Web-based Management Information System for Gampaha Technical College

K.H.S. Thilakarathne

2023

Web-based Management Information System for Gampaha Technical College

A dissertation submitted for the Degree of Master of Information Technology

K.H.S. Thilakarathne

2023

DECLARATION

Name of the student: K.H. Sanjeevani Thilakarathne
Registration number: 2019/MIT/092
Name of the Degree Programme: Masters in Information Technology
Project/Thesis title: Web-based Management Information System for Gampaha Technical College

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)**
	30/09/2024

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	Dr. Kasun Karunanayaka		
Signature			
Date	30/09/2024		

ABSTRACT

Technical education and training have emerged as a significant field in Sri Lankan education. Currently, there are 39 technical colleges scattered throughout Sri Lanka. Under the direction of the Ministry of Skills Development and Vocational Training, these institutions operate under the Department of Technical Education and Training (DTET). The vision of the technical college is to become an internationally renowned, leading technical education and training provider.

Gampaha Technical College represents the Western Province. There are 35 courses available at Gampaha Technical College. It has 75 permanent staff, 35 instructors, 3 lecturers, and 30 external lecturers. These technical institutes accept students ranging in age from 17 to 28. Students are free to choose their preferred courses. The rules and regulations are nearly comparable to those that govern the school system.

Each course enrolled 25 students, and then manual students' attendance is considered a critical aspect for both students and instructors/lecturers. A technical college instructor is required to write and submit lesson plans, reports, and other relevant paperwork in addition to lecturing. Sri Lanka uses a teacher-centered educational system. Teaching is the teacher's top priority. The time spent on these extra activities may diminish the effectiveness of the learning and teaching process.

A web-based management information system is proposed to address these issues. The purpose of this system is to enable teachers to complete their duties efficiently, accurately, and reliably. Using this approach, lecturers can easily develop year-long activity plans. The instructor or examiner may be able to record each student's attendance using their laptop or mobile phone with face recognition technology. The system enables instructor evaluation, student evaluation, and report generation. The director or principal may be able to track all activities that have been processed through the system. Also, academic staff could be able to manage all information efficiently and maintain consistent data without wasting paper or unnecessary storage.

The system utilizes various modern technologies, and users will experience a user-friendly environment throughout their procedure. The iterative and incremental development methodology is utilized in the development process because the requirements are confirmed and collected from the outset. To develop the web applications, the project used HTML, CSS,

Bootstrap, JavaScript, JSON, face recognition libraries, OpenCV, Stripe payment methods, Django, and MySQL Server to manage the database of the system.

ACKNOWLEDGEMENT

I would like to convey my sincere gratitude and appreciation to all those who assisted me accomplish my project successfully by offering advice and support. First, I would like to express my deepest gratitude to my project supervisor, Dr. Kasun Karunanayaka, Senior Lecturer at UCSC, for the useful comments, remarks, and consistent encouragement given throughout the entire process. Also, I would like to thank the course coordinator of the MIT program and the academic staff of UCSC for giving me this valuable opportunity to follow this recognized master's degree program and for providing me with the guidelines for successful completion.

I would like to thank Gampaha Technical College's Mr. B.D.P. Nandasiri(HOD commerce section) and all academic staff for their assistance with data collection and inquiries throughout the entire project. I would also be grateful to the experts and participants who contributed to delivering the necessary information for the project's implementation and evaluation. The feedback surveys could not have been successfully conducted without their motivation and participation.

Finally, I want to thank my family for their unwavering encouragement, patience, and support during the whole process.

TABLE OF CONTENTS

TABLE OF FIGURE

LIST OF TABLES

LIST OF ACRONYMS

AJAX – Asynchronous JavaScript and XML

CSS – Cascading Style Sheet

DBMS – Database Management System

GUI – Graphical User Interface

HOD – Head of the Department

HTML – Hypertext Markup Language

MIS – Management Information System

MVT – Model View Templates

PDF – Portable Document Format

SDLC – Software Development Life Cycle

SQL – Structured Query Language

UAT – User Acceptance Testing

UI – User Interface

UML – Unified Modeling Language

URL – Uniform Resource Locator

CHAPTER 1 - INTRODUCTION

1.1 Project Overview

Evidently, in today's world, file-based systems cannot be used to do tasks in an orderly, precise, or effective manner (Imene and Imhanzenobe, 2020). Therefore, having everything digitally recorded globally is a huge advantage, particularly for management information systems.

At the Gampaha Technical College, the administrative staff gathers all the data once each course has begun in order to manage the information about students, staff, trades, courses, and subjects. Keep these specifics in several files after that. Also, students' attendance is considered a critical aspect for both students and instructors or lecturers in the technical college. Students who attend technical colleges on a regular basis enhance the quality of tertiary educational technology.

The lecturer or instructor should develop the essential fundamental plans prior to the start of the teaching-learning process in the classroom in accordance with the pertinent curriculum material. The developed plans should be implemented by the due date and time, and the instructor should prepare the data and papers needed to measure their efficacy and efficiency. Additionally, student course payment receipts are kept on file by the administration staff.

This project facilitates the technical college's management information system by utilizing both a web application and a mobile web view approach. Through a mobile web app, it uses the most trustworthy method of uniquely identifying students. The technical college uses the web application to manage the management information system extremely effectively. The project aims to develop three integrated systems for Gampaha Technical College to enhance the overall learning experience and streamline administrative tasks. The systems include:

Student Face Recognition Information and Attendance Management System: This system enhances attendance management by utilizing face recognition technology. It involves developing a system that can capture and recognize students' faces and automatically mark attendance based on the face recognition data. The system will also maintain a record of attendance, generate reports, and provide real-time attendance tracking for instructors and administrators.

Instructor's Activity Plans: This system focuses on providing instructors with tools and features to plan and manage their course activities effectively. It includes functionalities such as

identifying course objectives, planning schedules, developing course content, and implementing the content. The system aims to streamline the process of creating and organizing course materials, assignments, and assessments.

Online Course Fee Payment: This system aims to simplify the fee payment process for students by providing a convenient and secure online payment platform. It involves identifying suitable payment options and gateways, developing a payment system that integrates with the Technical College's financial management system, and ensuring a smooth, hassle-free, and efficient fee payment process for students, enhancing their overall experience and reducing administrative efforts for the Technical College.

The project will follow the Iterative and Incremental Development Model, allowing for continuous improvement and refinement of each system through multiple iterations. Each iteration will focus on specific tasks and deliver a functional increment to the system.

The development process will include tasks such as identifying course objectives, planning schedules, developing and reviewing course content, and implementing the content for the Instructor's Activity Plans. For the Online Course Fee Payment, tasks will involve identifying payment options, developing the payment system, and integrating it with the Technical College's financial management system. The Student Face Recognition Information and Attendance Management System will require tasks like developing face recognition technology, integrating it with the attendance management system, and ensuring data security and privacy.

Additionally, the project includes testing and quality assurance phases to ensure the reliability, accuracy, and usability of the systems. Deployment and maintenance stages will be conducted to deploy the systems in the Technical College's infrastructure and provide ongoing support and maintenance.

By adopting the Iterative and Incremental Development Model, the project aims to deliver robust, efficient, and user-friendly systems that improve the Technical college's administrative processes, enhance the learning experience, and increase overall operational efficiency.

1.2 Motivation

Every higher education institution in Sri Lanka is required to keep accurate attendance records of their students in order to decrease student dropout rates and ensure the effective operation of the institution (Bandara et al., 2022). It typically takes 10 to 15 minutes to take attendance by calling out the names of each student or by circulating an attendance sheet. It causes several other problems, such as generating a fake presence and losing attendance. Additionally, students will be distracted from their focus on the questions in the examination hall when attendance is taken by verifying signatures.

Furthermore, prior to the start of the teaching-learning process in the classroom, the lecturer or instructor should develop the essential fundamental plans (year plan, assessment plan, semester notes, lesson plan) in accordance with the pertinent curriculum material. Making lesson plans for each subject taught, as well as compiling pertinent records (daily teaching logs, monthly delivery reports), papers (skill standards, curriculum, training guides, and teaching guides), and tasks, takes a significant amount of time and effort. An integral part of the teaching strategies is planning, writing, and reporting. However, it will have an effect on an instructor's effectiveness if they are too busy to remember or disregard them during the process.

If an automated system can successfully maintain information, attendance, and instructor activity plans, then this valuable time can be used for other important tasks like practicing difficult examples, advancing students' careers, clarifying doubts, imparting knowledge beyond the syllabus, and allowing students to concentrate solely on the exam paper during the examination. Therefore, the "Management Information System" will play a crucial role in achieving such output, minimizing time, reducing human errors, and increasing the accuracy of attendance records in the manual system. The Management Information System stems from the desire to enhance operational efficiency, improve user experiences, and leverage data for informed decision-making. By embracing these technological advancements, the technical college can position itself as a modern and efficient educational organization, ultimately benefiting instructors, students, and the overall technical college community.

1.3 Objectives

- Digitalize the current system of managing student and staff information to secures data, increase efficiency and reduce paper waste.
- Provide a reliable and secure platform for storing and managing student and staff information.
- To generate timetables for each class, generate daily attendance report automatically which marked using Face ID, and SMS or email notifications of the instructor's regular activity .
- Provide online payment facility for students.
- To provide different types of MIS reports according to user requests. Automatic generation of end-of-semester attendance reports with graphical views and Enhance reporting and data analytics capabilities to support decision-making.

1.4 Scope of Project

The suggested system is mainly concerned with automating and maintaining instructor activity plans, student attendance records. It also manages all the information about students, staff, exams, and course data. The web-based management information system will be efficient, time-saving, straightforward, and simple to use. The system could be used in a lecture series or in exams in the classroom. The user can select the relevant option depending on the required use of the system.

The project will be developed as a web-based application and a mobile web view application. The web-based system allows for system management, report generation, and overall event management via the system's web-based interface. The admin panel is totally built at the administrative end, and thus only the administrator is guaranteed access. The administrator can manage user accounts, student enrollment, timetable scheduling, and fee payments.

