.

The process begins with **initialization**, where the video file is accessed using OpenCV's VideoCapture module. In parallel, Haar cascade classifiers are loaded for face and eye detection, along with the pre-trained deep learning model used for head pose estimation.

Next, during the **frame-by-frame preprocessing stage**, the system iterates through each video frame. Each frame is converted to grayscale to reduce computational complexity while preserving the necessary features for accurate detection. Haar cascades are used to identify faces and eyes in each frame.

For **focus assessment**, frames in which no face or eyes are detected are automatically labelled as “not focus.” For the remaining frames that contain detected faces, the pre-trained head pose model is applied to classify the student's state as either “focus” or “not focus,” based on the orientation of the head.

The system then proceeds to **output annotation**, where each frame is visually annotated with the predicted focus status and corresponding bounding boxes around detected regions. These annotated frames are written back to the output video file, creating a visual record of the student’s focus states throughout the session.

Finally, **engagement metrics** are calculated by analyzing the ratio of “focus” versus “not focus” frames. If more than 20% of the frames are classified as “not focus,” the overall session is labelled as “not focused.” Otherwise, it is labelled as “focused.”

Attentiveness Detection Model

The functionality of SQLBooster’s attentiveness detection model is to design and assess the attentiveness of a student by analyzing their eye states throughout a video session. The model follows a structured sequence of operations to classify and evaluate student engagement based on eye activity.

The process begins with **initialization**, where the video file is loaded using OpenCV’s VideoCapture, and a video writer is configured to save the output video. Haar cascade classifiers are loaded for face and eye detection, and the pre-trained deep learning model is loaded for attentiveness detection.

During the **frame preprocessing phase**, the system iterates through each frame of the video. Each frame is converted to grayscale so that it will be possible to focus on the relevant details necessary for accurate eye detection. Haar cascades are then applied to detect the faces, along with left and right eyes. Frames in which no face or eyes are detected are excluded from attentiveness evaluation.

In the **eye state classification step**, the detected eye regions are passed to the deep learning model, which classifies each eye as either "Open" or "Closed." Based on this, the system annotates each frame with the corresponding label and writes the result back into the output video.

To determine overall attentiveness, the model proceeds to **status calculation**, where it computes the percentage of frames in which the eyes were classified as "Closed" and "Open." If more than 25% of the frames show closed eyes, the video is labelled as "not active." Otherwise, it is considered "active."

Finally, the **overall engagement status** for the video is saved in a global dictionary, allowing the result to be accessed later as part of the comprehensive tension detection workflow within SQLBooster.

Emotion-based Tension Detection Model

The functionality of SQLBooster's emotion-based tension detection model is designed to analyze the emotional state of a student and to associate these states with corresponding tension levels during a video session. The process follows a systematic sequence of operations.

The first stage involves **initialization**, where the pre-trained deep learning model for emotion classification is loaded along with its associated weights from a stored JSON file. OpenCV's VideoCapture is used to read the video file, and a video writer is used to save the output video. In parallel, a Haar cascade is loaded for the face detection within the video frames.

During the **frame preprocessing phase**, the model iterates through each frame of the video. Each frame is converted to grayscale to simplify the data and enhance facial feature detection. The Haar cascade is applied to detect faces in each frame, and only frames with detected faces proceed to the emotion classification stage.

In the **emotion classification step**, each detected face is passed to the deep learning model, which predicts the corresponding emotion. The identified emotion is then mapped to a predefined tension level. The categories used are: Positive: Normal/Balanced, Positive: Lower Tension, Negative: Lower Tension, and Negative: Higher Tension.

The model then proceeds with **output annotation**, where each frame is labeled with both the detected emotion and its associated tension level. An annotation color scheme is applied based on the intensity of the tension level, allowing for visual differentiation. These annotated frames are written to the output video file to produce a visual record of the emotional and tension states throughout the session.

Finally, in the **engagement metrics phase**, the model counts the number of frames corresponding to each tension level. Based on this distribution, it identifies the dominant emotional state and the overall tension level of the video. If the combined proportion of frames classified as Negative: Higher Tension and Negative: Lower Tension exceeds a predefined threshold, the session is labeled accordingly, indicating the need for potential instructor intervention.

High Level Architecture

Figure 17 represents the architecture diagram for student monitoring of SQLBooster. It mainly makes use of three models namely the head-pose detection model, attentiveness detection model, and the emotion-based tension detection model. All these models start by extracting frames from a video clip. These frames are first processed to segment and pass through the respective models. Each model will have its own configuration and pre-trained weights. This helps to ensure specialized analysis based on their unique functions.

To make sure that the assessment of the tension states of the students are accurate, SQLBooster combines the head-pose model, attentiveness detection model, and the emotion-based tension detection model. Initially, the head-pose detection model will determine whether the student is focusing on the screen. This is done by analyzing the proportion of the frames where the student is not focused. If less than 20% of the frames are categorized as 'not focused', the remaining 'focused' frames will be forwarded to the attentiveness detection model. Then, if more than 75% of the frames are labelled as ‘active’, then the active frames will be sent to the emotion-based tension detection model. This step is important because processing ‘not focused’ and ‘not active’ frames can lead the emotion-based tension detection model to interpret emotions in an inaccurate way. Then the emotion-based tension detection model will process these focused, active frames to classify emotions into specific tension categories. This can provide important information about the students such as whether they are stressed.

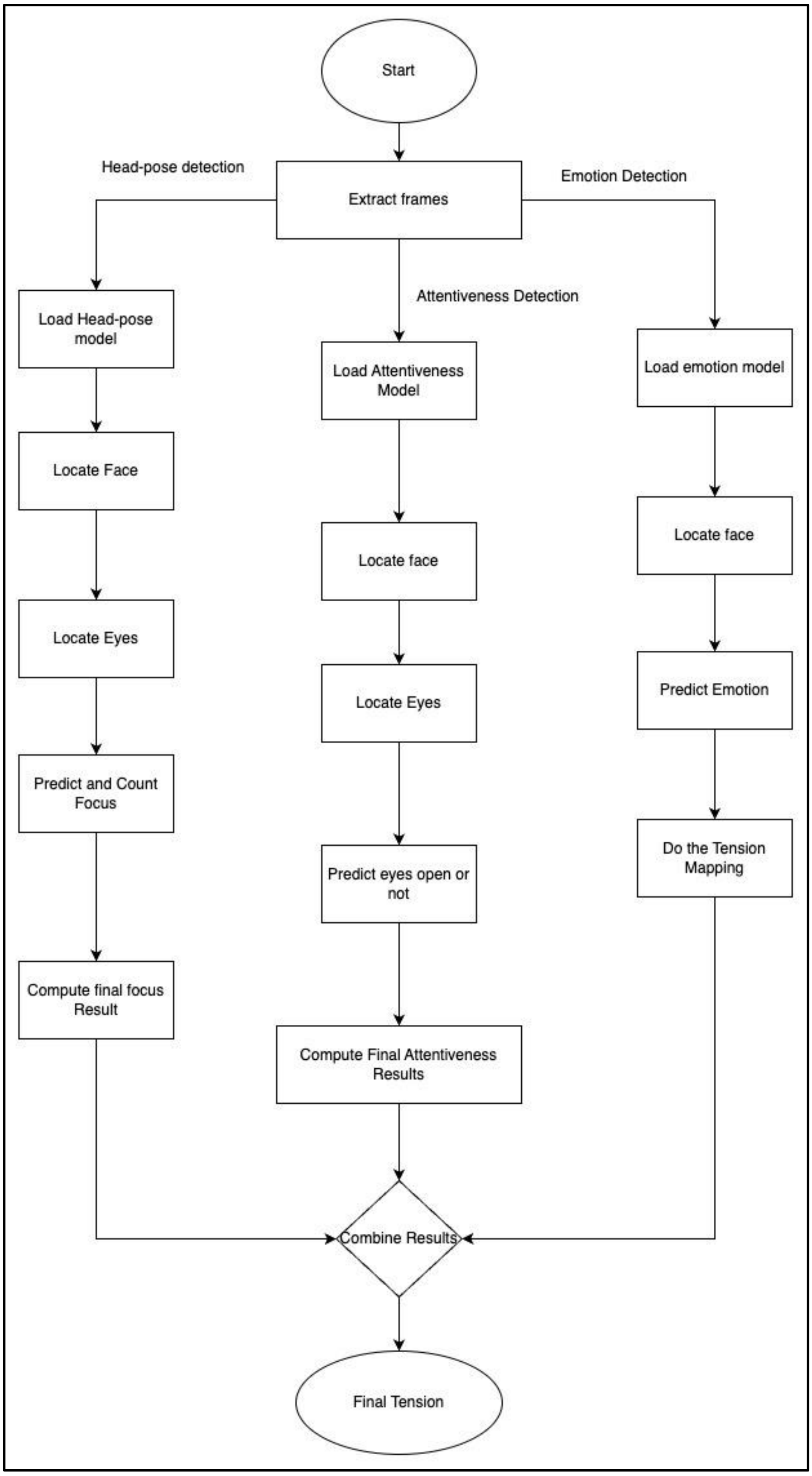


Figure 17. Architecture Diagram for student monitoring

3.3. Summary

This chapter presented the methodology used to develop SQLBooster explaining how techniques such as LLMs, RAG, LlamaIndex, and FastAPI can be used to deliver personalized learning experiences, real-time and adaptive feedback, partial grading, and interactive dashboards. A main element of SQLBooster's methodology is its student monitoring system. It uses three deep learning models namely the head pose detection model, attentiveness detection model, and emotion-based tension detection model. Since the aim here is to improve the SQL skills of the undergraduate students, it can be said that SQLBooster follows the pragmatist philosophy and data science methodology.

The innovative emotion-based tension detection model analyzes the emotions of the students to determine their tension levels. The tension levels are categorized as "Positive: Normal/Balanced", "Positive: Lower Tension", "Negative: Lower Tension", or "Negative: Higher Tension". SQLBooster first confirms that the student is focused by using the head pose detection model, and that the student is active (paying attention) by using the attentiveness detection model, before analyzing emotions. This helps to reduce errors that maybe caused due to unfocused head poses and inattentiveness. Overall, this methodology assists in providing a personalized educational experience and also helps in understanding and responding effectively to the tension states of the students.

CHAPTER 4: EVALUATION AND RESULTS

4.1. Introduction

The aim of this evaluation chapter is to assess the effectiveness of SQLBooster in enhancing SQL learning among undergraduate students. It examines how well SQLBooster meets its educational objectives by focusing on improvements in SQL proficiency and student engagement. The foundation of the evaluation is based on research related to student monitoring, adaptive learning systems, feedback mechanisms, and partial grading. These are key components that have been incorporated into the design of SQLBooster. The system achieves advanced student monitoring through its emotion-based tension detection model. This chapter aims to delve into details regarding the evaluation objectives, methodologies, results, and discussions on the effectiveness of SQLBooster. Finally, it concludes with a summary of the key findings related to the evaluation.

4.2. Evaluation Objectives

The evaluation of SQLBooster mainly focuses on three main objectives that are designed in order to assess its impact on SQL learning. These objectives play a major role in determining the overall effectiveness of the system and in identifying areas for future improvement.

The first objective is to assess the **accuracy of the monitoring features**, with a particular focus on the emotion-based tension detection model integrated within SQLBooster. This includes an evaluation of the precision, recall, and overall accuracy of the emotion detection, attentiveness detection, and head pose detection models.

The second objective is to evaluate the **usability of SQLBooster**, which involves examining the ease of use, interface design, and overall user experience. This part of the evaluation aims to determine how intuitively the students can navigate and use the system.

The third objective is to **evaluate the learning impact** of SQLBooster. This involves analyzing the initial impressions of students and the recognized impact of SQLBooster on their learning. The aim is to explore how effectively SQLBooster contributes to enhancing the SQL skills of undergraduate students.

Together, these objectives form the basis of a comprehensive evaluation strategy that supports the assessment of SQLBooster’s performance.

4.3. Evaluation Methodology, Results, and Analysis

This section outlines the evaluation methodology, presents the results obtained from assessing SQLBooster, and provides an analysis of those findings. The primary focus is on the performance of the monitoring models, as well as the usability and learning impact of SQLBooster through features such as personalization and partial grading.

4.3.1. Assessment of the effectiveness of the Monitoring Features

Here, the aim is to evaluate the effectiveness of the monitoring features of SQLBooster. These assessments are important to quantify how well the models Head-Pose Detection, Attentiveness Detection, Emotion Detection, and Emotion-based Tension Detection, meet the design objectives. To evaluate the performance of these models, several key metrics were used.

Precision is defined as the ratio of correctly predicted positive observations to the total predicted positives. This can be considered as a measure of the accuracy of the model in identifying only the relevant samples. Precision can be represented by using the following formula.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

Equation 1: Precision

Recall (Sensitivity) is the ratio of the correctly predicted positive observations to all actual positives. This is used to assess the ability of the model to identify all the relevant instances. Recall can be represented by the formula given below.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

Equation 2: Recall

F1-score provides the weighted average of Precision and Recall. This score takes both the false positives and the false negatives into account. It is mainly useful when the class distribution is not even. The F1-score can be calculated as follows.

$$F1\ score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Equation 3: F1 Score

Specificity measures the proportion of actual negatives that are correctly identified as such. This helps to assess the ability of the model to accurately identify and negate false alarms. Specificity is calculated by using the following formula.

$$Specificity = \frac{True\ Negatives}{True\ Negatives + False\ Positives}$$

Equation 4: Specificity

Overall accuracy represents the proportion of correctly predicted instances (both positive and negative) out of the total number of observations. This is used to measure the overall effectiveness of the model. The formula is as follows.

$$(Overall\ Accuracy) = \frac{True\ Positives + True\ Negatives}{(Total\ Observations)}$$

Equation 5: Overall Accuracy

With these performance metrics defined, the below sections present and analyze the evaluation results of the monitoring components in SQLBooster.

Head Pose Detection Model

After conducting experiments to improve the performance metrics of the head pose models as described in the methodology chapter, this section delves into the evaluation of these. Here, the aim is to observe whether there was an increase in accuracy after performing the experiments, and to determine if the enhancements have allowed any of these models to match or surpass the high-performance levels of DenseNet121. This analysis was conducted in order to choose the most optimal model for accurate head pose detection within SQLBooster.

The results of the performance evaluation for the head-pose model are presented in the table given below. This shows the precision, recall, F1-score, and overall accuracy for each model in scenarios with and without focus.