The web and mobile applications running on the PC, laptop, tablet, or mobile phone will be used to create all activity plans and capture attendance. At the end of the session, the application will automatically generate a report upon user request. Instructors, lecturers, examiners, principals, directors, and students can access the mobile web app through their user accounts. The instructor can use activity plans to create assignment plans, lesson plans, year plans, and term notes. The examiner can also easily recall the date and time of the examinations using a timetable and notifications. The system will also provide students with access to their personal

information, academic records, and attendance reports. They have the option of paying the course fee online.

The attendance time will be tracked with the help of the web-based management information system. Tracking student attendance helps the instructors keep track of student absenteeism and punctuality. It assists them in determining the average percentage of students attending technical college on any given day during a given year and in discovering ways to avoid dropouts.

The system will provide a user-friendly interface and improve efficiency in managing information. It will be developed using secure programming techniques to protect sensitive information. The system will generate reports and alerts for various events like fee payment reminders, absence notifications, exam schedules, and grades.

Assumptions:

The user should have a basic understanding of IT and the English language because the user interface will be provided in English.

Members must be affiliated with the technical college where the software will be implemented.

1.5 Feasibility Study

The purpose of this feasibility study is to determine the feasibility and viability of implementing a Management Information System for Gampaha Technical College using the Django framework. The project's technical, operational, economic, and schedule feasibility will be assessed in the study.

Technical Feasibility:

The technologies listed below will be used in the project.

Front-end Web Application:

- HTML: Used to build the base interfaces of the system.
- CSS: CSS style is used to make interfaces more user friendly and attractive. Styles basically decided the look of the system.
- Bootstrap: Bootstrap utilizes Sass to create a modular and customizable architecture. Enable global options like gradients and shadows, import only the elements you require,

and use our variables, maps, functions, and mixins to create custom CSS ("Bootstrap · The most popular HTML, CSS, and JS library in the world.," n.d.).

- JavaScript: JavaScript is a scripting language that allows to do just about everything, including creating dynamically updated content and managing multimedia. JavaScript tests in Django can be run in a browser or from the command line. The tests are kept in the top-level directory js_tests.

Back-end:

- Django Framework: Django is a free and open-source, Python-based web framework that follows the model–template–views architectural pattern. It is a high-level Python web framework that encourages rapid development and clean, pragmatic design. It handles a lot of the effort associated with web development, allowing us to concentrate on creating our app without having to invent the wheel ("Django overview | Django," n.d.).
- MySQL: MySQL is a free, open-source relational database management system (RDBMS) that uses a variety of commercial licenses, notably the GNU General Public License (GPL). MySQL, as an RDBMS, employs SQL to handle data within a database. It arranges associated data into one or more data tables, and the correlation aids in data structure (Vyas, 2023).
- OpenCV: OpenCV is a machine learning and computer vision library that is free to use. It supports Java and Python programming language connections. It runs on a variety of operating systems, including Linux, Windows, and others. It primarily focuses on video recording, image processing, and analysis. It has capabilities like face detection 15 and object detection. OpenCV can read and write images, as well as capture and save video. It can recognize features like as faces, automobiles, and photos, among other things ("OpenCV," 2023).
- Face recognition libraries: The facial recognition system is used to train the faces of the employees so that it can recognize their faces as they pass the camera. It can be used as an employee attendance system. The Python libraries dlib and face recognition are used to build the system (Divyansh, n.d.) .

Diagram drawing tools:

- Visio: Microsoft Visio is capable of producing basic or complex diagrams. It comes with a plethora of built-in shapes, objects, and stencils to work with. The main concept of Visio is to make diagramming as simple as possible for the user ("MS Visio," 2012).

Each technology is freely accessible, and the technical skills required are manageable. As a result, the project is technically feasible.

Operational Feasibility:

- The alignment of the system with the technical college's objectives and requirements.
- The ease of use and user-friendliness of MIS for instructors, students, and administrative staff.
- The impact on existing processes and workflows during the transition and adoption of the new systems.
- The ability to effectively manage and administer the Django-based systems on an ongoing basis.

The project is then operationally feasible to launch.

Economic Feasibility:

- The cost of developing, implementing, and maintaining the management information system within the technical college's budget.
- The availability of system resources and tools at a reasonable cost.
- The affordability of necessary hardware, software, and infrastructure upgrades.

As a result, this project is economic feasible.

Schedule Feasibility:

- The estimated time required for developing, testing, and deploying the systems using Django.
- The accessibility of resources to meet project schedules and deadlines.
- The potential impact on existing technical college activities and schedules during the implementation phase.
- The ability to manage and mitigate risks and obstacles during the project implementation.

Then this project is schedule feasible.

Resource Feasibility:

The following are the resources that will be used for the project.

- A laptop Internet connection is used for programming devices.

- Individual programming and testing

This is resource feasible because the project has required resources.

1.6 Structure of the Dissertation

The dissertation is the document that contains overall information about the project, chapter-wise. There are six chapters in this document.

Chapter 01: Introduction

An overview of the project is given in the first chapter to pique the reader's interest. Include the scope, objective, and proposed approach/study as well.

Chapter 02: Background

This will define and explain the background of the project and existing work that is related to the topic and the critical remarks of previous work that has been researched.

Chapter 03: Design:

Using UML Diagrams, this chapter describe Using UML Diagrams, this chapter will describe the module design and the overall system architecture.

Chapter 04: Implementation:

Additionally, describe the techniques and tools, which include general methodologies and useful tools. The chapter also incorporates concepts, any platform dependencies, and the selection of the implementation environment.

Chapter 05: Testing and Evolution:

This includes the work that was done on the system evaluation, testing, and implementation.

Chapter 06: Conclusion:

The last chapter will include a summary of the projects' components as well as any future follow-up work.

1.7 Summary

The first chapter of this thesis essentially discussed how the entire project would be carried out. It demonstrates Gampaha Technical College's current information management process and how that process will be improved as a result of this project.

CHAPTER 2 - BACKGROUND

2.1 Introduction

Manually recording information is a time-consuming process with several issues. From the registration file, instructors call out students' names, and students respond. In the attendance file, a student who is physically absent from class may be marked as represented by a student who routinely marks proxies by responding to a false name. This practice may allow students to falsify their attendance. Furthermore, to take an end-of-semester exam, a student must have attended eighty percent of the lectures during a semester in Sri Lanka's tertiary education system. If a student has less than eighty percent of the required attendance for a subject, they will not be allowed to take the end-of-semester exam for that subject.

In the management information system, the process flow is initiated by being able to detect the face ID from a camera and analyze and compare a picture that is stored in memory. The process automatically calculates the attendance using Face ID. The instructors and examiners can then take an attendance report without wasting time. Students will be able to access their user accounts to view their attendance percentage, which reduces dropout rates and raises educational rates in the technical college. Furthermore, the system allows instructors to create and maintain lesson plans for their courses, and it allows students to pay course fees online. The high-level use case diagram for the current manual system is shown in Figure 2-1 : Manual Information Management System.

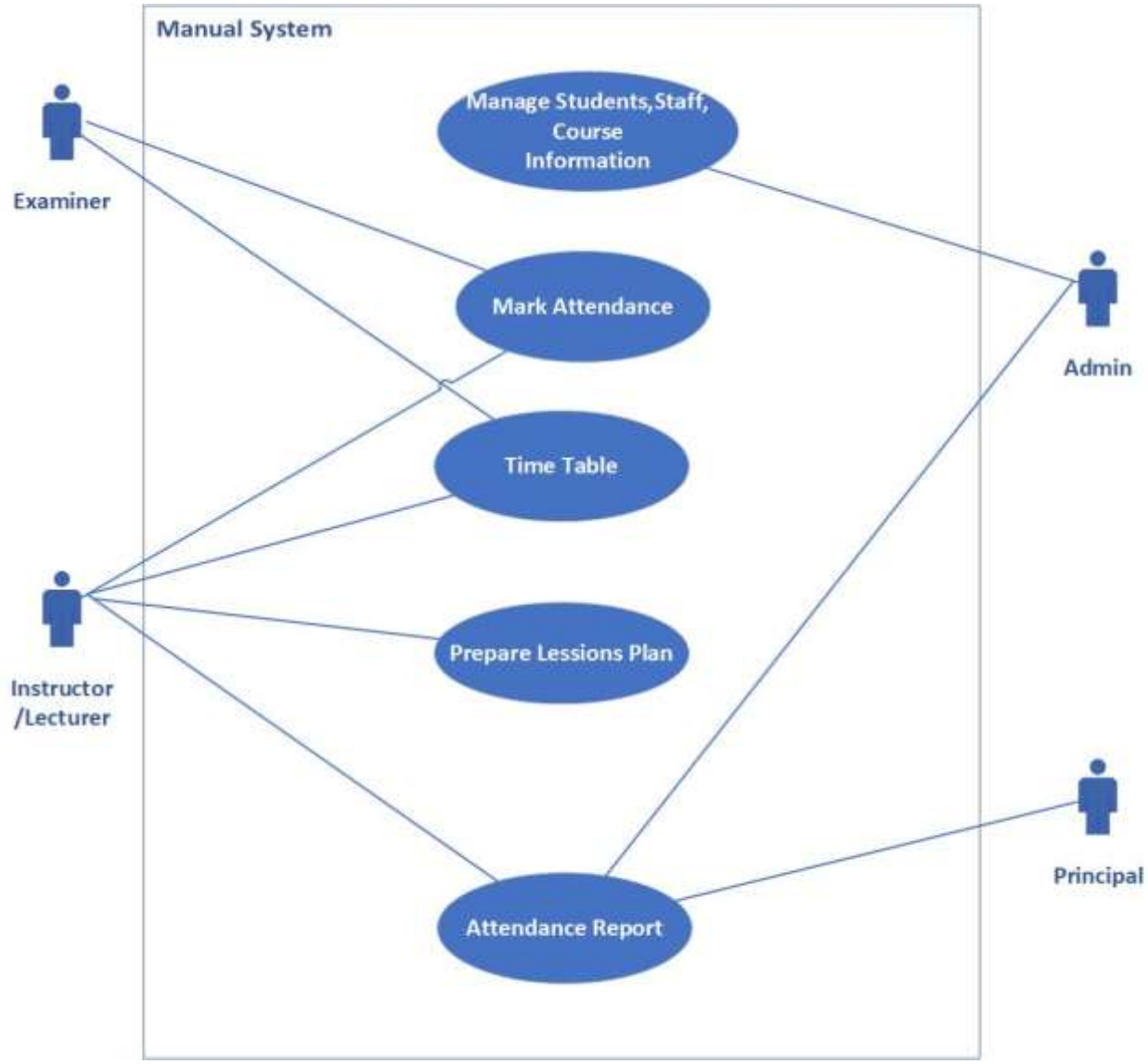


Figure 2-1 : Manual Information Management System

2.2 Data Collection Procedures

Data collection is a crucial step in the development of the Management Information System. The following steps outline the requirement gathering procedure:

Identify Stakeholders:

Identify the key stakeholders involved in the system, such as instructors, students, administrative staff, and management. Determined their roles and responsibilities in relation to the systems and their specific requirements.

Conduct Stakeholder Interview:

Arranged interviews with each stakeholder group to understand their specific requirements and expectations. Asked open-ended questions to gather insights about their pain points, current challenges, and desired functionalities. Then I documented the gathered information, ensuring clarity and accuracy.

Review Existing Documentation:

Evaluated the existing documentation, such as course outlines, fee payment processes, attendance management procedures, and instructor activity plans. Identified gaps and inefficiencies in the current systems and gathered insights on how the new systems can address those issues.

Document Functional and Non-functional Requirements:

Consolidate the gathered requirements and prioritize them based on their importance and feasibility.

Created a detailed requirements document that specifies what the systems should do, such as creating activity plans, accepting online payments, and capturing attendance through face recognition. Also, capture non-functional requirements like security, usability, performance, and scalability criteria.

2.3 Requirement Analysis

The proposed system covers different scopes based on requirements and system features. The first scope is the user scope. Users can be classified as administrators, lecturers or instructors, examiners, principals, and students. A feature of the admin's purview is the management of staff and student information, including the verification of user accounts, student registration, system updates, course payment, and the storage of attendance data. Additionally, provide information on managing courses by department and semester.

A registered student's face ID can also be verified using a management information system. Therefore, the system is used to ensure the validity of student attendance, minimize identity fraud, and minimize unrecorded student attendance. According to the results, the system's key advantage is that it uses Python to obtain images from any type of device camera, allowing for flexibility in hardware selection.

2.3.1 User Classes and Characteristics

- Admin

The Admin is the one who has full privileges on the admin page. This includes the creation, viewing, modification, and deletion of users, as well as the changing of user roles. Additionally, the admin can create, update, view, and delete information about staff, students, courses, trades, and subjects. They can also manage course fee payment transactions, generate reports, schedule timetables, and assign staff.

- Instructor/Lecturer

The instructor, or lecturer, has the privilege of taking attendance of the students. They can view the attendance report and dashboard. Additionally, instructors can set objectives, assign tasks, and set due dates for activities. Instructors can communicate with students about planned activities.

- Examiner

The examiner has the privilege to take student attendance in every exam. They can automatically receive a report on each exam's attendance.

- Principal

Access to view attendance records for each class is granted to the principal.

- Students

Students can access the system to view their attendance as well as the percentage of that attendance and their exam results. The system will notify the user if the attendance falls below 80%.

2.3.2 System Requirements

Guidelines are provided in this document before the system is configured. This is to make it easier for people to install the system on their own devices, including administrators. The selected device must satisfy the following hardware and software requirements in order to install the system.

Hardware Requirements:

Hardware	**PC/ Laptop**	**Mobile Device**
Processor	Intel core i3 processor	Mobile processor
Memory	4GB RAM or more	1GB RAM or more
Hard Disk Space	Minimum 10GB free space or higher	
Internet	Minimum 253 kbps connection	

Table 2-1 : Hardware Requirements

Software Requirements:

Software	**PC/ Laptop**	**Mobile Device**
OS	Windows, Linux	Minimum Android version 4
DB	My SQL	
Web Browser	Internet Explorer, Google Chrome, Fire Fox	

Table 2-2 : Software Requirements

2.3.3 Design Implementation Constraints

- GUI will only be available in English.
- The Management Information System requires an internet connection to work fully.

2.3.4 Functional Requirements

Functional requirements express the required behavior of the system to be built or what the system is supposed to do. Below are the main functional requirements of the proposed system:

- Login

This will allow users to login to the system using their username and password. A user who has a valid username and password can only log in to their respective accounts. It will aid in the authentication of the user who enters the system and provide an additional layer of security by ensuring that only authorized users can log in.

User registration:

- It will be formed like a structure where all the user details will be filled in. The admin has only the privileges to add, update, and delete data in the user registration.

Mange Information:

- The administrator has authority over user accounts, student registration, timetable creation, and course fee collection.
- The admin has privilege to insert new information into the system, update existing information and delete unwanted information from the system.

Course management:

- The technical college offers several different kinds of courses. The administrative staff of the specific department will be in charge of it.

Scheduling:

- The system enables instructors to design and maintain lesson plans for their courses.
- Instructors are able to set objectives, assign tasks, and set due dates for the activities.
- The system provides a platform for instructors to communicate with students regarding activity plans.
- For scheduling lessons, meetings, and events, the system gives instructors access to a calendar or scheduling function.
- The system will allow instructors to monitor the progress of students and provide feedback.
- The system will provide each class time table.

Online Course Fee Payment:

- The system allows students to make online payments for course fees.
- The system accepts payments through various payment gateways.
- The system provides a secure platform for making payments and storing payment information.
- The system provides students with payment receipts and confirmations.

Face Recognition Technology:

- The facial recognition feature can be integrated into the web-based management information system using open-source libraries and frameworks such as OpenCV, dlib, and face recognition to build custom facial recognition features into the application.
- Once the facial recognition feature is integrated into the attendance management system, the system can take attendance by scanning the faces of students.

Search information:

- The system will search for data on attendance, user information, course information, etc.

Attendance management:

- The system uses face recognition technology to identify and authenticate students.
- The system records attendance for each class session and stores it securely.
- Instructors will receive real-time attendance reports from the system.
- The system provides students with their attendance records.
- Instructors will have access to a component in the system that allows them to view attendance reports and export attendance data.
- The system should provide alerts to the administration and instructors for abnormal attendance patterns.

Notifications:

- Instructor's regular activity notification.
- If there is a shortage of attendees, it will be notified by SMS or email. If a student is absent for more than three days, the subject instructor must know when and why the student is absent. So, when a student is absent for more than three days, the system automatically sends an alert to the instructor or lecturer.

Generate report:

- Generate monthly delivery report.
- The system will generate daily, weekly, monthly, semester-wise, and yearly attendance reports for each class.
- Generate student’s Results.

Access dashboard:

- The dashboard will enable a more thorough graphical representation, and a solid overview of the system's condition and prior activities may be tracked and monitored easily using the MIS dashboard.

The functional requirements are important because they define what the system should do and how it should behave (Sommerville, 2011). They provide a clear understanding of the management information system's capabilities and ensure that the system meets the needs of its users. Functional requirements also serve as the basis for designing and testing the system, and they help to identify potential issues or areas for improvement.

2.3.5 Non-Functional Requirements

Non-functional requirements (Chung and Leite, 2009) for the management information system could include:

Security:

By giving the user a login interface, the system is administered. The system can only be accessed by those who have login privileges using their username and password.

The MIS system will be ensuring that student data, payment information, and other sensitive data is kept secure and protected from unauthorized access or hacking attempts.

Performance:

The system will have optimal performance, providing quick response times and efficient processing of user requests.

It will be able to handle peak loads during high traffic periods, such as course registration or fee payment deadlines.

The system's performance will be monitored regularly, and any bottlenecks or performance issues should be addressed promptly.

The system will scale and adapt to accommodate future growth in user base and MIS system requirements.

Usability:

Users could become familiar with the system by attending user training.

The system should be easy to use and navigate, with a simple and intuitive user interface that instructors and students can quickly learn and use.

Processing Time:

The system should be accessible only for academic hours in the technical college.

Reliability:

The system will be highly reliable, ensuring its availability and uninterrupted operation.

It will have a robust backup and recovery mechanism to prevent data loss and minimize downtime in case of system failures or disasters.

The system will be designed to handle a high volume of concurrent users and transactions without degradation in performance.

It will have mechanisms in place to detect and handle errors, ensuring system stability and minimizing user disruptions.

Maintainability:

The MIS system will be easy to maintain, allowing for updates, bug fixes, and enhancements.

The codebase will be well-documented, following coding standards and best practices.

Modularity and separation of concerns will be maintained to facilitate code maintainability.

The system will have version control and release management processes in place.

Portability:

Both mobile and web applications are available on the system. The Web application could be run on a PC, laptop, or mobile device.

2.4 Review of Similar System

After researching Institute information management systems all throughout the world, I came up with the systems listed below, which are used in the schools or colleges.

Radio Frequency Identification-based Attendance System:

Inspired in the current RFID-based system, students must carry a Radio Frequency Identity Card and place it on the card reader to identify their attendance for each day. Unauthorized access is the key issue with these strategies. Some students may use another student's ID to guarantee their presence when that student is absent, or they may even attempt to misuse it (Arulogun et al., 2013).

Fingerprint-based attendance System:

A portable fingerprint device must be configured with the student's fingerprint before using the existing fingerprint-based attendance system. The student must register the fingerprint on the set device later in the day, either during each subject hour or prior, to ensure their attendance for the day. The disadvantage of this strategy is that it may divert students' attention away from the lesson (Li and Kot, 2012).

Iris-Recognition Based Attendance System:

The student must stand in front of a camera in order for the camera to scan the student's Iris in the Iris-based student attendance system. The features of the iris were retrieved and matched with student data contained in the database, and the attendance on their presence needed to be updated. It is built on a real-time facial detection technology that is extremely secure, dependable, and quick to access, but it still requires a lot of work in varied lighting circumstances (Khatun et al., 2015).