Table 3. Performance metrics of head-pose detection model

Method	No Focus Precision	No Focus Recall	No Focus F1-Score	Focus Precision	Focus Recall	Focus F1-Score	Overall Accuracy
VGG16	96%	94%	95%	93%	96%	94%	95%
ResNet50	87%	86%	86%	86%	86%	86%	86%
MobileNetV2	92%	98%	95%	98%	92%	95%	95%
DenseNet121	99%	100%	100%	100%	99%	100%	100%

Table 3 shows that DenseNet121 have significantly outperformed the other models in all the evaluated metrics. It was able to achieve perfect F1-scores in both focused and non-focused settings. Furthermore, this showed perfect recall in non-focused settings and perfect precision in focused settings. This shows its exceptional capability to identify correct head poses without errors. The 99% to 100% precision range further highlighted its ability to minimize false positives. This ensured highly accurate pose predictions.

In comparison, even though VGG16 and MobileNetV2 showed good results, they could not reach the consistency and the peak performance of DenseNet121. ResNet50 showed the lowest performance among the model with a uniform score of around 86% across all the metrics. Thus, it was decided to use DenseNet121 in SQLBooster.

Classification Report:

	precision	recall	f1-score	support
focus	1.00	0.99	1.00	127
not focus	0.99	1.00	1.00	135
accuracy			1.00	262
macro avg	1.00	1.00	1.00	262
weighted avg	1.00	1.00	1.00	262

Figure 18. Head-pose classification report

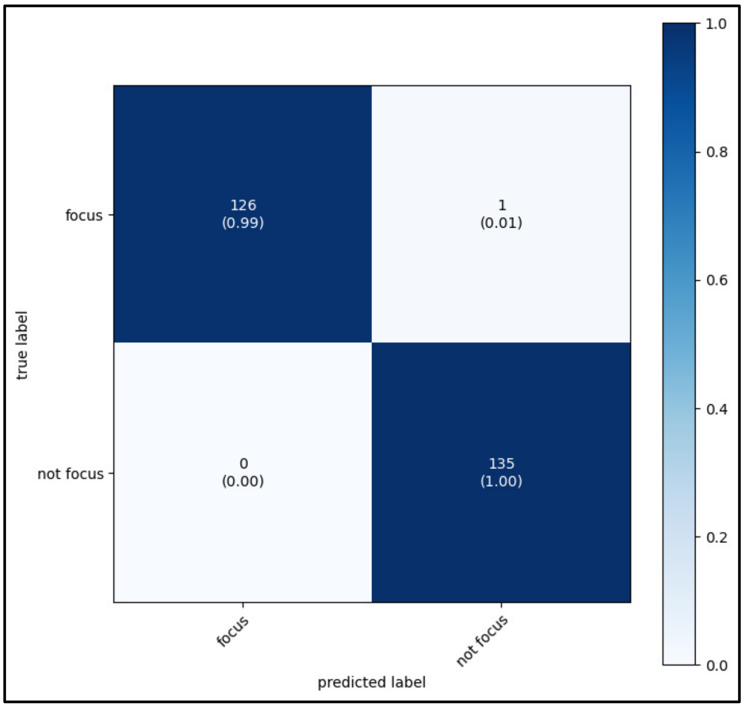


Figure 19. Head pose Confusion Matrix

Figure 18 and Figure 19 visually represents the performance metrics and further validate the superior capabilities of DenseNet121. The classification report (Figure 18) shows that there is an exceptional balance between the precision and recall (F1-score = 1). This helps to ensure that DenseNet121 can accurately identify and classify both the focused and non-focused head poses with minimum error. Furthermore, the confusion matrix (Figure 19) shows that out of 127 instances labeled as 'focus', 126 were correctly identified. It also shows that all 135 instances labeled as 'not focus' were accurately classified. This confirms the ability of the model to effectively discriminate between the two classes without getting any false negatives. The single instance of misclassification shows the high precision and the rarity of false positives of the model.

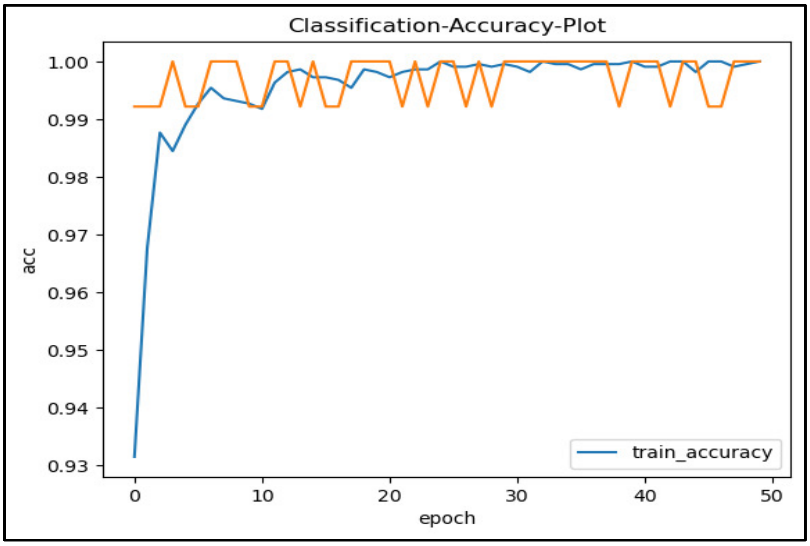


Figure 20. Head-pose classification accuracy plot

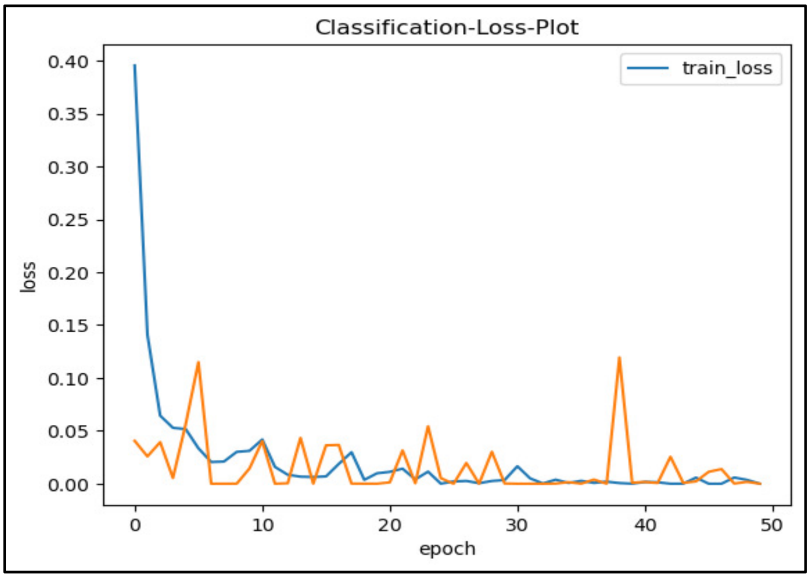


Figure 21. Head pose classification loss plot

Figure 20 and Figure 21 provide deeper insights into the training dynamics and the stability of DenseNet121 model over multiple epochs. These figures help to show the performance at a single threshold as well as the consistency and reliability of the model throughout the learning process.

Figure 20 represents the classification accuracy of the DenseNet121 during training. The plot shows that the accuracy quickly rises to near-perfect levels within the initial epochs and maintains a high level by oscillating around 99% to 100%. This high accuracy level across the training epochs shows the robust ability of DenseNet121 to generalize from the training data to similar unseen data.

Figure 21 shows the loss during training the model and provides a complementary view to the accuracy plot. Initially, the loss decreases in a sharp manner and this indicates a rapid learning and effective adjustment to the training data. As the epochs progress, the model shows minimal loss with occasional spikes that quickly resolve. This shows the resilience of the model to overfitting and its capacity to recover and adapt to the variations in the data.

Together, these plots highlight the training stability and efficient learning curve of the model. The consistent high performance across accuracy and loss metrics reassures that DenseNet121 is both accurate and dependable over extensive uses. This makes it an ideal choice for the use within SQLBooster.

The integration of the DenseNet121 model for head-pose detection in SQLBooster plays a major role by serving as a foundational input for the innovative emotion-based tension detection model. The accuracy of the head-pose data ensures that the analysis of the student tension levels is based on reliable and precise engagement metrics. This precise detection is very important as it directly influences the efficiency of the emotion-based tension detection model. Thus, having an accurate model, will help to accurately assess and respond to the tension states of the students, and in turn this will be helpful to address any challenges that may occur due to the emotional distress of the students.

Attentiveness Detection Model

This section evaluates the effectiveness of the attentiveness detection model in accurately identifying whether eyes are opened or closed. After the development of the custom CNN model explained in the methodology chapter, here the idea is to observe the improvement in model performance across various training epochs. Here, the model configuration that optimally balances the performance and computational efficiency is selected.

After implementing the custom CNN model, the accuracy was tested over a series of training epochs to determine the most effective setup for the attentiveness detection task. The results of these experiments are given in the following table.

Table 4. Training Results of Custom CNN Model - Attentiveness Detection

Model Architecture	Epochs	Accuracy
CNN	5	0.96
CNN	20	0.97
CNN	50	0.97
CNN	100	0.97

As shown in Table 4, the accuracy of the custom CNN model remains at 0.97 after 20 epochs. Extending the training to 50 and 100 epochs did not result in any improvement in accuracy. This shows that the model is capable of efficiently capturing the relevant features necessary for this classification task within the first few training phases.

However, it was decided to consider 100 epochs. This is because the model trained for only 20 epochs did not perform well in real-world settings, even though it showed high accuracy in initial tests. The extended training period allowed the model more time to learn and refine its understanding of the features. Thus, I proceeded with the 100-epoch model, and the following classification report and confusion matrix provide insights into its performance.

```
Classification Report
              precision    recall  f1-score   support

  Closed Eye       0.97      0.97      0.97       109
    Open Eye       0.97      0.97      0.97       109

    accuracy                           0.97       218
   macro avg       0.97      0.97      0.97       218
weighted avg       0.97      0.97      0.97       218
```

Figure 22. Classification model - Attentiveness Detection

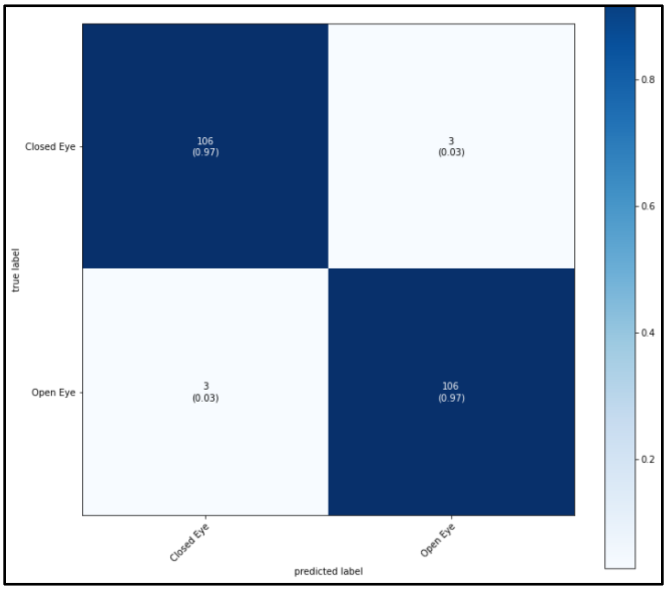


Figure 23. Confusion Metrix - Attentiveness model

Figure 22 shows metrics such as precision, recall, and F1-score of the model for both closed and opened eye states. Both the states achieved a precision, recall, and F1-score of 0.97. This high level of accuracy in all of these metrics shows the capability of the model to identify and classify each state with minimal error in a reliable manner. Such consistency is important for applications which need accurate attentiveness detection.

Figure 23 further shows the performance of the model. It shows the number of true positives and negatives, as well as the false positives and negatives. For both the closed and opened eyes, the model identified 106 out of 109 instances correctly. Only 3 instances were misclassified in each category. This represents a high true positive rate and a very low false positive rate. This confirms the effectiveness of the model in distinguishing between open and closed states.

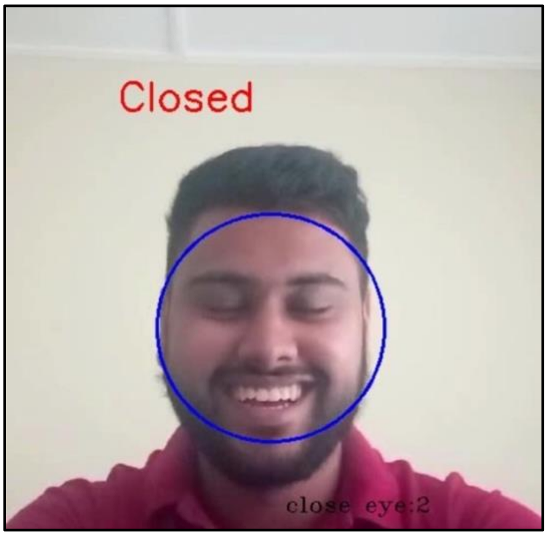


Figure 24. Closed Eye State

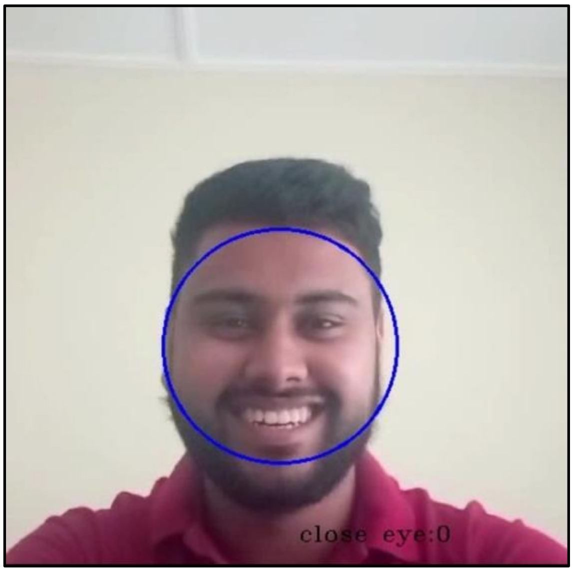


Figure 25. Open Eye State

Figure 24 and Figure 25, visually demonstrates the practical application of the attentiveness detection model in real-time scenarios. Figure 24 shows a scenario where the eyes of the individual are closed. This model accurately detects and labels this state as 'Closed' with a close eye score of 2. Figure 25 shows an instance where the eyes are open. The model successfully identifies this condition and correctly labels it as 'Open' with a close eye score of 0. These images are showing the real-world efficacy of the model, and validates its ability to consistently distinguish between open and closed eye states under various conditions. This confirms that the model meets both the theoretical accuracy standards and performs effectively in everyday applications.