Face Based Recognition Attendance System:

- Facial recognition attendance system with a CCTV camera installed at the classroom door. This captures the individual's image and compares it to the face database. It serves two purposes. making a comparison with the database to record student attendance while identifying an unauthorized person. Utilizing 3D face recognition, which increases the accuracy of matching images in the database, the image is verified. The system's primary flaw is that it compares a user's recognized face to the complete database to verify their attendance. This will be time-consuming and expensive (MuthuKalyani.K and VeeraMuthu.A, 2013).
- Raspberry Pi-based face recognition for attendance records in classes. By comparing the input image from the recorded video frame to the trained image, it offers a way to recognize

the person. Both LBP and Haarcascaed are used to extract features from the facial image. By producing reliable results, the method created to merge LBP and deep learning neural networks can stabilize the system. The issue is that this technique still requires some manual work (Al-Badri and Hasani, 2019).

Online Course Management System:

This application provides students with a convenient and helpful feature for their study purposes. It has three panels: an admin panel, a teacher panel, and a student panel. Students can use the course code to access their online classroom, post papers and other assignments, comment on posts, submit assignments, and view the total number of students in their classroom. Administrators can add any department, teacher, course, and others, check the current state of all the portals, and access all of them (Popy, 2020).

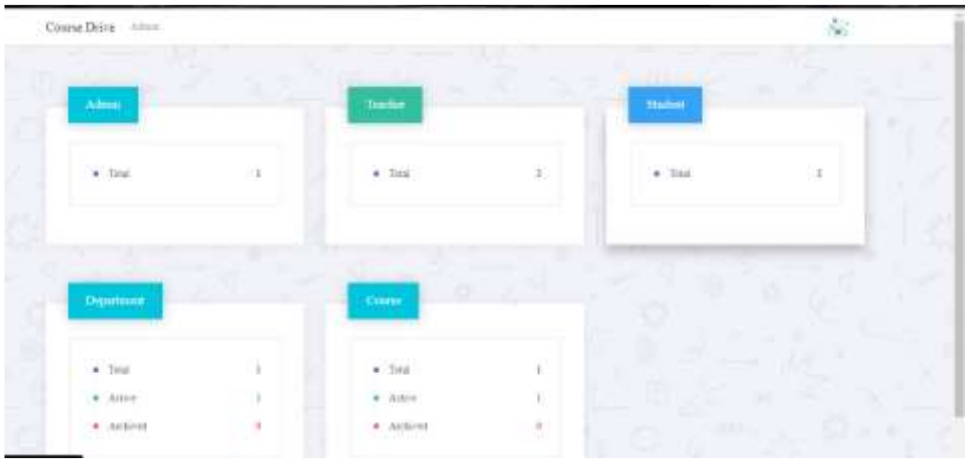


Figure 2-2 : Online course management system admin panel

Similar work comparison:

Features	Similar work						Proposed System
	RFID	Finger print	Iris	Smart application AMS using FR	Class attendance record-based FR	Online management system	Management Information System
User Login	×	✓	×	×	×	✓	✓
Information management	✓	✓	✓	✓	×	✓	✓
Face Recognition attendance	×	×	×	✓	✓	×	✓
Course management	×	×	×	×	×	✓	✓
Scheduling	×	×	×	×	×	✓	✓
Notification	✓	×	×	✓	×	×	✓
Online Course payment	×	×	×	×	×	×	✓
Report	✓	✓	✓	✓	✓	✓	✓
MIS dashboard	×	×	×	×	×	✓	✓

Table 2-3 : Comparison of similar works

2.5 Related Technology

Web Application Front End:

Django:

When considering the development of the Management Information System for Technical College, there are several reasons why Django stands out as a preferred framework over other methods. The following justifications highlight the benefits of Django in this context:

Rapid Development: Django follows the "Don't Repeat Yourself" (DRY) principle and provides a high level of abstraction, which significantly speeds up the development process (B,

2022). It offers a comprehensive set of built-in features, libraries, and tools that facilitate the creation of web applications, reducing the need for manual configuration and boilerplate code.

Security: Django incorporates numerous security features and follows best practices, making it a secure framework for application development. It provides protection against common vulnerabilities such as cross-site scripting (XSS), cross-site request forgery (CSRF), and SQL injection. Django's security features enable developers to focus on application-specific security concerns, ensuring a robust and secure system.

Scalability: Django's scalability is well known, enabling the system to handle increasing workloads and accommodate a growing user base. It incorporates a modular and component-based architecture that allows for the addition of new functionalities and the optimization of existing ones without impacting the entire system. Django's scalability ensures the Instructor's Activity Plans, Online Course Fee Payment, and Student Face Recognition Information Management System can effectively handle future growth.

ORM and Database Support: Django's Object-Relational Mapping (ORM) layer abstracts the database interactions, making it easier to work with databases. It supports multiple databases, including popular choices like PostgreSQL, MySQL, and SQLite. The ORM provides an intuitive and efficient way to query and manipulate data, simplifying the database-related tasks in the system.

Admin Interface: Django offers a powerful and customizable admin interface, which can significantly reduce the development time for administrative features. The admin interface provides a ready-to-use content management system, allowing administrative staff to manage various aspects of the system, including user management, activity planning, fee payment tracking, and attendance management.

Community and Documentation: Django has a large and active community of developers, providing extensive support, resources, and third-party libraries. The community actively maintains and updates the framework, ensuring continuous improvement and bug fixes. Django also offers comprehensive documentation, making it easier for developers to learn and utilize its features effectively.

Versatility: Django is a versatile framework that can handle diverse requirements. It supports the development of both simple and complex web applications, making it suitable for the Instructor's Activity Plans, Online Course Fee Payment, and Student Face Recognition

Information Management System. Django's flexibility allows for customization and integration with various external systems, ensuring seamless functionality across different modules of the system.

Time and Cost Efficiency: Django's rapid development capabilities, extensive feature set, and built-in functionalities significantly reduce the time and effort required for system development. Its open-source nature also makes it a cost-effective choice for institutes that have budget constraints.

Overall, Django's combination of rapid development, robust security, scalability, database support, admin interface, community support, versatility, and cost-efficiency make it the ideal framework for implementing the Instructor's Activity Plans, Online Course Fee Payment, and Student Face Recognition Information Management System. Its strengths align well with the requirements of the Technical College, ensuring a stable, secure, and efficient system (“Django,” n.d.).

Face Recognition Libraries:

Face recognition is the process of recognizing individuals from facial photographs. Using either specialized libraries or a Python library that is already built-in, it can detect faces (Geitgey, n.d.).

There are several reasons why face recognition libraries are better than other methods for this project:

Simplified development process: Face recognition libraries simplify the development process by providing pre-built functionality for detecting and recognizing faces. This saves developers a lot of time and effort compared to developing facial recognition capabilities from scratch.

Accurate and reliable: Face recognition libraries are designed to be accurate and reliable, which is critical for a facial recognition attendance management system. They use advanced algorithms and machine learning techniques to ensure that faces are correctly detected and recognized, even in challenging conditions such as low light or partial occlusion.

Easy to integrate with Django: Most face recognition libraries are designed to be easy to integrate with Django web applications. They typically provide Python bindings that allow for write Python code to interact with the library and incorporate facial recognition capabilities into their Django web application.

Good documentation and community support: Face recognition libraries have a large and active community of developers and users, which means that there is a wealth of documentation and support available online. This makes it easier to learn how to use the library and troubleshoot any issues that may arise during development.

Efficient and scalable: Face recognition libraries are designed to be efficient and scalable, making them ideal for use in real-time facial recognition systems. Various hardware platforms and specialized hardware accelerators can be used to optimize them for performance.

OpenCV:

OpenCV is a machine learning and computer vision library that is free to use. It supports Java and Python programming language connections. It runs on a variety of operating systems, including Linux, Windows, and others. It primarily focuses on video recording, image processing, and analysis. It has capabilities like face detection 15 and object detection. OpenCV can read and write images, as well as capture and save video. It can recognize features like as faces, automobiles, and photos, among other things (Team, 2021).

Simple to integrate with Django: Because Python bindings are available, OpenCV can be easily integrated with Django web applications. It follows that programmers can use Python to interact with the OpenCV library and add facial recognition functionality to their Django web application.

Good documentation and community support: OpenCV have a large and active community of developers and users, which means that there is a wealth of documentation and support available online. This makes it easier for developers to learn how to use the library and troubleshoot any issues that may arise during development.

Efficient and scalable: OpenCV is designed to be efficient and scalable, which makes it ideal for use in real-time facial recognition systems. It can be optimized for performance on a range of hardware platforms, and specialized hardware accelerators.

These factors make it an attractive option for Django web applications that incorporate facial recognition capabilities into the system.

HTML:

When it comes to developing a Django system, HTML (Hypertext Markup Language) is an essential and widely used tool (“Django Template,” n.d.) for several reasons .

Structure and Presentation Separation: HTML allows for clear separation of content structure and presentation. It provides a clean and organized way to define the structure of web pages using elements like headings, paragraphs, lists, and tables. With the use of CSS (Cascading Style Sheets), HTML can control the visual presentation, ensuring consistent styling across the application.

Integration with Django Template Language: Django, as a web framework, seamlessly integrates with HTML through its template language. HTML templates allow for dynamic content rendering by embedding Django template tags and variables within HTML code. This integration simplifies the process of rendering dynamic data and generating HTML responses, making development more efficient.

Browser Compatibility: HTML is a standard markup language supported by all modern web browsers. It ensures consistent rendering and compatibility across different devices and platforms, providing a reliable user experience. By following HTML standards, developers can avoid compatibility issues and ensure broad accessibility to their Django system.

Wide Developer Community and Resources: HTML has been around for decades and has a vast and active developer community. There are abundant online resources, tutorials, and documentation available for HTML development. Developers can easily find solutions, get support, and stay updated with best practices, making HTML a reliable choice for Django system development.

Flexibility and Extensibility: HTML offers flexibility in terms of integrating with other technologies and frameworks. It can be combined with JavaScript frameworks like React or Vue.js for enhanced interactivity and dynamic behavior. HTML's extensibility allows developers to incorporate libraries, frameworks, or custom scripts to enhance the functionality of Django systems.

Separation of Concerns: HTML promotes the separation of concerns by focusing on the structure and presentation of content. It allows developers to separate their logic (Python code) from the presentation layer (HTML), ensuring clean and maintainable codebases.

This separation makes it easier to collaborate, troubleshoot, and modify different parts of the Django system independently.

CSS:

CSS (Cascading Style Sheets) is a powerful and widely used method for styling web pages, including Django systems (“Django CSS,” n.d.).