The high accuracy of the attentiveness detection model plays a major role in the novel emotion-based tension detection model. This helps to ensure that the data fed into the tension detection model is reliable, and allows for more accurate assessments of the mental states of the students. Getting results with high precision and recall helps to accurately identify signs of disengagement or stress, and this allows the lecturers to make informed decisions to support the students effectively. Therefore, the ability of the model to consistently and accurately classify eye states impacts the effectiveness of the learning experience as well.

Emotion Detection Model

Here, the focus is to observe the results that were obtained after training and fine-tuning the DCNN model, as explained in the methodology chapter. This evaluation looks into how adjustments in training duration can impact the performance of the model in classifying various emotional expressions.

Initially, the model was trained for 100 epochs to establish a baseline for its accuracy. This was observed at 78%. The next objective was to check whether extending the training could further enhance the precision and ability of the model to generalize across different emotional states. To ensure this, the training was done for 150 epochs and this resulted in a slight increase in the overall accuracy to 79%.

	precision	recall	f1-score	support
0	0.75	0.74	0.74	899
1	0.98	1.00	0.99	899
2	0.66	0.67	0.67	899
3	0.90	0.86	0.88	899
4	0.70	0.60	0.65	899
5	0.88	0.91	0.90	899
6	0.67	0.75	0.71	899
accuracy			0.79	6293
macro avg	0.79	0.79	0.79	6293
weighted avg	0.79	0.79	0.79	6293

Figure 26. Classification Model - Emotion Detection

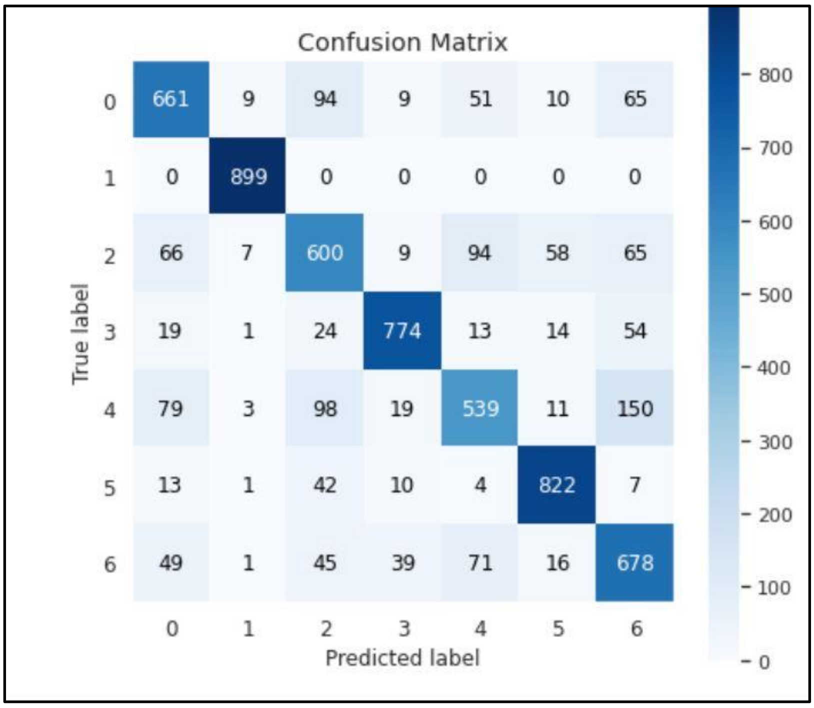


Figure 27. Confusion Matrix - Emotion Detection

Figure 26 and Figure 27 shows the outcomes of the 150-epoch training regimen. The classification report and confusion matrix that was derived, provide a granular look at the performance of the model across the emotions Anger, Disgust, Fear, Happiness, Sadness, Surprise, and Neutral. These reflect the numerical improvements in accuracy and highlight the specific areas of strength and weakness in the predictive capabilities of the model.

The classification report shown in Figure 26 shows the precision, recall, and F1-scores for each emotion category. This represents the improved consistency in emotion detection of the model. The categories are labeled from 0 to 6 where 0 = Anger, 1 = Disgust, 2 = Fear, 3 = Happiness, 4 = Sadness, 5 = Surprise, 6 = Neutral. The metrics help to identify which emotions are predicted with high reliability and which might require further refinement. It is clear that the model excels in identifying 'Disgust' with almost perfect accuracy. It also shows a strong performance in detecting 'Happiness' and 'Surprise'. However, it struggles more with emotions such as 'Fear' and 'Neutral'. For these the number of correct predictions is lower comparatively. Furthermore, the 'support' figure for each emotion is at 899 instances. This indicates the number of actual cases evaluated, and provides a balanced view of the performance of the model across similarly sized test sets for each emotion.

Figure 27 visually represents the accuracy of the predictions across all emotions. It offers insights into how frequently each emotion is correctly identified and how often it is confused with other states. This helps to pinpoint specific misclassifications, such as between closely related emotional expressions.

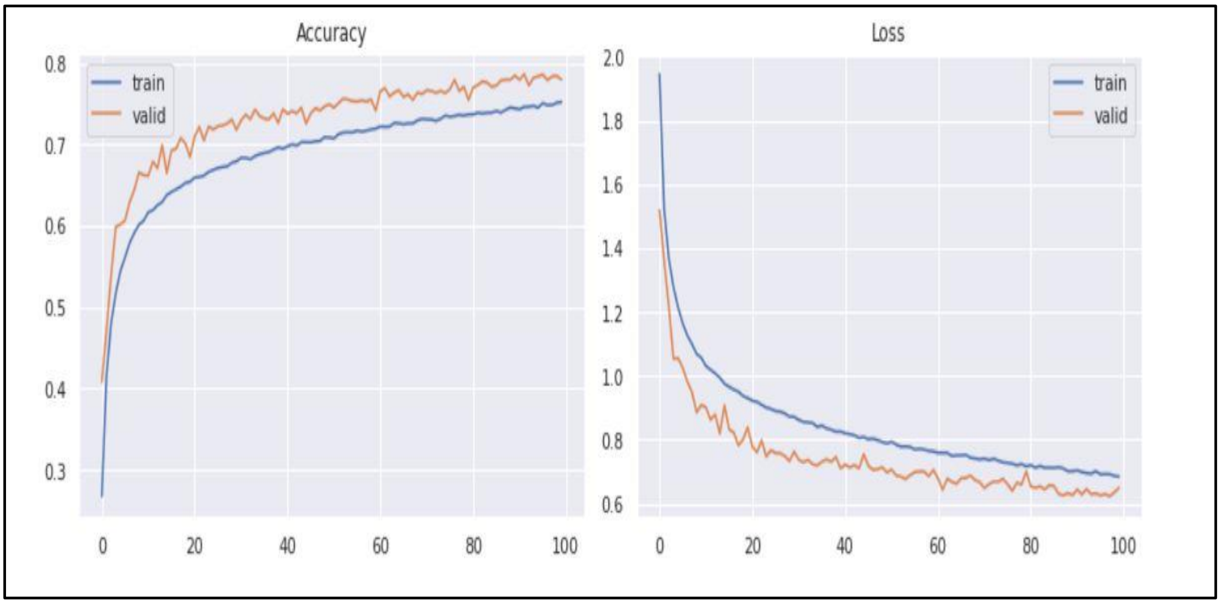


Figure 28. Classification plots - Emotion Detection

The accuracy and the loss graphs in Figure 28 show the performance of the model over an extended training period. The accuracy graph shows a steady improvement that stabilizes around the 80th epoch. Extending the training to 150 epochs was aimed at confirming the stability and the robustness of the model. This showed a slight increase in accuracy, and this further justifies the extended training period. It also helped to ensure that it maintains high performance even as the training conditions vary slightly. This suggests that the model has reached its learning capacity under the current configuration by achieving a balance between training and validation accuracy.

The loss graph shows a significant reduction in both training and validation loss and stabilizes post-60 epochs. This supports the efficiency of the model in minimizing the error over time. These trends validate the efficacy of the training regimen, and this confirms that the additional epochs contribute to refine the ability of the model to discern between complex emotional states more accurately. Thus, this enhances its reliability for practical applications in emotion recognition.

It is important for the model to be sufficiently trained because emotions play the most important role in the novel emotion-based tension detection model. The precision ensures that the inputs into the tension detection system are reliable, and this enables more accurate evaluations of the emotional states and associated tension levels of the students. Accurate emotion detection allows the identification of the effect of external factors like stress among students. Since this helps the lecturers to tailor their teaching methods, this ultimately helps to achieve the objective of this project which is improving the SQL skills of the students.

Emotion-based Tension Detection Model

This section looks into the evaluation phase of the tension detection component detailed in the methodology. Here, the focus is on factors such as accuracy and sensitivity in classifying student tension states.

The tension detection model was tested using a set of 40 video clips. These were evenly divided into two categories. 20 clips depicted the normal balanced (positive) scenarios and 20 clips depicted the high tension (negative) scenarios. Each video was analyzed frame by frame and the model assigned a tension label to each frame. The overall tension for each of the videos was determined by using the predominant label across all the frames. Due to the limited availability of clips for the other potential categories, the evaluation was conducted as a binary matrix by focusing only on normal balanced (positive) and high tension (negative) tension states. This label was then compared with the actual scenario category to evaluate the performance of the model.

Table 5. Evaluation Results of Tension Detection Component

	Actual - Normal Balanced	**Actual - High Tension**
Detected - Normal Balanced	*19 (TP)*	*2 (FP)*
Detected – High Tension	*1 (FN)*	*18 (TN)*

Table 5 represents the details obtained by evaluating the emotion-based tension detection model. It shows that 19 out of 20 actual normal balanced scenarios were correctly classified as normal balanced (TP). 18 out of 20 actual high-tension scenarios were correctly classified as high tension (TN). However, 2 normal balanced scenarios were incorrectly classified as high tension (FP) and 1 high tension scenario was incorrectly classified as normal balanced (False Negative, FN).

Using the values from Table 5, the performance matrix was calculated as follows.

Accuracy = (TP + TN) / (TP + TN + FP + FN)

= (19 + 18) / (19 + 18 + 2 + 1) = 37/40 = 0.925

Sensitivity = TP / (TP + FN)

= 19 / (19 + 1) = 19/20 = 0.95

Specificity = TN / (TN + FP)

= 18 / (18 + 2) = 18/20 = 0.90

Table 6. Performance Results of Tension Detection

Criteria	**Score**
Accuracy	92.5%
Sensitivity	95%
Specificity	90%

The high sensitivity score obtained for the emotion-based tension detection model suggests that the model is highly effective at identifying actual high-tension scenarios. The strong specificity score indicates a reliable performance in recognizing normal balanced conditions. This minimizes the unnecessary interventions or disruptions in learning environments. However, the presence of false positives and a false negative, highlights areas for further refinement of the model to improve its accuracy and reliability.

The Figure 29 shows how the tension will be shown in the emotion-based tension detection model. This offers insights into the operation of the model in real-world educational settings.

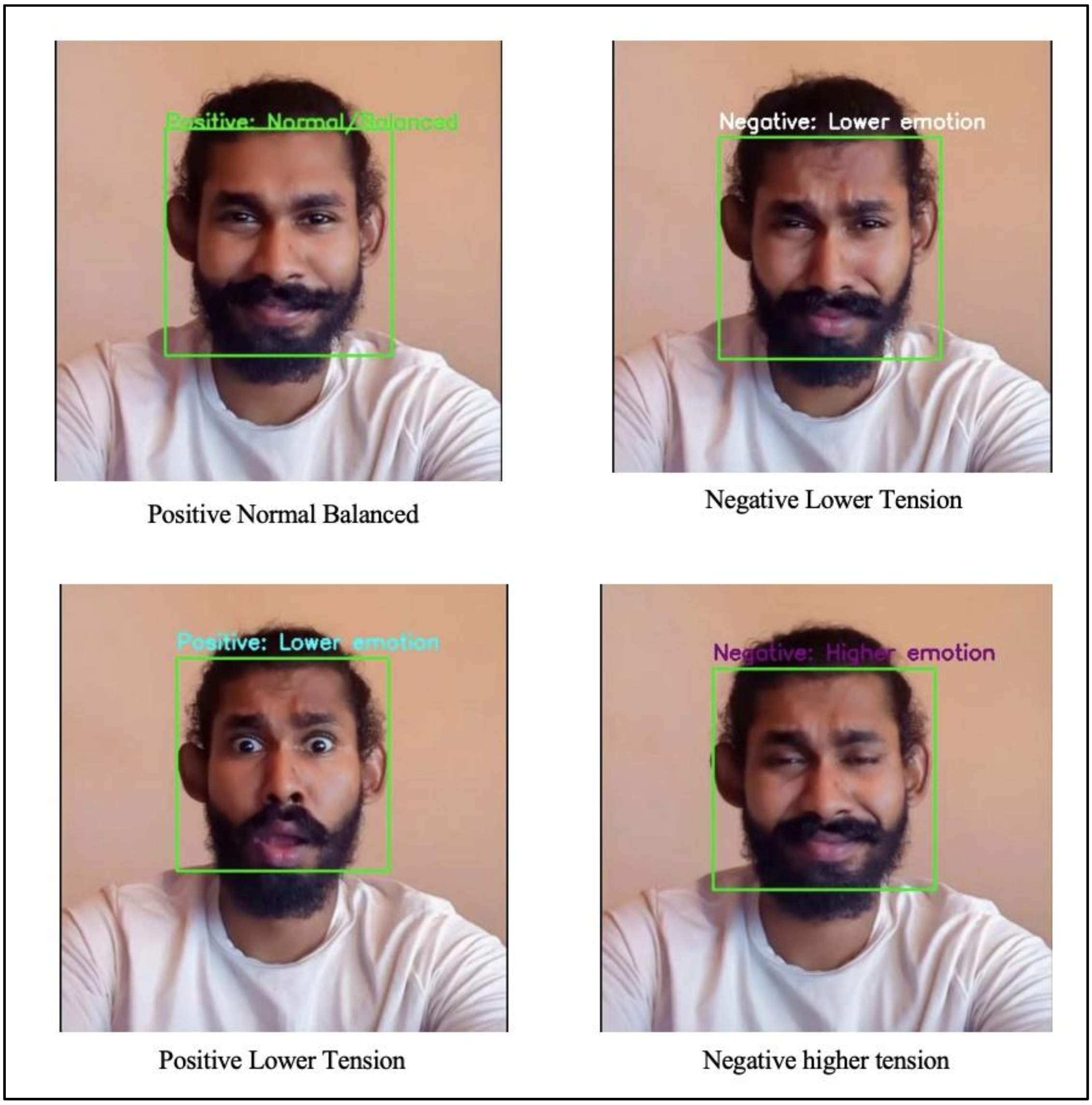


Figure 29. Tension Levels

The tension detection model is a valuable tool for lecturers to understand and respond effectively to student tensions. The results validate the potential of the model to significantly impact the educational outcomes by making it possible for more informed and responsive teaching strategies.

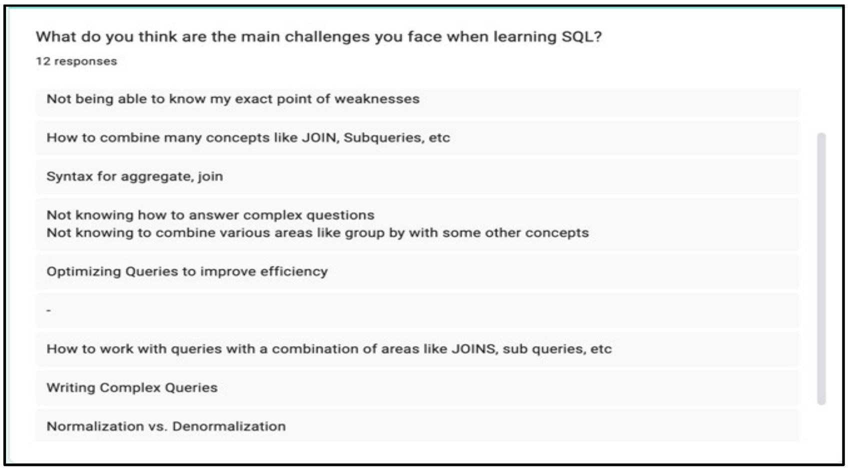
4.3.2. Usability and the Learning Impact of SQLBooster

After distributing the initial questionnaires as explained in the methodology, the results were analyzed to gain insights into various aspects of SQLBooster. The analysis of these initial responses helped to identify the main areas for development and provided a baseline understanding about the student needs and challenges in SQL learning.

Next, after the development of SQLBooster according to the insights obtained from the initial analysis, a demonstration was conducted to the students to gather their ideas. Think Aloud evaluation strategy was used to capture the real-time reactions and ideas of the students as they were introduced to various features of the tool. Then, after the demonstration, a second questionnaire was given to further identify the ideas of the students. Finally, a chi-squared test was done to quantitatively evaluate whether there is an impact of SQLBooster in improving the confidence of the students in their SQL skills.

Initial Data Collection and Analysis

Data needed for this phase was collected by distributing a Google Form questionnaire to the undergraduate students across all four years at various universities and faculties. This questionnaire was able to obtain a total of 20 responses. The results obtained from the questionnaire were analyzed to identify specific areas within SQL that the students find difficult and to identify the features that would help them to improve their SQL writing skills. The aim here was to use this understanding to create a tool that can effectively help the students to improve their SQL knowledge.


What do you think are the main challenges you face when learning SQL?
12 responses

Not being able to know my exact point of weaknesses

How to combine many concepts like JOIN, Subqueries, etc

Syntax for aggregate, join

Not knowing how to answer complex questions
Not knowing to combine various areas like group by with some other concepts

Optimizing Queries to improve efficiency

-

How to work with queries with a combination of areas like JOINS, sub queries, etc

Writing Complex Queries

Normalization vs. Denormalization

Figure 30. Challenges in SQL Learning

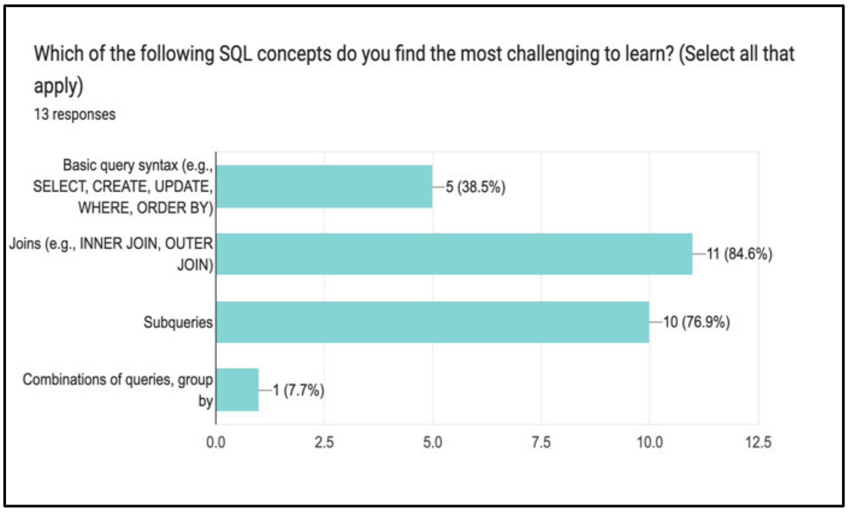


Figure 31. Challenges in SQL Concepts

Figure 30 and Figure 31 shows the areas students find difficult in SQL. It is clear that they struggle more with complex queries. Figure 31 shows that a significant 84.6% of the students identified JOINSs as a significantly challenging area and 76.9% found subqueries challenging. An important observation here was the addition of "Combinations of queries, group by" by a student under the 'other' category. Although this response was less frequent, this can be due to most students selecting the options given rather than adding the areas they find hard. So, this can be an area that most students struggle in. Even though not as frequent as others, there were students who found basic query syntax difficult as well. Furthermore, Figure 30 highlights that most students struggle with various syntaxes and query optimization. This shows a diverse range of proficiency levels among the students, Thus, it is better if the tool included different levels of queries personalized to their skill levels, so that they can improve on the areas that they mostly struggle in. This will help to address the challenges such as not being able to identify their weaknesses mentioned in Figure 30.

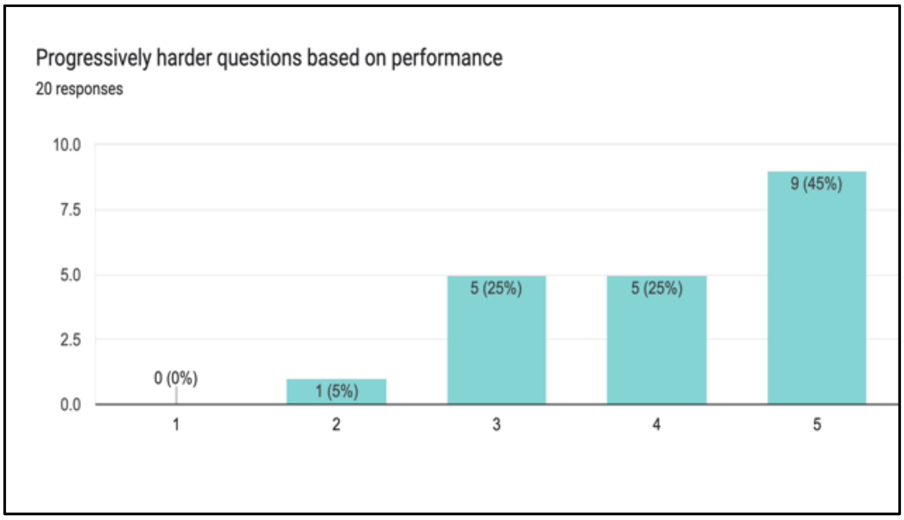


Figure 32. Progressive Questions Importance

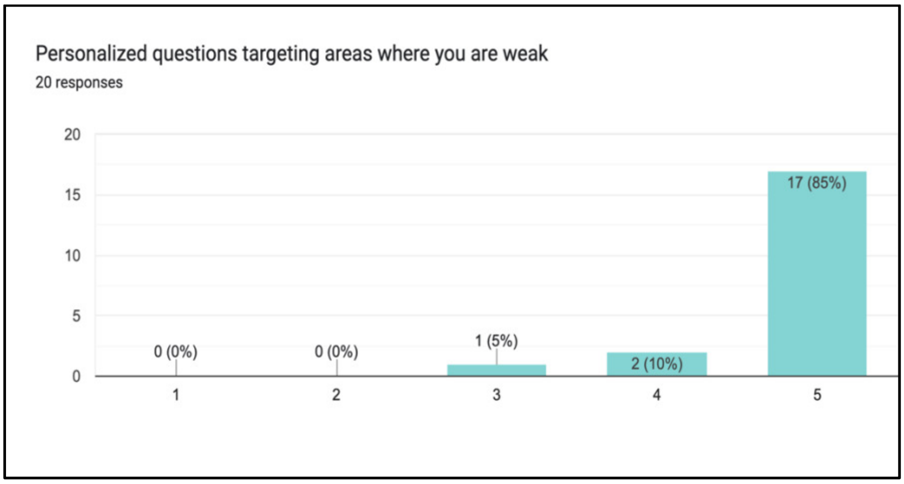


Figure 33. Personalized Questions Importance

Shifting the focus into features that can enhance the student learning in SQL, the questionnaire data indicates a strong preference for tailored educational tools. Figure 33 shows that personalized questions that target weaker areas are highly valued among the students. Furthermore, Figure 32 shows that most students consider progressively harder questions based on performance important. These show that the availability of a learning management tool that challenges students appropriately as their skills develop can be important in advancing the student understanding and proficiency in SQL.

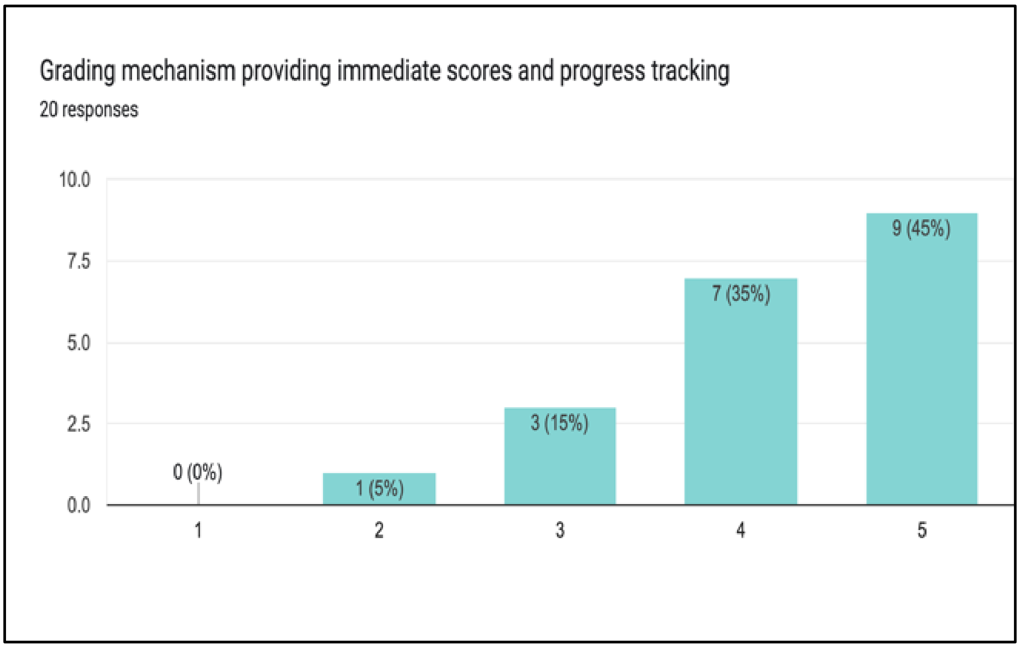


Figure 34. Grading Mechanism Importance

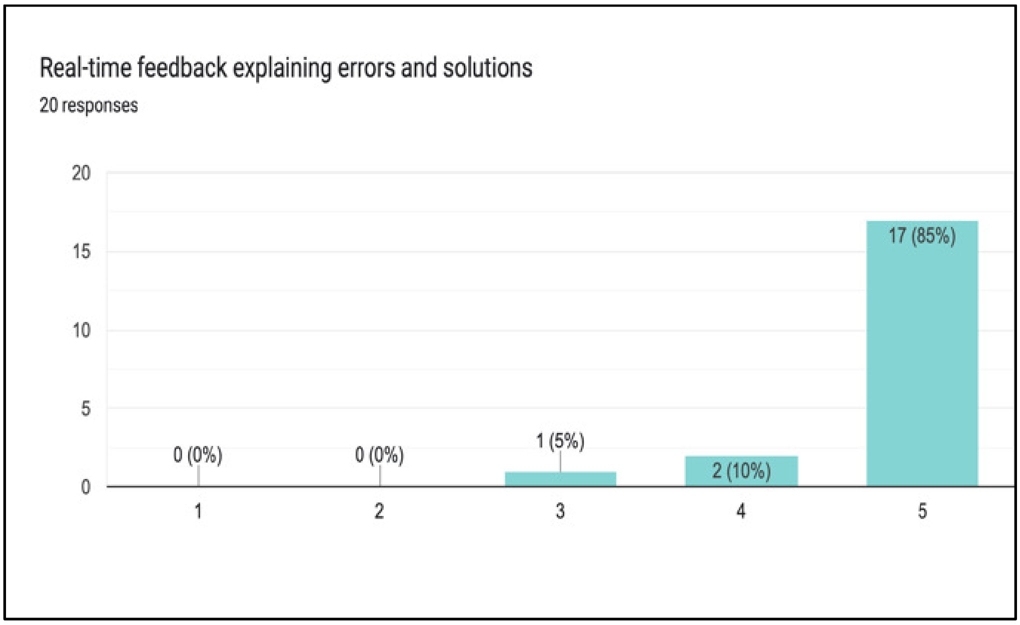


Figure 35. Real-time feedback importance

Building on this understanding, the data obtained from the questionnaire, further highlights the significant value students place on real-time feedback mechanisms. Figure 34 shows that most students appreciate the availability of automated grading that provides immediate scores and progress tracking. They allow the students to immediately understand and correct their mistakes. Thus, as shown in Figure 35, the high regard for real-time feedback can help the students quickly grasp the SQL concepts.

Furthermore, a majority of the students appreciated confidence-boosting feedback, as shown in Figure 37. This type of feedback encourages continuous learning of the students by validating their efforts. Therefore, the availability of these motivates them to engage more deeply with the material. Figure 36 shows that a significant number of students consider the availability of a tool which can provide real-time feedback and progress tracking as very important and none thinks that it is not important.