Separation of Concerns: CSS allows for a clear separation of concerns between the structure (HTML), behavior (JavaScript), and presentation (CSS) of a web page. This separation promotes a more maintainable and modular codebase, making it easier to update and modify the visual aspects of the Django system without impacting the underlying functionality.

Consistency and Reusability: By utilizing CSS, you can define styles once and apply them consistently across multiple pages or elements within the Django system. This reusability reduces code duplication and ensures a consistent and professional appearance throughout the system, enhancing the user experience.

Flexibility and Customization: CSS provides extensive flexibility and customization options for styling web pages. It offers a wide range of selectors, properties, and values that allow developers to target specific elements or groups of elements and modify their appearance according to specific requirements. This flexibility empowers developers to create unique and visually appealing designs tailored to the Django system's needs.

Efficiency and Performance: CSS allows for lightweight and efficient styling of web pages. By separating the presentation layer from the content, CSS files can be cached by browsers, reducing the amount of data transferred between the server and the client. This caching mechanism improves page load times, resulting in better performance and user satisfaction.

Responsive Design: CSS plays a crucial role in creating responsive web designs that adapt to different screen sizes and devices. Using media queries and responsive CSS techniques, developers can ensure that the Django system looks and functions optimally on various platforms, such as desktops, tablets, and mobile devices. This responsiveness improves the system's accessibility and user experience across different devices.

Browser Compatibility: CSS enjoys excellent browser support and compatibility. It has become a standard language for web styling, and most modern browsers adhere to CSS specifications. This broad compatibility ensures that the Django system's visual presentation remains consistent across different browsers, reducing the need for extensive cross-browser testing and fixes.

Integration with Django: CSS seamlessly integrates with Django, allowing for efficient and organized management of stylesheets within the project structure. Django provides support for static files, including CSS, making it easy to reference and serve CSS files within the system. Additionally, Django's template system allows for the dynamic inclusion of CSS classes or inline styles based on specific conditions, enhancing the system's flexibility.

While other methods, such as inline styles, have their use cases, CSS stands out as the preferred method for styling Django systems due to its separation of concerns, consistency, reusability, flexibility, performance, responsiveness, browser compatibility, and seamless integration with Django. Leveraging CSS empowers developers to create visually appealing and maintainable Django systems that meet modern web design standards and provide an exceptional user experience.

Bootstrap:

Bootstrap, a popular front-end framework, offers several advantages over other methods (Team, 2019) when used in conjunction with Django for the development of the Management Information System for the Technical College.

Responsive Design: Bootstrap provides built-in responsive design features, allowing the system to adapt and provide a consistent user experience across various devices and screen sizes. With Bootstrap, the user interface automatically adjusts to fit desktops, laptops, tablets, and smartphones, ensuring usability on different platforms.

Time and Cost Efficiency: Bootstrap offers a wide range of pre-designed and customizable UI components, such as buttons, forms, navigation bars, and modals. By utilizing these ready-to-use components, developers can save time on designing and coding from scratch, accelerating the development process and reducing development costs.

Consistent and Professional Look: Bootstrap follows a consistent and visually appealing design language, providing a polished and professional look to the system. The framework offers a set of predefined styles, typography, and color schemes, ensuring a cohesive and harmonious user interface throughout the system.

Customization Flexibility: Bootstrap provides extensive customization options, allowing developers to tailor the design to match the Technical College's branding and specific requirements. Developers can easily modify the framework's CSS classes and utilize its powerful grid system to create unique layouts and designs.

Cross-Browser Compatibility: Bootstrap is developed and tested to be compatible with a wide range of web browsers, including Chrome, Firefox, Safari, and Internet Explorer. By utilizing Bootstrap, developers can ensure a consistent user experience and minimize the need for browser-specific adjustments or fixes.

Active Community and Support: Bootstrap has a large and active community of developers and designers, providing access to a wealth of resources, documentation, and community support. Any issues or challenges faced during development can be easily addressed through online forums, tutorials, and community-driven support channels.

Integration with Django: Bootstrap seamlessly integrates with Django, enabling developers to leverage the power of both frameworks together.

There are several Django packages and libraries available that provide easy integration with Bootstrap, simplifying the development process and ensuring compatibility.

Java Script:

JavaScript is a widely used programming language with a robust ecosystem that offers several advantages for developing web applications, including those built with Django ("Django JavaScript," n.d.).

Client-Side Interactivity: JavaScript is executed on the client-side, allowing for dynamic and interactive user experiences. This is particularly useful in the management system as it enables real-time updates, form validation, and other client-side interactions without the need for server round trips. It enhances the responsiveness and interactivity of the system, providing a smoother user experience.

Asynchronous Communication: JavaScript, particularly when used with AJAX (Asynchronous JavaScript and XML), enables asynchronous communication with the server. This means that requests can be made to the server without reloading the entire page, allowing for faster and more efficient data retrieval and submission. It enhances the performance and efficiency of the system, reducing page reloads and improving the overall user experience.

Rich UI Components and Libraries: JavaScript has a vast array of UI libraries and frameworks, such as React, Vue.js, and Angular, that provide ready-to-use components and tools for building modern and visually appealing user interfaces. These libraries offer features like data

binding, state management, and component reusability, which can significantly streamline the development process and improve the aesthetics of the system.

Integration with Third-Party APIs and Services: JavaScript's flexibility allows for seamless integration with various third-party APIs and services. In the context of the management system, this can be beneficial for integrating payment gateways, face recognition services, and other external functionalities. JavaScript's versatility makes it easier to consume APIs and handle data exchanges, expanding the capabilities and functionality of the system.

Cross-Platform Compatibility: JavaScript is supported by all major web browsers, making it a highly cross-platform compatible language. This means that the Instructor's Activity Plans, Online Course Fee Payment, and Student Face Recognition Information Management System developed using JavaScript can be accessed and used by users on different devices and platforms, including desktops, laptops, tablets, and smartphones. It ensures a broader reach and accessibility for the system's users.

Developer Community and Resources: JavaScript has a vast and active developer community, which translates into a wealth of resources, documentation, and community support. This can be particularly advantageous during the development process, as developers can rely on community-driven solutions, code examples, and discussions to overcome challenges and find best practices. The availability of resources and community support reduces development time and promotes code quality.

Back End:

MySQL: MySQL, a popular open-source relational database management system (RDBMS), offers several advantages that make it a suitable choice for the Django Management Information System for the Technical College.

Wide Adoption and Community Support: MySQL has a large and active user community, making it easier to find resources, documentation, and support. Its widespread adoption ensures that developers are familiar with MySQL, making it easier to find skilled professionals for maintenance and troubleshooting.

Compatibility with Django: Django, a high-level Python web framework, has excellent support for MySQL out of the box. Django provides a built-in MySQL database connector, making it seamless to integrate and work with MySQL in the Django Management System. The compatibility ensures smooth data interactions and efficient development processes.

Performance and Scalability: MySQL is known for its excellent performance, especially when handling large datasets and complex queries. It offers robust indexing capabilities, query optimization techniques, and efficient caching mechanisms, resulting in faster response times. MySQL can handle high concurrency and scale horizontally to accommodate increasing data and user loads, making it suitable for the Technical College management system.

Reliability and Stability: MySQL has a proven track record of stability and reliability, with many years of production use in various industries. It has undergone extensive testing, bug fixing, and performance optimizations, making it a dependable choice for critical systems. MySQL's data integrity features, such as transaction support and ACID compliance, ensure reliable and consistent data management.

Flexibility and Feature Set: MySQL offers a wide range of features and functionality to support diverse application requirements. It supports various data types, including numeric, text, date/time, and spatial data, allowing for flexibility in data modeling. MySQL provides advanced features like stored procedures, triggers, and views, enabling complex database operations and business logic implementation.

Cost-Effectiveness: MySQL is open-source, meaning it is freely available and does not require any licensing fees. This makes it a cost-effective choice for the Technical College, allowing them to allocate their resources efficiently without compromising on functionality or performance.

While there are other database management systems available, MySQL's combination of compatibility with Django, performance, scalability, reliability, flexibility, and cost-effectiveness makes it a favorable choice for the Management Information System in the Technical College setting (Vyas, 2023).

Excel: Django-excel, which is built on PyExcel, makes it simple to consume or produce data saved in Excel files over the HTTP protocol as well as on the file system. This library can convert Excel data into a list of lists, a list of records (dictionaries), and dictionaries of lists. And the other way around (“django-excel,” n.d.). The Management Information System will be managed and reports will be generated using Excel application.

2.6 Summary

This chapter essentially demonstrates the Management Information System's current process. Gampaha Technical College's functional and non-functional requirements, as well. Furthermore, in comparison to other systems currently on the market.

CHAPTER 3 - DESIGN ARCHITECTURE

3.1 Introduction

This chapter aims to present the system design for the proposed management information system at Gampaha Technical College. It gives an overview of system analysis and system design. Requirements gathering and analysis are significant practices for a successful project. The main processes of this phase include domain understanding, requirement collection, classification, structuring, prioritization, and validation. Appropriate methods and processes were engaged to carry out the analysis phase in an effective way.

The software design phase is an iterative process in which the requirements acquired during analysis are turned into a "blueprint" for developing the system (Pressman, 2000). These can then be further developed into specific functional and behavioral requirements. The system specification is the end result of the design phase. As the first step of the product design, this chapter comprises the system architecture and its related UML diagrams, such as the use case diagram, class diagram, sequence diagram, and activity diagram. The chapter will also assist in defining the key structural elements of the system and their interrelationships.

3.2 Related System Design Strategy

The design strategy is the method used to model the system. There are various design methodologies, including rapid application development, modern structured design, and prototyping. Object-oriented design was selected above others due to its code reuse and recycling capabilities, design advantages, and maintainability with Objects and classes.

A key component of object-oriented design is the use of the Unified Modeling Language (UML). UML enables programmers to easily comprehend object models, allowing them to develop software more efficiently (Larman, 2012) . The below diagram lists a few structural and behavioral UML diagrams that are used to design the Management Information System.