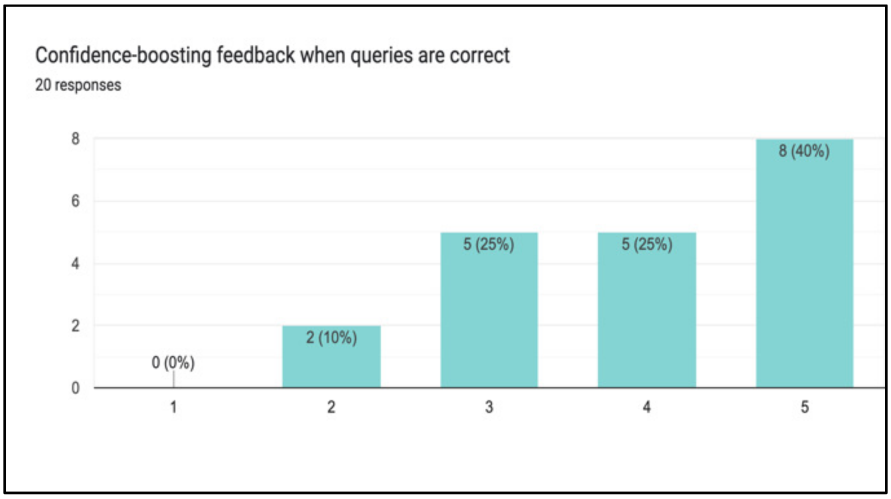


Figure 36. Importance of progress tracking and feedback providing tool

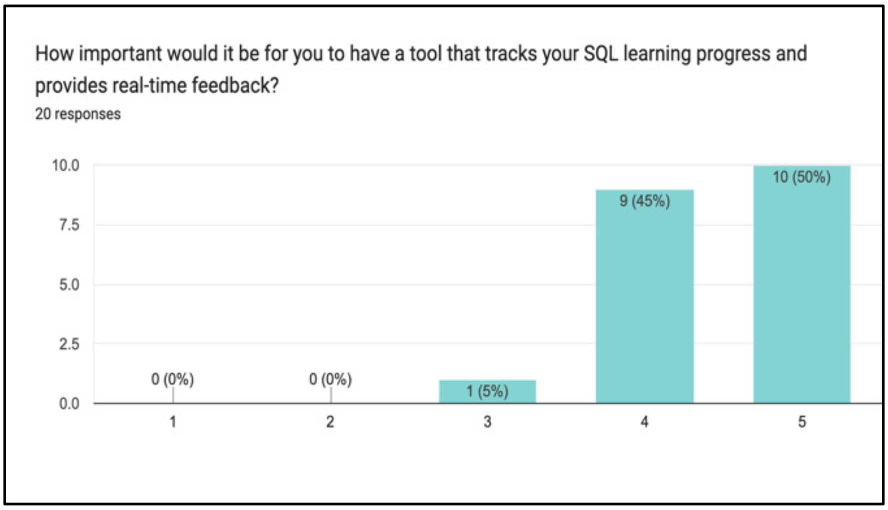


Figure 37. Importance of confidence boosting feedback

Thus, this understanding lays the groundwork for deciding which features to include in the tool and validates the need for such a tool. The next section will focus on how the tool was showcased to students and how the feedback was gathered.

Demonstration and Feedback of SQLBooster

SQLBooster was demonstrated to a group of 12 students using Google Meet. This allowed for an interactive session where I could showcase the functionalities of SQLBooster and explain its overall concept. The demonstration used the Think Aloud evaluation strategy, where students vocalized their thoughts in real-time when they interacted with the tool. This method provided immediate insights into their user experience and understanding.

During the session, I recorded the feedback of the students. I categorized these comments later using an approach similar to affinity diagrams. This helped to ensure that similar responses were grouped together and analyzed in a systematical way. This allowed for a structured evaluation of the feedback and following are the main points that was collected during this session.

The students expressed a high level of satisfaction with SQLBooster. Several students even asked whether they could incorporate it into their current learning. This enthusiasm showed the relevance and the potential impact of SQLBooster on the SQL learning process. One significant suggestion from the session was the customization of starting levels. Students recommended to allow the users to start at various levels of difficulty according to their skill level rather than beginning from the basic level.

Another main well-received area of the tool was the search functionality. That is the feature which allows the students to query any topic in the curriculum and receive clear and concise answers. Furthermore, the functionality that provided insights into common student weaknesses was also highlighted as invaluable. One student suggested expanding this to include views of individual progress and specific difficulties as well.

After this session, a Google Form questionnaire was distributed to all of the students who participated in this session. The analysis of these details revealed insightful data about the students' perceptions of the features of SQLBooster.

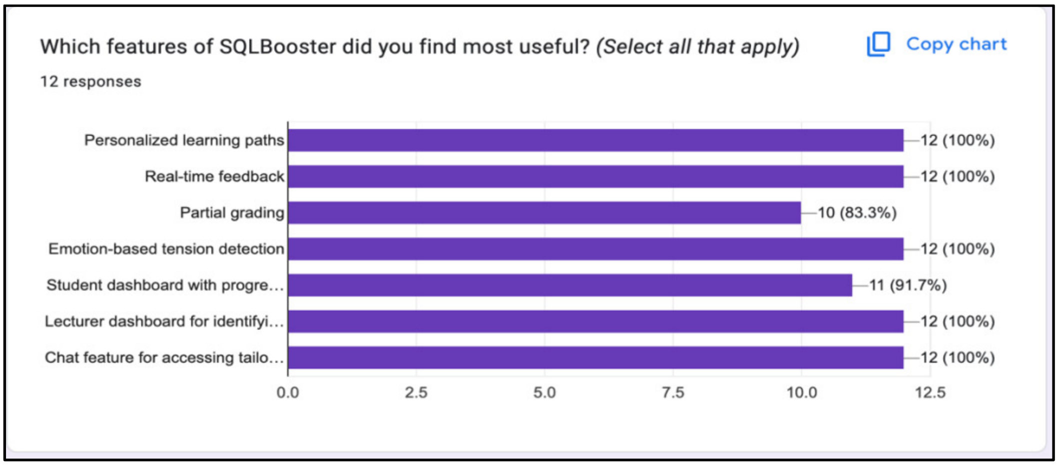


Figure 38. Most useful features of SQLBooster

Figure 38 shows that every student found the personalized learning paths and real-time feedback to be highly useful. 100% affirmed their value in enhancing their SQL learning experience. Similarly, all students appreciated the availability of the lecturer dashboard for identifying the student weaknesses, along with the chat feature for accessing tailored responses. All the students liked the concept of emotion-based tension detection. Even though not all, most of the students valued the partial grading capability and the student dashboard with progress tracking as well.

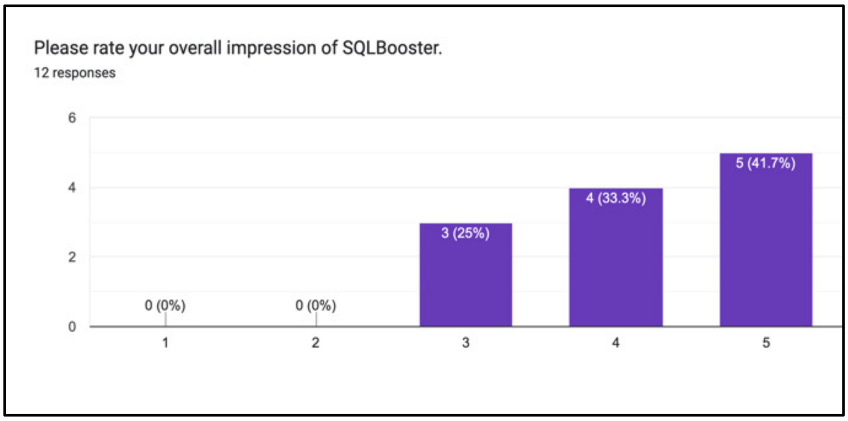


Figure 39. Overall impression of SQLBooster

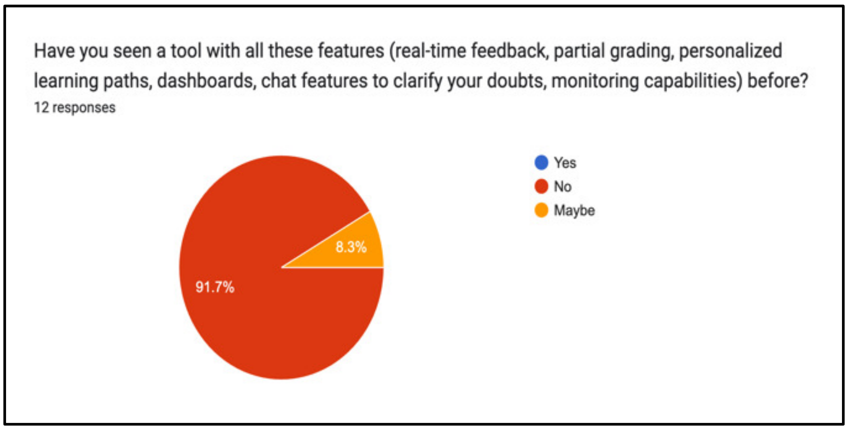


Figure 40. Previous Tools with all Features

Figure 39 and Figure 40 highlight the innovation and potential impact of SQLBooster on enhancing SQL education. A striking 91.7% of the students reported that they have never encountered a tool combining all of the features of SQLBooster. A small portion (8.3%) has responded telling their uncertainty and none of the students have marked as they have seen a tool with all these features. This suggests its novelty in the educational technology market.
The high satisfaction rates shown in Figure 39, with all students rating their overall impression at 3 or above and 75% rating it 4 or 5, reflect the effectiveness of SQLBooster in meeting the educational needs. This strong positive feedback validates the capacity of the tool to improve the SQL writing skills of the students. These results suggest that SQLBooster might be able to make a transformative impact in SQL learning, by increasing both the competence and confidence among students.

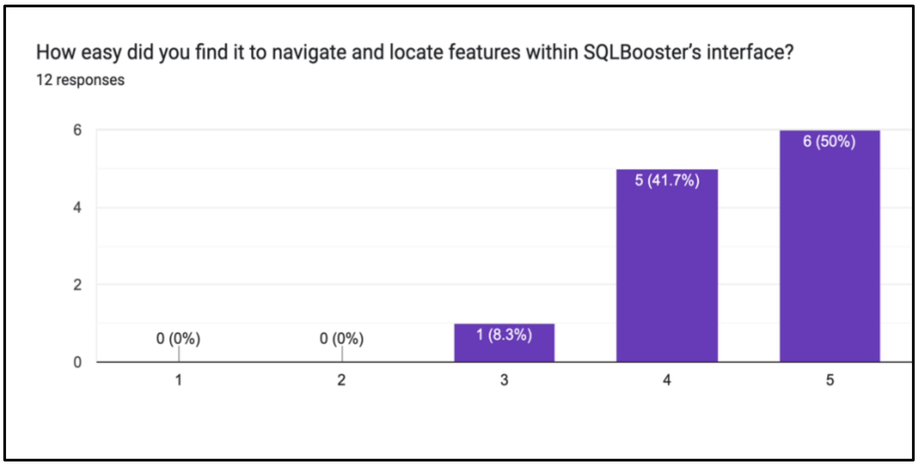


Figure 41. Navigation in SQLBooster's interface

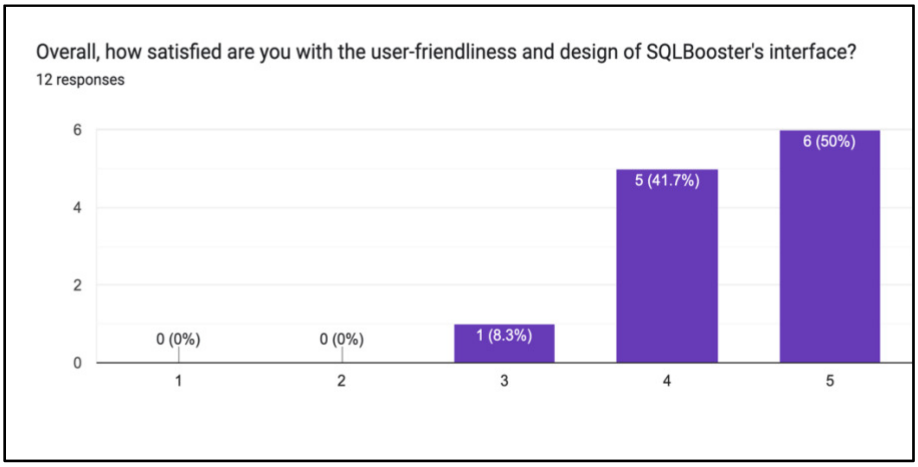
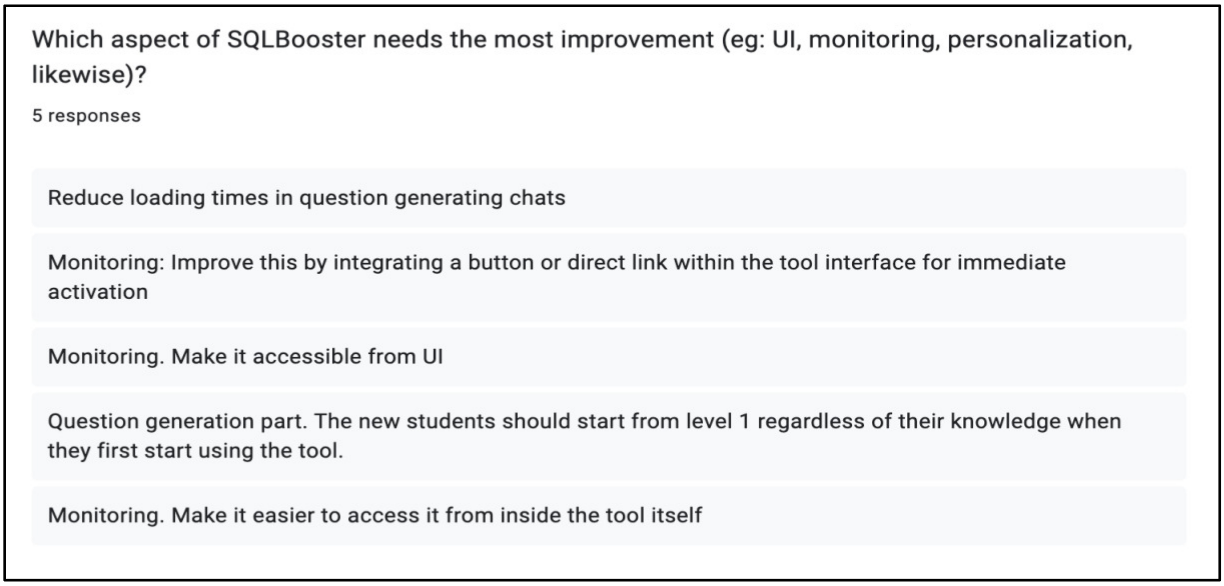


Figure 42. Overall Satisfaction with user-friendliness

When considering the usability and the user satisfaction with SQLBooster's interface, the data reflects a high degree of ease among the users. As shown in Figure 41, 91.7% of respondents rated the ease of navigation and feature location within SQLBooster as either 4 or 5. This indicates that most users find the interface intuitive and user-friendly. Similarly, Figure 42 illustrates that the overall satisfaction with the interface's design and user-friendliness is also high, with 91.7% of students expressing a high level of satisfaction (ratings of 4 and 5). These statistics highlight the SQLBooster's effective design and suggest that its user-friendly interface could significantly enhance the learning experiences.



Which aspect of SQLBooster needs the most improvement (eg: UI, monitoring, personalization, likewise)?

5 responses

Reduce loading times in question generating chats

Monitoring: Improve this by integrating a button or direct link within the tool interface for immediate activation

Monitoring. Make it accessible from UI

Question generation part. The new students should start from level 1 regardless of their knowledge when they first start using the tool.

Monitoring. Make it easier to access it from inside the tool itself

Figure 43. Improvements needed in SQLBooster

Even though the responses obtained for SQLBooster was mainly positive, there are some areas where the students have asked for improvements. As shown in Figure 43, users have identified areas for improvement mainly in the monitoring features and their accessibility from SQLBooster. Users have suggested improvements such as reducing loading times for question-generating chats, and making monitoring features more readily accessible from the user interface. These improvements can further improve the user experience and can make the tool more efficient and user-friendly.

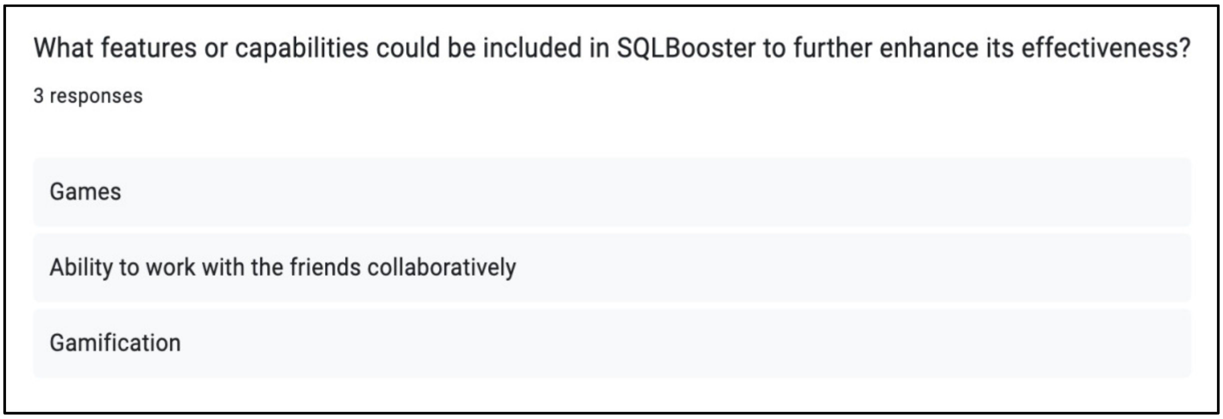


Figure 44. Other features that can be added to SQLBooster

Furthermore, as shown in Figure 44, suggestions for additional features like gamification and collaborative capabilities highlight a demand for more interactive and engaging learning methods. Adding these elements to SQLBooster can make it both a tool for learning SQL and an enjoyable platform that encourages consistent use and interaction among peers. Such features can enhance the learning process by making it more dynamic and effective.

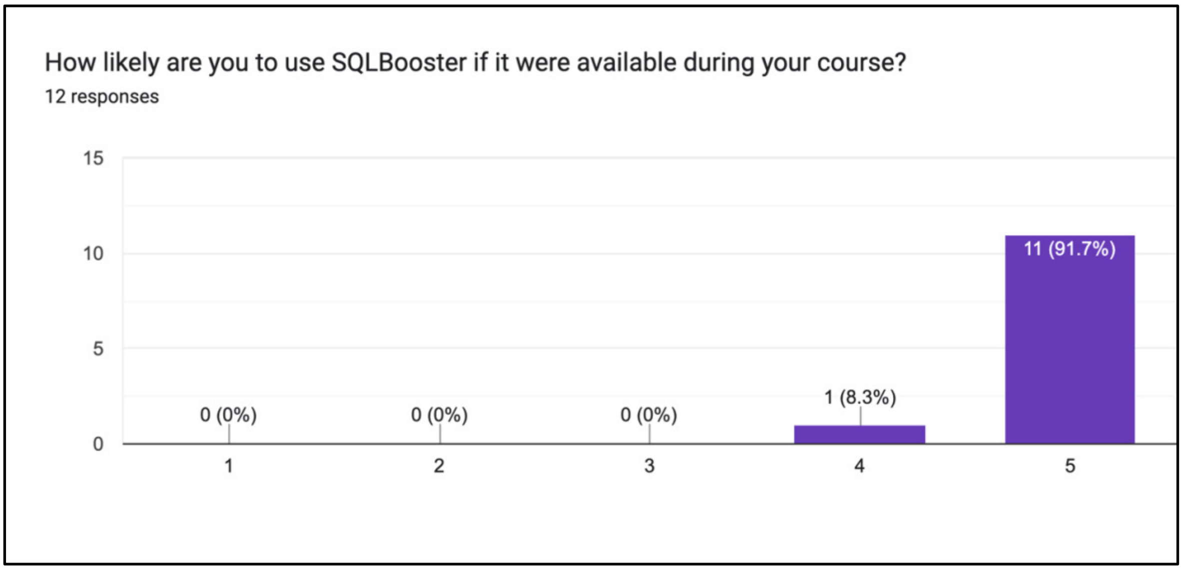


Figure 45. Likelihood of using SQLBooster

Finally, as shown in Figure 45 an overwhelmingly positive 91.7% of the respondents have indicated a high likelihood (rating of 5) of using SQLBooster. This shows the strong enthusiasm and the value in its features. This enthusiasm is further validated by the absence of any students rating their likelihood as low (1 or 2). This strong positive response shows a high level of enthusiasm and a recognized need for such a resource among students.

Quantitative Analysis of the impact of SQLBooster

The main objective of this research is to improve the SQL knowledge of the undergraduate students. In order to determine whether SQLBooster has the potential to improve the SQL skills, a chi-squared test was conducted. The null hypothesis (H_0) states that SQLBooster has no effect on the improvement of students' confidence in writing SQL queries. The alternative hypothesis (H_1) proposes that SQLBooster has a significant effect on the improvement of students' confidence in writing SQL queries. Here, the distribution of responses regarding the students' confidence in their SQL skills were compared before and after considering the potential impact of SQLBooster.

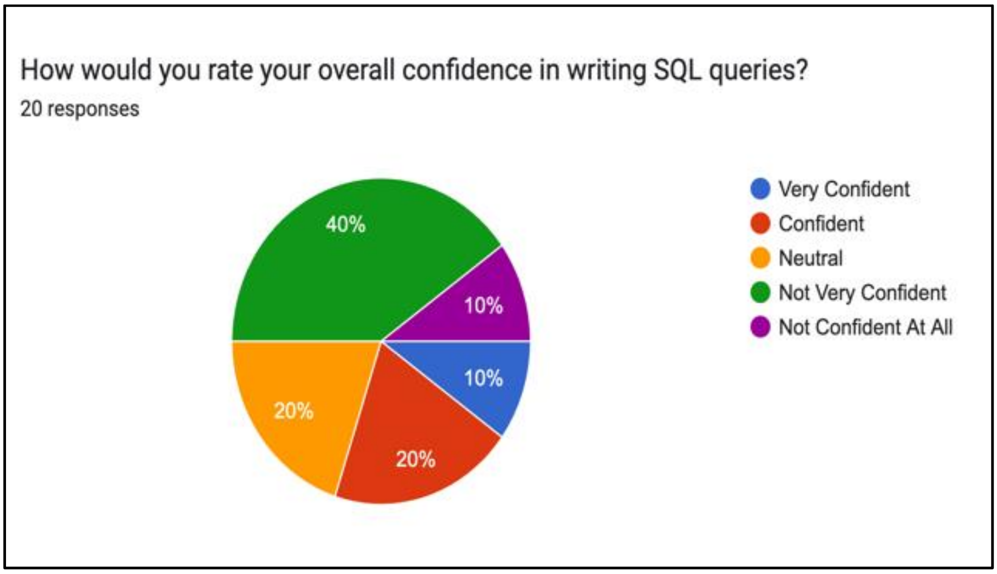


Figure 46. Overall confidence in SQL before SQLBooster

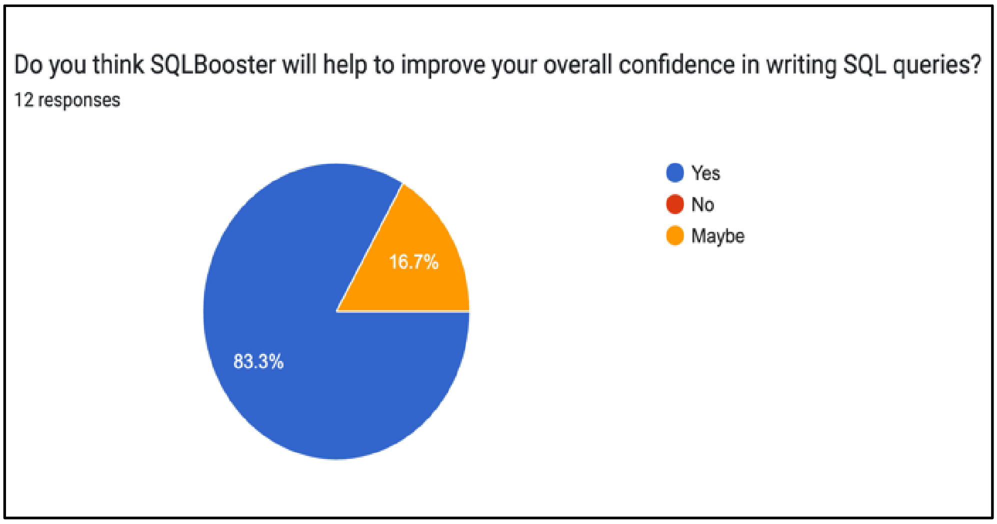


Figure 47. Overall confidence in SQL after SQLBooster

According to the data obtained from the questionnaires, the initial student confidence levels in SQL was divided into 5 categories namely ‘Very Confident’, ‘Confident’, ‘Neutral’, ‘Not Very Confident’, and ‘Not Confident At All’. After presenting SQLBooster to the students, their assessments of its potential to enhance their SQL skills were categorized into three responses as ‘Yes’, ‘No’, and ‘Maybe’. Then in order to conduct a valid comparison, these data were re-categorized into two groups (Confident, Not Confident) for both pre- and post-intervention data.

For the pre-intervention data, the responses ‘Very Confident’ and ‘Confident’ were combined under the ‘Confident’ category, while the responses ‘Neutral’, ‘Not Very Confident’, and ‘Not Confident At All’ were grouped under ‘Not Confident’. For the post-intervention data, the response ‘Yes’ was classified as ‘Confident’, and both ‘No’ and ‘Maybe’ were categorized as ‘Not Confident’.

Next, in order to prepare for the chi-squared test, the number of students in each category was calculated based on the percentage distribution from the survey results. The calculation was first done for the pre-intervention data, as shown below:

Confident = 'Very Confident' + 'Confident’

= 10% + 20% = 30% of the students= 20 * 30% = 6 students

Not Confident = ‘Neutral’ + 'Not Very Confident’ + 'Not Confident At All'

= 20% + 40% + 10% = 70% of the students = 20 * 70% = 14 students

The same process was applied for the post-intervention data:

Confident = ‘Yes’ = 83.3% of the students

= 12 * 83.3% = 10 students

Not Confident = ‘No’ + ‘Maybe’ = 0% + 16.7%

= 16.7% = 12 * 16.7% = 2 students

Table 7. Contingency Table

	Confident	**Not Confident**	**Total**
Before	$O_{1,1} = 6$	$O_{1,2} = 14$	20
After	$O_{2,1} = 10$	$O_{2,2} = 2$	12
Total	16	16	32

Table 7 represents the contingency table that was constructed from the observed distribution of student responses before and after the introduction of SQLBooster. To perform the chi-squared test, the expected frequencies for each cell in the contingency table should be calculated. The formula for calculating expected frequencies is shown in Equation 6.

$$Expected\ Frequency = \frac{row\ total * column\ total}{ground\ total}$$

Equation 6: Expected Frequency

This formula helps to determine the frequency expected in each cell of the table under the assumption that the null hypothesis is true. Table 8 represents the expected frequency distribution that was obtained by applying the above formula.

Table 8. Expected Frequency Distribution

	Confident	**Not Confident**
Before	$E_{1,1} = 10$	$E_{1,2} = 10$
After	$E_{2,1} = 6$	$E_{2,2} = 6$

The next step is to calculate the chi-squared (χ^2) test statistic. This can be calculated using the formula in Equation 7 Here O_i is the observed frequency, and E_i is the expected frequency.

$$\chi 2 = \frac{\Sigma(\mathrm{Oi} - \mathrm{Ei})^2}{\mathrm{Ei}}$$

Equation 7: Chi-square formula

The following table represents the χ^2 test statistic calculation.

Table 9. Calculation of Chi-Squared Statistic

Cell	$O_{i,j}$	$E_{i,j}$	$O_{i,j} - E_{i,j}$	$(O_{i,j} - E_{i,j})^2 / E_{i,j}$
$C_{1,1}$	6	10	-4	1.6
$C_{1,2}$	14	10	4	1.6
$C_{2,1}$	10	6	4	2.67
$C_{2,2}$	2	6	-4	2.67
χ^2				**8.54**

Next, the degrees of freedom need to be calculated using Equation 8.

$$Degrees\ of\ freedom = (No.of\ rows - 1) * (No.of\ columns - 1)$$

Equation 8: Degrees of freedom

For our table, Degree of freedom = (2-1) *(2-1) = 1

The critical value for the χ^2 test was determined using the chi-square distribution table, assuming a significance level (α) of 0.05. For a degree of freedom of 1 and α=0.05, the critical value is 3.84. The χ^2 distribution will be as shown in Figure 48.