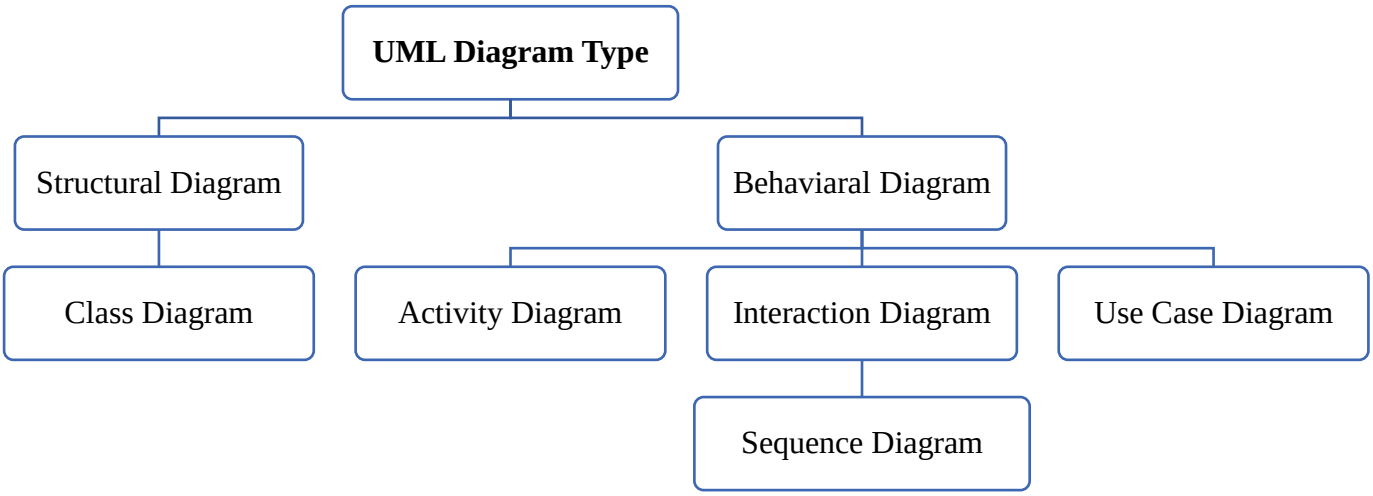


Figure 3-1： MIS Related UML Diagram Hierarchy

3.3 System Architecture

The system development takes advantage of the Model View Template architecture. The software design pattern known as MVT, which represents Model View Template, is used for creating web applications (“Django Project MVT Structure,” 2019). It splits a given software program into three interrelated parts and has full support for both rapid web application development and dynamic database interaction. Below, Figure 3-2 : Proposed system architecture depicts the proposed system architecture(Nige, 2022).

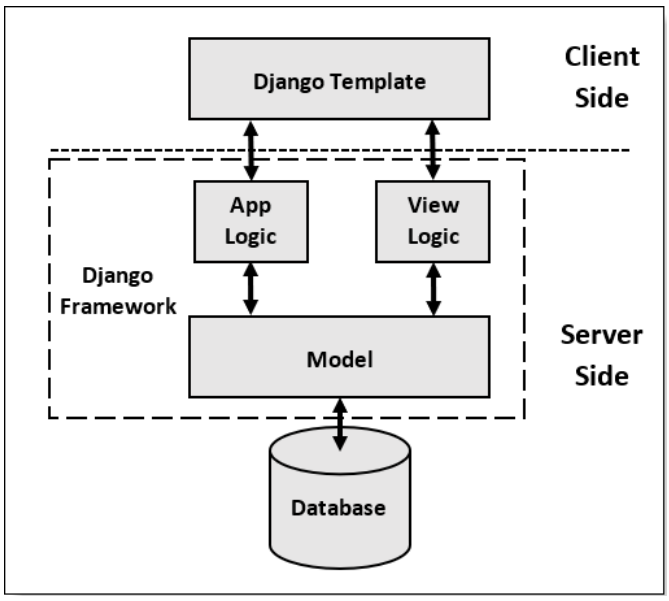


Figure 3-2： Proposed system architecture

The three components of the MVT structure are as follows:

- Model: The model is the component of the management information system application that serves as a liaison between the database and the internet interface. It is represented by a database and operates as the logical data structure for the entire application (generally relational databases such as MySQL and SQLite). The data domain logic for the application is implemented by this object. When an application just uses data from a certain dataset and sends it directly to the view (UI component) without the need for a database, that dataset is referred to as a model.
- View: This is the point where we would actually be mentioning logic. This coding is done through the python file views.py. This view also sends responses to the user when the application is used. View.py can be linked to the urls.py file, which keeps track of all the pages that are created and uses that list to map each one.
- Template: A template includes both static elements of the desired HTML output and unique syntax that specifies where dynamic content will be included. The configuration of the template is done in the settings.py file under INSTALLED_APPS. Therefore, the template subdirectory's files would be searched for through the Python code.
- Security: A component responsible for ensuring the security of the management information system, including protecting sensitive information stored in the database and controlling access to the system. It includes user authentication and authorization mechanisms as well as database encryption.
- Reporting: A component that generates reports based on data stored in the database, including attendance reports, performance reports, and other reports as needed.
- Integration: A component that integrates with other systems used by the technical college, such as user information systems, activity plan management, online payment systems, and attendance management, to ensure seamless and efficient data flow between systems.

Django uses the request and response objects to communicate between the client and the server. The request object is used to receive data from the client, while the response object is used to send data back to the client. This communication happens over the HTTP protocol, which is a standard way for web servers and clients to exchange data.

By using request and response objects, Django provides a simple and consistent way to handle HTTP requests and responses. This makes it easier to develop web applications that can communicate effectively with clients and provide a seamless user experience.

3.4 UML Diagrams for Proposed System

3.4.1 Use case Diagram

The use case model represents an interaction between an actor and the system. The system's functionality is represented by use cases. All the defined ways a system may be used are represented by the collections of use cases for that system. The key project actors are depicted in Table 3-1.

Actor	Description
Admin	The Person, who is responsible for managing the overall system and has the authority to perform administrative tasks such as system configuration, user management, and access control.
Principal	The person who has access to view attendance records for each class.
Instructor/ Lecturer	The person, who create activity plans, take student attendance, and interact with students.
Examiner	The person who has privilege to take student attendance in every exam.
Student	The person, who enroll in courses offered by the Technical college. They interact with the system to access course materials, make payments, and view attendance.

Table 3-1 : Main Actors of the MIS

To develop a use case, first define the various sorts of actors who interact with the system. The roles that people perform while the system works are literally represented by these actors. The Administrator, Instructor/Lecturer, Examiner, Principal, and Student are the five actors in this system. These use cases describe the primary functionalities and interactions within the management information system, allowing the actors to complete their responsibilities and achieve their objectives. Diagram in Figure 3-3 depicts the overall high-level Use case of the proposed system.

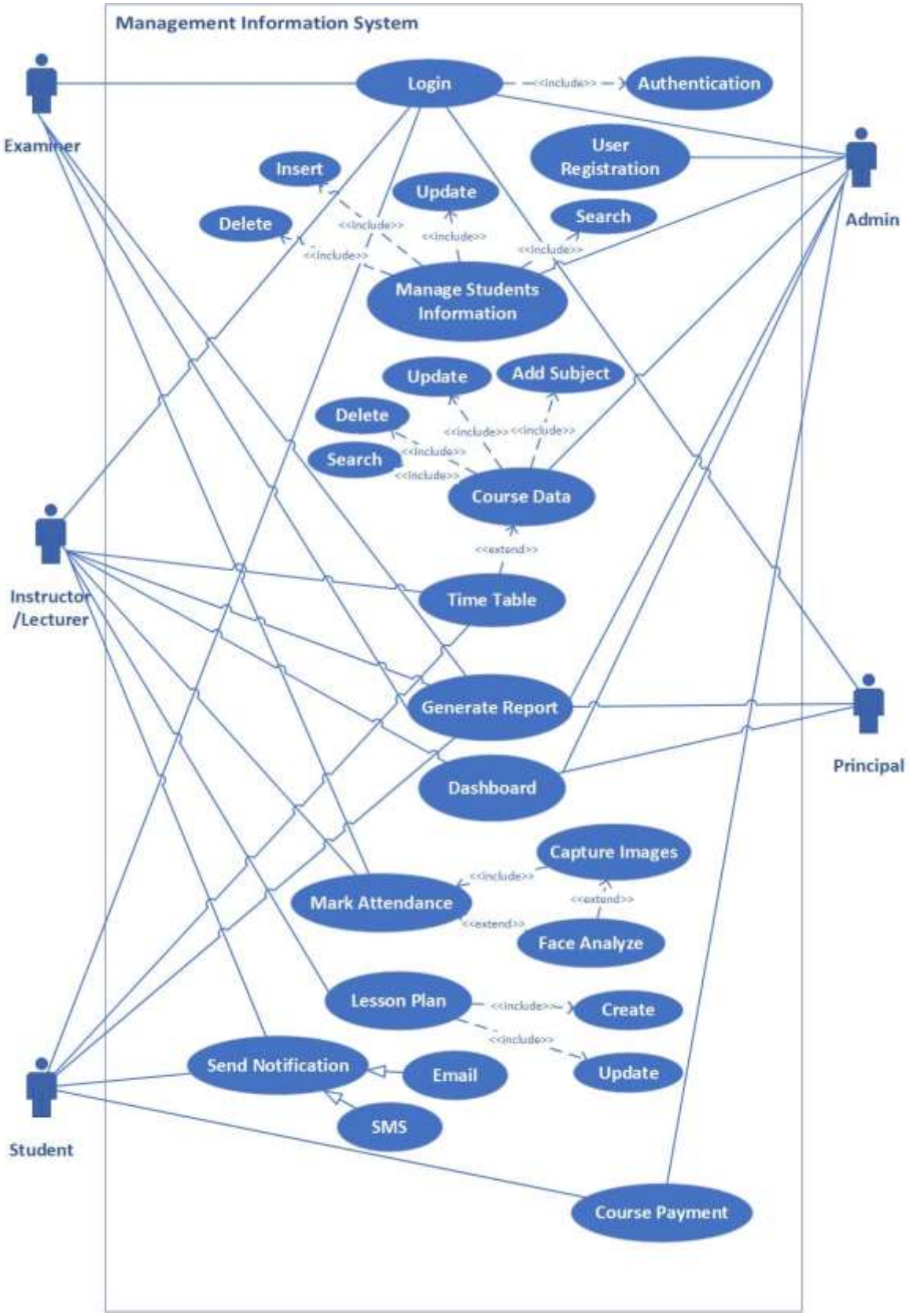


Figure 3-3 : Use case diagram of the proposed system

Use case Narratives:

The system's functions define the tasks performed by the Management Information System.