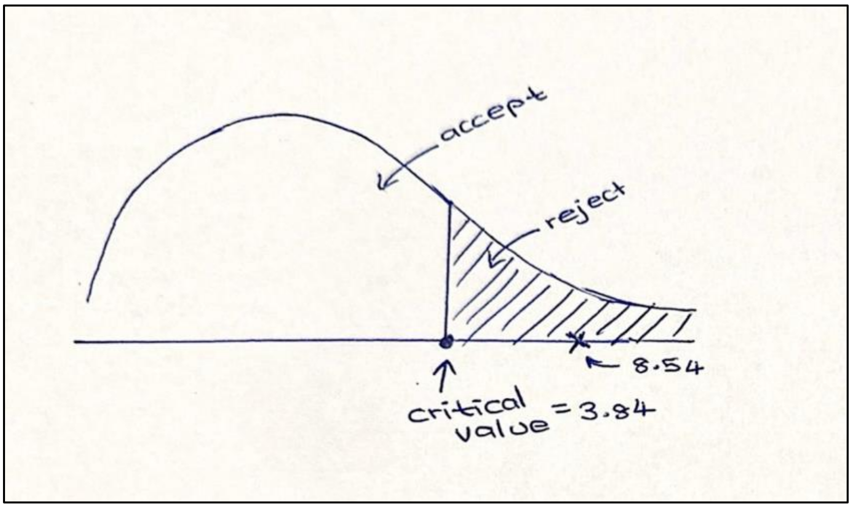


Figure 48. $\chi 2$ Distribution

Since the calculated χ^2 value of 8.54 is greater than the critical value of 3.84, the null hypothesis is rejected. This statistical outcome provides strong evidence to support the alternative hypothesis which is that SQLBooster has an effect on enhancing the SQL skills of the students. Therefore, it can be concluded that SQLBooster effectively improves the knowledge and confidence in SQL of the students.

4.4. Summary

The evaluation of the monitoring features of SQLBooster focused on the effectiveness of the emotion detection, head-pose detection, and attentiveness detection models. It also highlighted the novel emotion-based tension detection model. This used inputs from the previously mentioned models to categorize the detected student emotions into four distinct levels of tension. In order to analyze the emotion-based tension detection model, a binary matrix was used. All these monitoring capabilities showed a highly accuracy and an effectiveness.

The quantitative analysis of SQLBooster revealed its significant impact on improving the SQL skills of the students. It considered areas such as personalized learning paths, real-time feedback, and partial grading. A chi-squared test was conducted to assess the enhancement in students' confidence in their SQL abilities before and after demonstrating SQLBooster. This showed a statistically significant improvement. This analysis not only confirmed the effectiveness of SQLBooster in boosting the student confidence and competence but also highlighted the role of its innovative features in achieving these outcomes.

CHAPTER 5: CONCLUSION AND FUTURE WORK

5.1. Conclusion

This thesis presents SQLBooster which is a web-based learning management tool, that is designed to enhance SQL proficiency among the undergraduate students. It is developed using the design science methodology and uses pragmatism because the aim is not only to address the theoretical knowledge gaps but also to improve the practical outcomes that directly enhance the learning experience. SQLBooster goes beyond the other tools by integrating personalized and adaptive learning strategies with innovative monitoring technologies.

SQLBooster addresses the main research question which is “how can the tool impact the engagement and the SQL learning outcomes of the undergraduate students?” by using an array of features. These are adaptive content delivery, personalized and progressive questioning, student monitoring, real-time comprehensive feedback, and advanced partial grading capabilities that extend to both DDL and DML queries. The effectiveness of these features was proven by an evaluation, including a chi-squared test, which confirmed a statistically significant improvement in the students' confidence in SQL.

SQLBooster uses highly personalized and adaptive learning strategies to provide a dynamic educational experience to the students. It uses a multi-agent AI architecture, and uses technologies like LLM, RAG, and FastAPI to enable a dynamic learning environment that is tailored according to each students’ individual needs. Furthermore, the lecturers can upload course related documents from the lecturer dashboard and these will be intelligently integrated into the learning path of the students. It ensures that the student gets content that is aligned with the course and their individual performance levels. SQLBooster also allow the lecturers to identify the common areas of difficulties among the students which will allow them to take important decisions about the content delivery.

According to the feedback collected from the students, it could be seen that they showed a strong appreciation and favorable response towards SQLBooster. The responses highlighted that the tool has a high potential to improve SQL proficiency by addressing their specific skill gaps. Additionally, students valued the real-time feedback and partial grading features, which provided immediate insights into their performance. This feedback aligns directly with addressing some of the key research questions.

SQLBooster includes an innovative feature, the emotion-based tension detection model, which uses the inputs from the emotion detection model, head-pose detection model, and attentiveness detection model to achieve high accuracy. This model stands out because of its capability to categorize the student emotions into four levels of tension and it helps the lecturers to determine if poor performance is due to external factors like stress. Thus, this monitoring setup helps to effectively ensure the academic integrity and to assist lecturers to provide necessary assistance to the students. This marked a substantial advancement in how educational technologies can support student learning. Overall, SQLBooster has a potential to significantly enhance the SQL learning by providing tailored educational support that meets the unique needs of each student and advanced monitoring.

5.2. Future Work

SQLBooster has shown significant potential in improving the SQL learning experiences of the undergraduate students. However, several advancements can be made to further expand its capabilities.

One potential direction is to **evaluate SQLBooster using a quasi-experimental design**. Initially, a group of undergraduate students can be selected and divided into two balanced subgroups based on academic performance. This division should be conducted methodically, such as by distributing students using alternating rank order, to ensure both groups have comparable baseline capabilities. Then, a standardized test consisting of SQL questions can be given to both groups to assess their initial proficiency. The average scores should be compared to confirm that there is no significant difference in SQL knowledge at the beginning of the study. After that, SQLBooster should be introduced to only one group, while the other continues learning through traditional methods. Both groups must receive the same amount of instructional time and supporting materials, with the only difference being the use of SQLBooster in the experimental group. Finally, a follow-up SQL test with the same format as the initial one should be given to both groups to determine whether students who used SQLBooster show measurable improvement in SQL proficiency.

Another improvement involves the **execution of SQL queries** within SQLBooster. Currently, even though the students can improve their knowledge with the help of personalized content, they cannot actually execute the SQL queries and observe the results in real-time. Addition of a feature such as this can allow the students to experiment with SQL commands and directly see the impact of their queries.

Additionally, **adding a gamification layer** was a commonly requested feature in student feedback. Including features such as competitions, badges, points, and achievement leader boards can help to further motivate the students and to make the learning process more interesting (Gupta and Dr.S, 2017).

Introducing shared workspaces is another enhancement that could encourage collaboration. By making it possible for the students to work together on SQL queries, the tool can encourage a community environment and enhance peer learning. Collaborative learning not only helps the students to develop critical thinking and problem-solving skills, but also increases the engagement (Zitha et al., 2023). Therefore, shared workspaces can be an important addition to SQLBooster.

Another potential upgrade involves **allowing level customization**. Currently, the new students should start answering the personalized questions from level one because it gathers data with the performance of the students. Allowing the students to start at different levels based on their existing SQL knowledge can help to further personalize their learning experience.

To improve performance, **optimization of resource consumption and processing time** is necessary. The current monitoring models in SQLBooster need a lot of resources and processing time. This can impact the efficiency of the system. To address these challenges, future developments can focus on optimizing these models to reduce their resource consumption and to improve processing speeds.

Lastly, **enhancing the lecturer dashboard** is a valuable opportunity. Currently, the lecturer dashboard in SQLBooster offers features such as content upload and the ability to observe common areas of weaknesses among the students. However, it does not include an option to track the individual progress of the students. Improving the lecturer dashboard with this capability can allow the instructors to tailor their teaching strategies more effectively, according to the specific needs of each student

APPENDICES

I. Dataset used for attentiveness detection:
https://drive.google.com/drive/folders/1vJDxt5klmIX6tqavzKXw-hx7H-4HSgBH?usp=drive_link

II. Dataset used for head pose detection:
https://drive.google.com/drive/folders/1rG3LgTdpgdpD_3VGzMoln2CGlepUKPZ8?usp=drive_link

III. Initial Questionnaire Responses:
https://drive.google.com/file/d/1YWEZpY_cqlpLUWZK17aphNxtfh5l-IcW/view?usp=drive_link

IV. Survey Responses Following the SQLBooster Demonstration:
https://drive.google.com/file/d/1Oqx9Yb_pd2mM0_ydP9S46nfGAVfyYoE9/view?usp=drive_link

REFERENCES

Ali, S., Hafeez, Y., Abbas, M., Aqib Mehmood, M., Nawaz, A., 2021. Enabling remote learning system for virtual personalized preferences during COVID-19 pandemic. Multimedia Tools and Applications 80. https://doi.org/10.1007/s11042-021-11414-w

Alkilani, A.H., Nusir, M.I., 2022. Emotion Tracking System for Students during Online Exams, in: 2022 19th International Multi-Conference on Systems, Signals & Devices (SSD). pp. 1575–1581. https://doi.org/10.1109/SSD54932.2022.9955652

altexsoft [WWW Document], n.d. . The Good and the Bad of the Elasticsearch Search and Analytics Engine. URL https://www.altexsoft.com/blog/elasticsearch-pros-cons/

Armfield, J., 2006. Cognitive vulnerability: A model of the etiology of fear. Clinical psychology review 26, 746–68. https://doi.org/10.1016/j.cpr.2006.03.007

Asperti, A., Filippini, D., 2023. Deep Learning for Head Pose Estimation: A Survey.

Atchariyachanvanich, K., Nalintippayawong, S., Permpool, T., 2017. Development of a MySQL Sandbox for processing SQL statements: Case of DML and DDL statements, in: 2017 14th International Joint Conference on Computer Science and Software Engineering (JCSSE). pp. 1–6. https://doi.org/10.1109/JCSSE.2017.8025930

Aupperlee, A., 2021. New Research Shows Learning Is More Effective When Active.

Avdimiotis, S., Konstantinidis, I., Stalidis, G., Stamovlasis, D., 2024. Coping with Examination Stress: An Emotion Analysis. Sensors 24, 4297. https://doi.org/10.3390/s24134297

Baixauli, E., 2017. Happiness: Role of Dopamine and Serotonin on Mood and Negative Emotions. Emergency Medicine: Open Access 07. https://doi.org/10.4172/2165-7548.1000350

Barba Guamán, L.R., Valdiviezo-Diaz, P., 2022. The Attention of Students in Online Education: Using Head Pose Techniques to Detect Attention in Videoconferencing Platforms: A Case Study. International Journal of Emerging Technologies in Learning (iJET) 17, 144–155. https://doi.org/10.3991/ijet.v17i22.34241

Bhardwaj, S., Ramu, S., Guddeti, R.M.R., 2024. An Ensemble Deep Learning Approach for Emotion Monitoring System in Online Examinations, in: 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT). pp. 1–5. https://doi.org/10.1109/ICCCNT61001.2024.10724366

Blair, R., 2012. Considering anger from a neuroscience perspective. Wiley interdisciplinary reviews. Cognitive science 3, 65–74. https://doi.org/10.1002/wcs.154

Brkić, L., Mekterović, I., Fertalj, M., 2024. Automated SQL Query Evaluations in Massive Database Courses, in: 2024 47th MIPRO ICT and Electronics Convention (MIPRO). pp. 193–198. https://doi.org/10.1109/MIPRO60963.2024.10569204

Brocke, J. vom, Hevner, A., Maedche, A., 2020. Introduction to Design Science Research. pp. 1–13. https://doi.org/10.1007/978-3-030-46781-4_1

Bromberg-Martin, E.S., Matsumoto, M., Hikosaka, O., 2010. Dopamine in Motivational Control: Rewarding, Aversive, and Alerting. Neuron 68, 815–834. https://doi.org/10.1016/j.neuron.2010.11.022

Canedo, D., Trifan, A., Neves, A., 2018. Monitoring Students' Attention in a Classroom Through Computer Vision. pp. 371–378. https://doi.org/10.1007/978-3-319-94779-2_32

Chai, W., Chen, J., Wang, J., Velipasalar, S., Venkatachalapathy, A., Adu-Gyamfi, Y., Merickel, J., Sharma, A., 2023. Driver Head Pose Detection From Naturalistic Driving Data. IEEE Transactions on Intelligent Transportation Systems 24, 9368–9377. https://doi.org/10.1109/TITS.2023.3275070

Chandra, B., Banerjee, A., Hazra, U., Joseph, M., Sudarshan, S., 2019. Automated Grading of SQL Queries, in: 2019 IEEE 35th International Conference on Data Engineering (ICDE). pp. 1630–1633. https://doi.org/10.1109/ICDE.2019.00159

Chapman, H., Anderson, A., 2012. Understanding disgust. Annals of the New York Academy of Sciences 1251, 62–76. https://doi.org/10.1111/j.1749-6632.2011.06369.x

Cohen, A.O., Dellarco, D., Breiner, K., Helion, C., Heller, A., Rahdar, A., Pedersen, G., Chein, J., Dyke, J., Galvan, A., Casey, B., 2016. The Impact of Emotional States on Cognitive Control Circuitry and Function. Journal of Cognitive Neuroscience 28, 446–459. https://doi.org/10.1162/jocn_a_00906

Context [WWW Document], n.d. URL https://context.ai/compare/gpt-4o/gpt-4-turbo-preview

Cowen, P., Sherwood, A., 2013. The role of serotonin in cognitive function: Evidence from recent studies and implications for understanding depression. Journal of psychopharmacology (Oxford, England) 27. https://doi.org/10.1177/0269881113482531

Daugherty, K., Morse, R., Schmauder, A., Hoshaw, J., Taylor, J., 2022. Adjusting the Future of Adaptive Learning Technologies via a SWOT Analysis. Intersection: A Journal at the Intersection of Assessment and Learning 3. https://doi.org/10.61669/001c.36295

Dingmin, W., Qiuyuan, H., Jackson, M., 2024. Retrieve What You Need: A Mutual Learning Framework for Open-domain Question Answering.

Doyle, L., Brady, A.-M., Byrne, G., 2009. An overview of mixed methods research. Journal of Research in Nursing 14, 175–185. https://doi.org/10.1177/1744987108093962

Eapen, J., V S, A., 2023. Personalization and Customization of LLM Responses. International Journal of Research Publication and Reviews 4, 2617–2627. https://doi.org/10.55248/gengpi.4.1223.123512

Ebisawa, Y., 2008. Head pose detection with one camera based on pupil and nostril detection technique, in: 2008 IEEE Conference on Virtual Environments, Human-Computer Interfaces and Measurement Systems. pp. 172–177. https://doi.org/10.1109/VECIMS.2008.4592775

Elbawab, M., Henriques, R., 2023. Machine Learning applied to student attentiveness detection: Using emotional and non-emotional measures. Education and Information Technologies 28, 1–21. https://doi.org/10.1007/s10639-023-11814-5

Elshani, L., Pireva Nuci, K., 2021. Constructing a personalized learning path using genetic algorithms approach. https://doi.org/0.48550/arXiv.2104.11276

Facial Expression Recognition Challenge data [WWW Document], 2019. . kaggle. URL https://www.kaggle.com/datasets/nicolejyt/facialexpressionrecognition

Firoz, M., 2023. University Student's Mental Stress Detection Using Machine Learning.

Fujita, S., Takano, K., Hayami, H., 2019. Analysis of Learning Effect using a SQL Learning Support System in the Class, in: 2019 IEEE International Conference on Engineering, Technology and Education (TALE). pp. 1–4. https://doi.org/10.1109/TALE48000.2019.9225905

Gourier, N., Crowley, J., 2004. Estimating Face orientation from Robust Detection of Salient Facial Structures. FG Net Workshop on Visual Observation of Deictic Gestures.

Gupta, A., Dr.S, G., 2017. A Review on Gamification and its Potential to Motivate and Engage Employees and Customers: Employee Engagement through Gamification. International Journal of Sociotechnology and Knowledge Development 9, 42–52. https://doi.org/10.4018/IJSKD.2017010103

Gupta, B., 2024. Feedback: An Inevitable Tool for Learning, Development and Transformation 62, 5–12.

Head Pose Image Database [WWW Document], n.d. . Head Pose Image Database. URL http://crowley-coutaz.fr/Head%20Pose%20Image%20Database.html

Hosen, M., Islam, R., Naeem, Z., Folorunso, E., Chu, T., Mamun, M.A.A., Orunbon, N., 2024. Data-Driven Decision Making: Advanced Database Systems for Business Intelligence. Nanotechnology Perceptions 20, 687–704. https://doi.org/10.62441/nano-ntp.v20iS3.51

Hossen, M., Uddin, M.S., 2023. Attention monitoring of students during online classes using XGBoost classifier. Computers and Education: Artificial Intelligence 5, 100191. https://doi.org/10.1016/j.caeai.2023.100191

Hu, Y., Lu, Y., 2024. RAG and RAU: A Survey on Retrieval-Augmented Language Model in Natural Language Processing. ArXiv abs/2404.19543.

IMAGENET [WWW Document], 2021. URL https://www.image-net.org/

Indi, C., Kcs, V., Acharya, V., Prakasha, Dr.K., 2021. Detection of Malpractice in E-exams by Head Pose and Gaze Estimation. International Journal of Emerging Technologies in Learning (iJET) 6, 47. https://doi.org/10.3991/ijet.v16i08.15995

Joseph, M., Radhakrishnan, B., Acharya, S., Sudarshan, S., 2016. Partial marking for automated grading of SQL queries. Proceedings of the VLDB Endowment 9, 1541–1544. https://doi.org/10.14778/3007263.3007304

Kaushik, V., Christine, W., 2019. Pragmatism as a Research Paradigm and Its Implications for Social Work Research.

Khan, B.H., Mohamed, A., 2015. International Handbook of E-learning, Volume 1: Theoretical Perspectives and Research. Routledge.

Kulik, J.A., Kulik, C.-L.C., 1987. COMPUTER-BASED INSTRUCTION:WHAT 200 EVALUATIONS SAY.

Kusal, S., Patil, S., Choudrie, J., Kotecha, K., Vora, D., 2022. A Review on Text-Based Emotion Detection -- Techniques, Applications, Datasets, and Future Directions. https://doi.org/10.48550/arXiv.2205.03235

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., Kiela, D., 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks, in: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (Eds.), Advances in Neural Information Processing Systems. Curran Associates, Inc., pp. 9459–9474.

Liyanage, S., 2024. Next.js vs Angular: Framework Comparison on top of SSR and Performance Enhancements [WWW Document]. Medium. URL https://medium.com/@saniraliyanage/comprehensive-guide-to-next-js-0f41bf9df45b

Magalingam, A., 2022. Technology of Teaching. https://doi. org/10.34293/sijash. v10iS1-5225

Mantosh, M., 2023. JETBRAINS Blog [WWW Document]. Django vs FastAPI: Which is the Best Python Web Framework? URL https://blog.jetbrains.com/pycharm/2023/12/django-vs-fastapi-which-is-the-best-python-web-framework/

Matthews, K., Janicki, T., He, L., Patterson, L., 2012. Implementation of an Automated Grading System with an Adaptive Learning Component to Affect Student Feedback and Response Time. Journal of Information Systems 23, 71–84.

MongoDB and Oracle Compared, n.d.

MongoDB [WWW Document], n.d. . MongoDB vs. MySQL Differences. URL https://www.mongodb.com/resources/compare/mongodb-mysql

Murphy-Chutorian, E., Trivedi, M.M., 2009. Head Pose Estimation in Computer Vision: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 31, 607–626. https://doi.org/10.1109/TPAMI.2008.106

Nakamoto, R., Flanagan, B., Dai, Y., Yamauchi, T., Takami, K., Ogata, H., 2023. Enhancing Self-Explanation Learning through a Real-Time Feedback System: An Empirical Evaluation Study. Sustainability 15, 15577. https://doi.org/10.3390/su152115577

Nalintippayawong, S., Atchariyachanvanich, K., Julavanich, T., 2017. DBLearn: Adaptive e-learning for practical database course — An integrated architecture approach, in: 2017 18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD). pp. 109–114. https://doi.org/10.1109/SNPD.2017.8022708

Negron, T.P., Graves, C.A., 2017. Classroom Attentiveness Classification Tool (ClassACT): The system introduction, in: 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops). pp. 26–29. https://doi.org/10.1109/PERCOMW.2017.7917513

Nishchal, J., Reddy, S., Navya, P.N., 2020. Automated Cheating Detection in Exams using Posture and Emotion Analysis, in: 2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT). pp. 1–6. https://doi.org/10.1109/CONECCT50063.2020.9198691

Noordewier, M., Topolinski, S., Dijk, E., 2016. The Temporal Dynamics of Surprise. Social and Personality Psychology Compass 10, 136–149. https://doi.org/10.1111/spc3.12242

OpenAI Pricing [WWW Document], n.d. URL https://openai.com/api/pricing/

Ouared, A., Amrani, M., Schobbens, P.-Y., 2023. Learning Analytics Solution for Monitoring and Analyzing the Students' Behavior in SQL Lab Work.

Ozdamli, F., Aljarrah, A., Karagozlu, D., Ababneh, M., 2022. Facial Recognition System to Detect Student Emotions and Cheating in Distance Learning. Sustainability 14, 13230. https://doi.org/10.3390/su142013230

Panni, F.A.K., Hoque, A.S.Md.L., 2020. A Model for Automatic Partial Evaluation of SQL Queries, in: 2020 2nd International Conference on Advanced Information and Communication Technology (ICAICT). pp. 240–245. https://doi.org/10.1109/ICAICT51780.2020.9333475

Park, Y., Jo, I.-H., n.d. Development of the Learning Analytics Dashboard to Support Students' Learning Performance.

Permpool, T., Nalintippayawong, S., Atchariyachanvanich, K., 2019. Interactive SQL Learning Tool with Automated Grading using MySQL Sandbox.

Petroni, F., Siciliano, F., Silvestri, F., Trappolini, G., 2024. IR-RAG@SIGIR24: Information Retrieval's Role in RAG Systems.

Potluri, T., S, V., K, V., 2023. An automated online proctoring system using attentive-net to assess student mischievous behavior. Multimedia Tools and Applications 82, 1–30. https://doi.org/10.1007/s11042-023-14604-w

Prabhakar, P., Palaniswamy, S., Pati, P.B., 2023. Stress Analysis in Online Examination using Deep Learning Approach, in: 2023 IEEE 8th International Conference for Convergence in Technology (I2CT). pp. 1–7. https://doi.org/10.1109/I2CT57861.2023.10126298

Raiaan, M.A.K., Mukta, Md.S.H., Fatema, K., Fahad, N.M., Sakib, S., Mim, M.M.J., Ahmad, J., Ali, M.E., Azam, S., 2024. A Review on Large Language Models: Architectures, Applications, Taxonomies, Open Issues and Challenges. IEEE Access 12, 26839–26874. https://doi.org/10.1109/ACCESS.2024.3365742

Rajandran, R., 2024. Marketcalls [WWW Document]. URL https://www.marketcalls.in/python/flask-vs-fastapi-sync-and-async-comparison-in-fintech-applications-python-tutorial.html#:~:text=Synchronous%20by%20nature%3A%20Flask%20processes,I%2FO%2Dbound%20operations.&text=FastAPI%20is%20a%20modern%2C%20asynchronous,ASGI%20and%20designed%20for%20speed.

Ramaswami, G., Susnjak, T., Mathrani, A., 2023. Effectiveness of a Learning Analytics Dashboard for Increasing Student Engagement Levels. Journal of Learning Analytics 10, 1–20. https://doi.org/10.18608/jla.2023.7935
Redis [WWW Document], n.d. . Introducing the Redis Vector Library for Enhancing GenAI Development. URL https://redis.io/blog/introducing-the-redis-vector-library-for-enhancing-genai-development/
Reilly, C.F., 2018. Experience with Active Learning and Formative Feedback for a SQL Unit, in: 2018 IEEE Frontiers in Education Conference (FIE). pp. 1–9. https://doi.org/10.1109/FIE.2018.8659173
Robinson, J., Demaree, H., 2009. Experiencing and regulating sadness: Physiological and cognitive effects. Brain and Cognition 70, 13–20. https://doi.org/10.1016/j.bandc.2008.06.007
Rodríguez-Arce, J., Lara Flores, L., Portillo-Rodríguez, O., Martinez Mendez, R., 2020. Towards an anxiety and stress recognition system for academic environments based on physiological features. Computer Methods and Programs in Biomedicine 190, 105408. https://doi.org/10.1016/j.cmpb.2020.105408
Ruliah, D., Syahrial, Z., Muchtar, H., 2019. The Computer Assisted Instruction Model Based on a Combination of Tutorial Model and Drill and Practice Model in the Instructional Design of Database Systems in Information Technology Colleges. Universal Journal of Educational Research 7, 117–124. https://doi.org/10.13189/ujer.2019.071614
Sadiq, S., Orlowska, M., Sadiq, W., Lin, J., 2004. SQLator: an online SQL learning workbench, in: ACM SIGCSE Bulletin. pp. 223–227. https://doi.org/10.1145/1026487.1008055
Saeed, A., 2015. Role of Database Management Systems (DBMS) in Supporting Information Technology in Sector of Education 6, 6–391. https://doi.org/10.21275/ART20173499
Sahoo, A., 2024. React.js vs Next.js : The Ultimate Comparison for Modern Developers [WWW Document]. infiways. URL https://www.infyways.com/react-js-vs-next-js/
Samir, M.A., Maged, Y., Atia, A., 2021. Exam Cheating Detection System with Multiple-Human Pose Estimation, in: 2021 IEEE International Conference on Computing (ICOCO). pp. 236–240. https://doi.org/10.1109/ICOCO53166.2021.9673534
Schmidt, P., Reiss, A., Dürichen, R., Van Laerhoven, K., 2019. Wearable-Based Affect Recognition—A Review. Sensors 19, 4079. https://doi.org/10.3390/s19194079
Shi, J., Li, H., Gu, H., Zhou, L., 2017. Research and development of intelligent online examination monitoring system, in: 2017 12th International Conference on Computer Science and Education (ICCSE). pp. 57–62. https://doi.org/10.1109/ICCSE.2017.8085463
Soler, J., Prados, F., Boada, I., Poch, J., 2006. A web-based tool for teaching and learning SQL.
Strahl, A., 2024. 10 SQL Skills You Need to Know in 2024 [WWW Document]. DATAQUEST. URL https://www.dataquest.io/blog/10-sql-skills-you-need-to-know/ (accessed 10.27.24).
Tankala, Y., K. Paul, J., Manikandan, V.M., 2021. A MULTIPURPOSE ARTIFICIAL INTELLIGENCE BASED ATTENTIVENESS MONITORING SYSTEM. IAEME. https://doi.org/10.34218/IJARET.12.2.2021.068
Taylor, D., Yeung, M., Bashet, A.Z., 2021. Personalized and Adaptive Learning. pp. 17–34. https://doi.org/10.1007/978-3-030-58948-6_2
Tengli, M., 2020. RESEARCH ONION: A SYSTEMATIC APPROACH TO DESIGNING RESEARCH METHODOLOGY.
The Importance of Writing Efficient SQL Code, 2023.
Titovskaia, Titovskii, Titovskaya, Pyzhikov, 2019. Interactive Approach And Technique of Teaching Future Experts in Database Development.

Todorova, G., 2022. Optimizing Feedback in Learning Management Systems to Increase Student Engagement.
Venkatamuniyappa, V.K., 2018. Towards automatic grading of SQL queries.
Wang, H., Huang, T., Zhao, Y., Hu, S., 2023. The Impact of Dashboard Feedback Type on Learning Effectiveness, Focusing on Learner Differences 15.
Wang, H., Xia, Y., 2021. Research and design of an attention monitoring system based on head posture estimation, in: 2021 6th International Conference on Intelligent Computing and Signal Processing (ICSP). pp. 1456–1459. https://doi.org/10.1109/ICSP51882.2021.9408711
What is SQL [WWW Document], 2024. . Amazon Web Services (AWS). URL https://aws.amazon.com/what-is/sql/#:~:text=SQL%20was%20invented%20in%20the,SQL%20relational%20database%20management%20system. (accessed 10.26.24).
Wooldridge, M., R. Jennings, N., 1995. Intelligent agents: Theory and practice. Knowledge Engineering Review.
Wu, X., Fan, J., Guo, J., Zhou, J., Ren, J., Liu, C., Luo, J., 2017. Regulating Anger under Stress via Cognitive Reappraisal and Sadness. Frontiers in Psychology 8, 1372. https://doi.org/10.3389/fpsyg.2017.01372
Zitha, I., Mokganya, G., Sinthumule, O., 2023. Innovative Strategies for Fostering Student Engagement and Collaborative Learning among Extended Curriculum Programme Students. Education Sciences 13, 1196. https://doi.org/10.3390/educsci13121196