Use Case	User Registration
Actors	Admin
Description	
The administrator can add new users to the system. Create different user accounts. Set a User Name and Password. Give access to the different users.	
Pre-Condition	
If users are not existed register users to the system.	
Flow of Event	
1. The admin should be login to the System 2. Select user registration 3. Fill information required 4. Click user registration button	
Post Condition	
Confirm registration.	

Table 3-2 : Use-Case Narrative for User Registration

Use Case Name:	Login
Actors	Admin, Instructor/ Lecturer, Examiner, Student, Principal, Student
Description	
This use case allows access to the relevant functions according to the user's role. The various user roles are student, instructor, lecturer, examiner, and principal. Using a user name and password, users can login to the system.	
Pre-Condition	

Actor has to have a valid account.
Flow of Event
1. The user enters valid username and password 2. The system validates user credentials 3. User credentials are correct user can logon to the system
Post Condition
System displays the relevant user interfaces.

Table 3-3 : Use-Case Narrative for Login

Use Case	Add Data
Actors	Admin
Description	
Using the web application, the administrator can add the trade, course, subject data to the database.	
Pre-Condition	
The admin login to the system. The admin can enter the technical college's upcoming trades, courses, and subjects.	
Flow of Event	
1. The admin should be logged into the system. 2. Select the required menu 3. Fill in the information required and click the add button	
Post Condition	
Store in the database.	

Table 3-4 : Use-Case Narrative for Course Data

Use Case	Update Data
Actors	Admin

Description
All of the data in the system's web app can be updated by the administrator.
Pre-Condition
The admin should be logged into the system. If any pertinent information has to be modified, update the details in the system.
Flow of Event
1. The admin should be logged into the system. 2. Select the required menu 3. Update the information required and click the add button
Post Condition
Confirm updating details.

Table 3-5 : Use-Case Narrative for Update Data

Use Case	Delete Data
Actors	Admin
Description	
Data in the web-based application can be deleted by the system administrator.	
Pre-Condition	
The admin must be logged in. Remove data from the database.	
Flow of Events	
1. The admin should be logged into the system. 2. Select the required menu 3. Select the data that needs to be deleted and click the add button	
Post Condition	
Confirm deleting details.	

Table 3-6 : Use-Case Narrative for Delete Data

Use Case	Search Details
Actors	Admin, Principal, Instructor/ Lecturer
Description	
The instructor, lecturer, principal, or admin can be able to search for student attendance from the system.	
Pre-Condition	
Search for the required details from the system.	
Flow of Event	
1. The user logging to the system 2. Select search bar 3. Enter the relevant information and hit the search button.	
Post Condition	
If there is an internet connection. Student attendance was found successfully. Display search details	

Table 3-7 : Use-Case Narrative for Search Details

Use Case	Attendance Data
Actors	Instructor, Lecturer, Examiner
Description	
The instructor, lecturer, or examiner could be able to take student attendance using the MIS application.	
Pre-Condition	
By comparing the student's profile image, device cam analyzes their face ID.	
Flow of Events	
1. The Instructor or Examiner must be logged in 2. Go to the attendance menu 3. Click on the start recognition button 4. Take attendance	
Post Condition	
Confirm attendance details.	

Table 3-8 : Use-Case Narrative for Attendance Data

Use Case	Lesson Plan
Actors	Instructor, Lecturer
Description	
The instructor, lecturer, could be able to manage lesson plan using the MIS application.	
Pre-Condition	
Instructors can set lesson plan, assign tasks, and set due dates for activities.	
Flow of Events	
1. The instructor or lecturer must be logged in 2. Select the Lesson Plan menu 3. Select the subject modulation and scheduling 4. Make lesson plan	
Post Condition	
Confirm lesson plan details.	

Table 3-9 : Use-Case Narrative for Lesson Plan

Use Case	Course fee payment
Actors	Admin, Student
Description	
Students could be able to make payment using the MIS application. Admin could be able to manage course fee payments.	
Pre-Condition	
Students allows to make online payments for course fees. Admin receive notification and manage payment data.	
Flow of Events	
1. Student must be logged in 2. Go to the course fee payment menu 3. Select the payment mode 4. If the payment validation is successful, confirm the payment 5. Admin must be logged in 6. Receive notification of payment	
Post Condition	

Confirm payment details.

Table 3-10 : Use-Case Narrative for Course Fee Payment

Use Case Name:	Send Notification
Actors	Admin, Principal, Instructor/ Lecturer, Examiner, Student
Description	
If a student's attendance falls below 80%, the application will send them an email and a SMS.	
Pre-Condition	
Notify the student, instructor, and admin.	
Flow of Events	
1. The User must be logged in 2. The system sends the appropriate notification to the user 3. The user should be able to read the message	
Post Condition	
Send notification by SMS or by Email.	

Table 3-11 : Use-Case Narrative for Send Notification

Use Case Name:	Generate Report
Actors	Admin, Principal, Instructor, Lecturer, Examiner
Description	
Request selected report and generate report.	
Pre-Condition	
The users can select the relevant report.	
Flow of Event	
1. The user must be logged in 2. The user enters the appropriate fields needed to generate the report.	

3. When viewing the report, the user can print it as a pdf or an excel sheet.
Post Condition
Print/ Download/ Display the generated report.

Table 3-12 : Use-Case Narrative for Generate Report

Use Case Name:	Dashboard
Actors	Admin, Principal, Instructor, Lecturer
Description	
Through charts, the user may see student performance, and attendance progress by daily and monthly.	
Pre-Condition	
Login to the system to see student performance, daily and monthly attendance.	
Flow of Event	
1. The user must be logged in 2. Select the dashboard	
Post Condition	
Display the attendance progress.	

Table 3-13 : Use-Case Narrative for Dashboard.

3.4.2 Sequence Diagram

In the UML, sequence diagrams are generally used to model interactions between actors and objects in a system, as well as interactions between objects themselves (Sommerville, 2011). A variety of interactions can be described according to UML's extensive syntax for sequence diagrams.

The object interactions depicted in the sequence diagrams below are grouped in the time sequence of the MIS at Gampaha Technical College. It depicts the objects engaged in the scenario as well as the sequence of messages sent between the objects required to carry out scenario functionality.

3.4.2.1 Admin user registration and manage information

Figure 3-4 : Sequence Diagram – User Registration and Information management

3.4.2.2 Lecturer take attendance and manage lesson plans

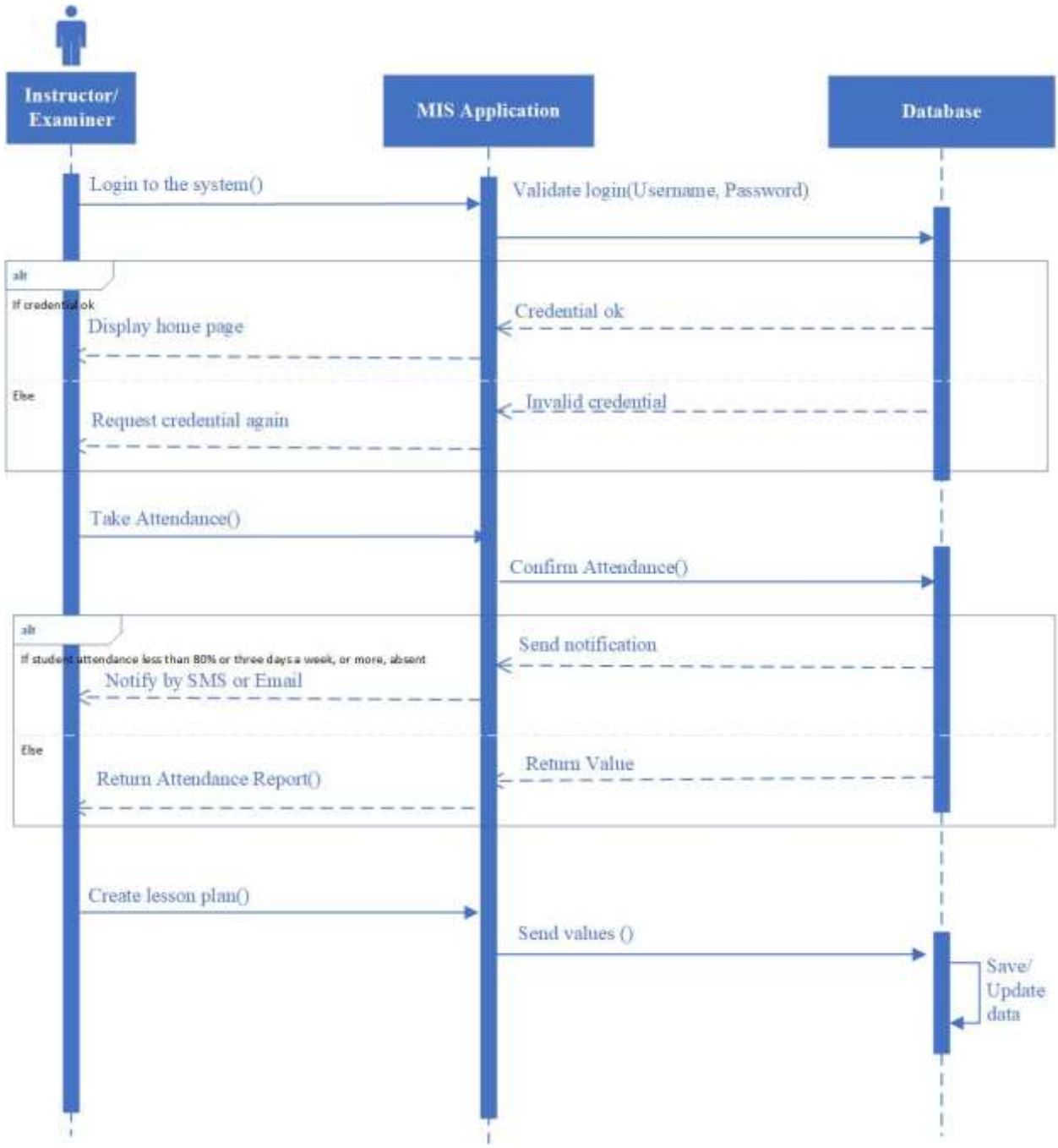


Figure 3-5 : Sequence Diagram – Take Attendance and Manage Lesson Plans

3.4.2.3 Student course fee payment

Figure 3-6 : Sequence Diagram – View attendace and Make course fee payment

3.4.2.4 Generate Reports

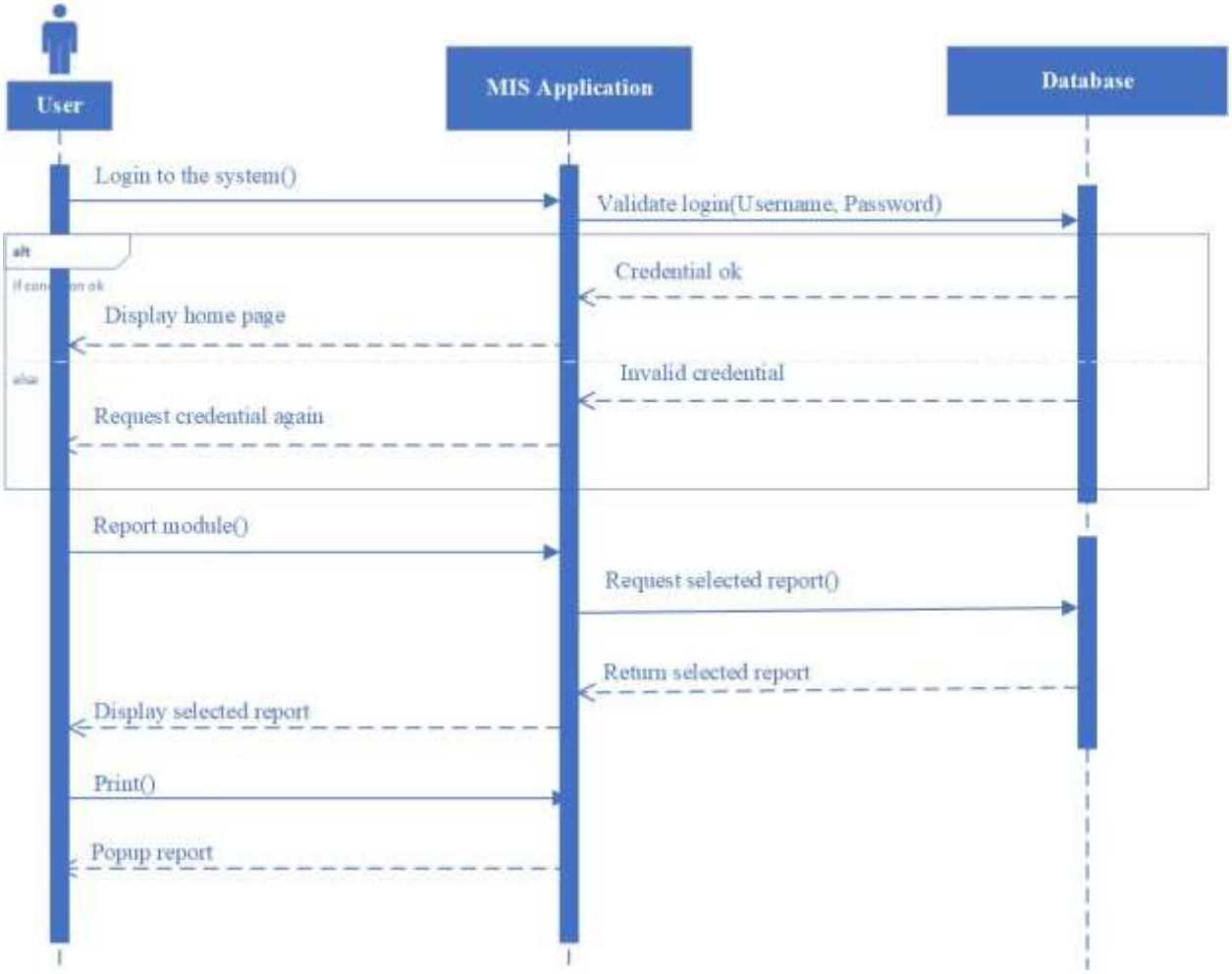


Figure 3-7 : Sequence Diagram – Generate Reports

3.4.3 Activity Diagram

Activity diagrams are designed to depict the activities that comprise a system process as well as the flow of control from one activity to the next. A filled circle denotes the beginning of a process, whereas a filled circle inside another circle denotes its conclusion. The flow of activities at Gampaha Technical College is illustrated in the activity diagrams below.

3.4.3.1 Manage Information

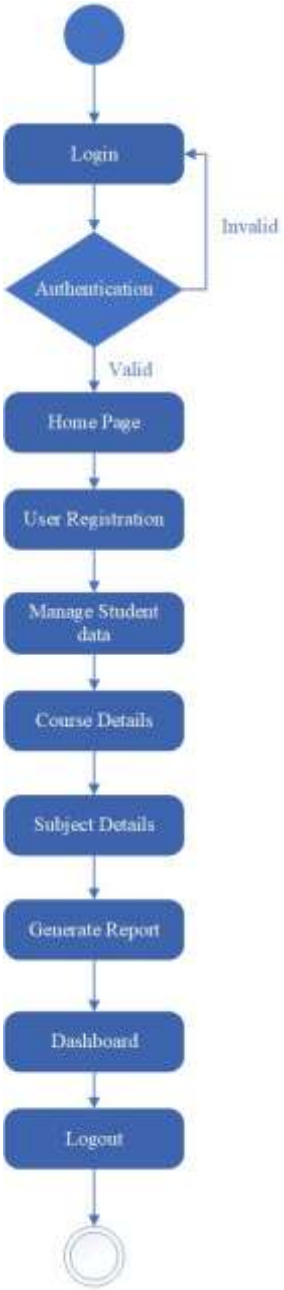


Figure 3-8 : Activity Diagram – Manage Information

3.4.3.2 Take Attendance

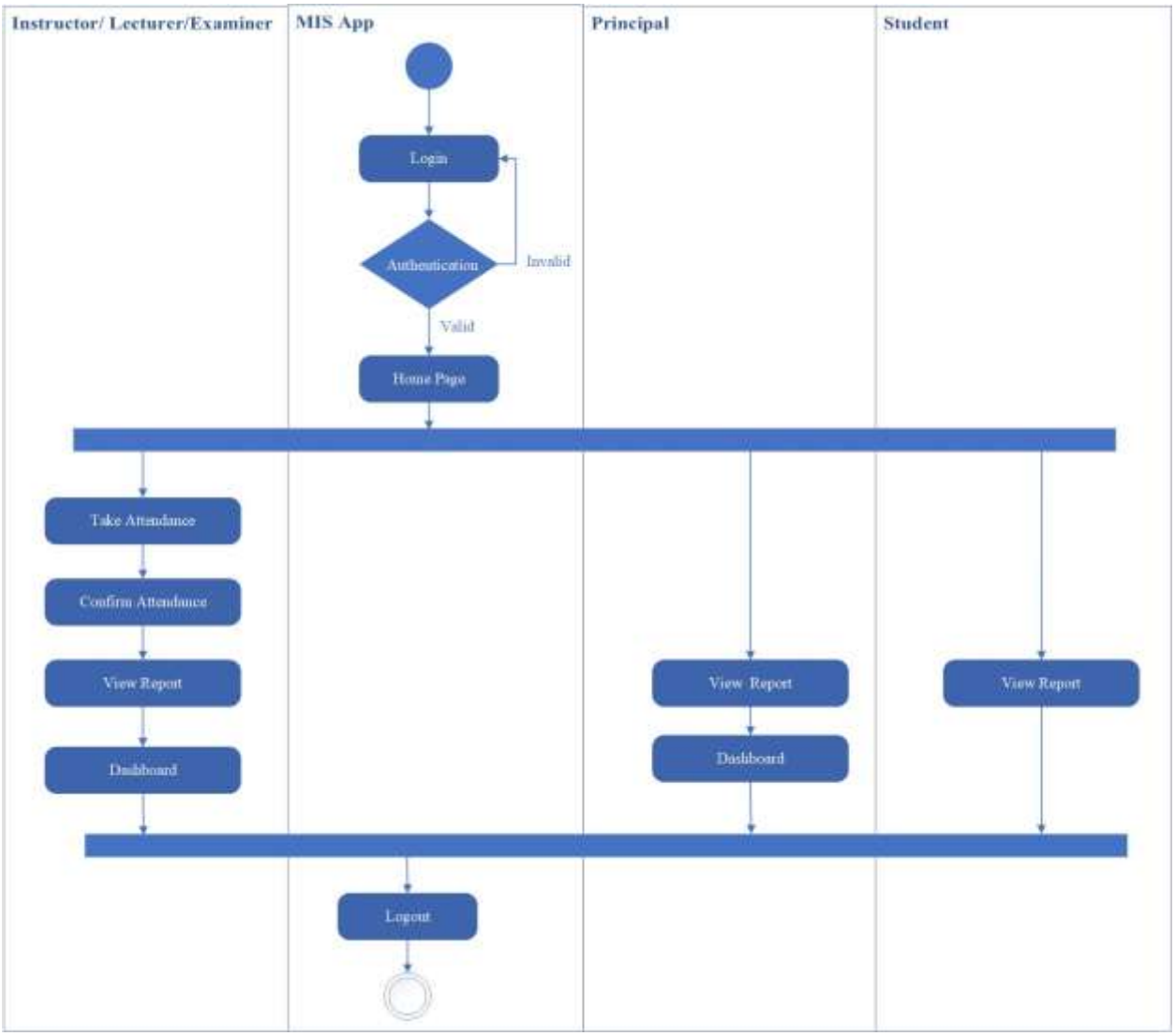


Figure 3-9 : Activity Diagram – Take Attendance

3.4.3.3 Course Fee Payments

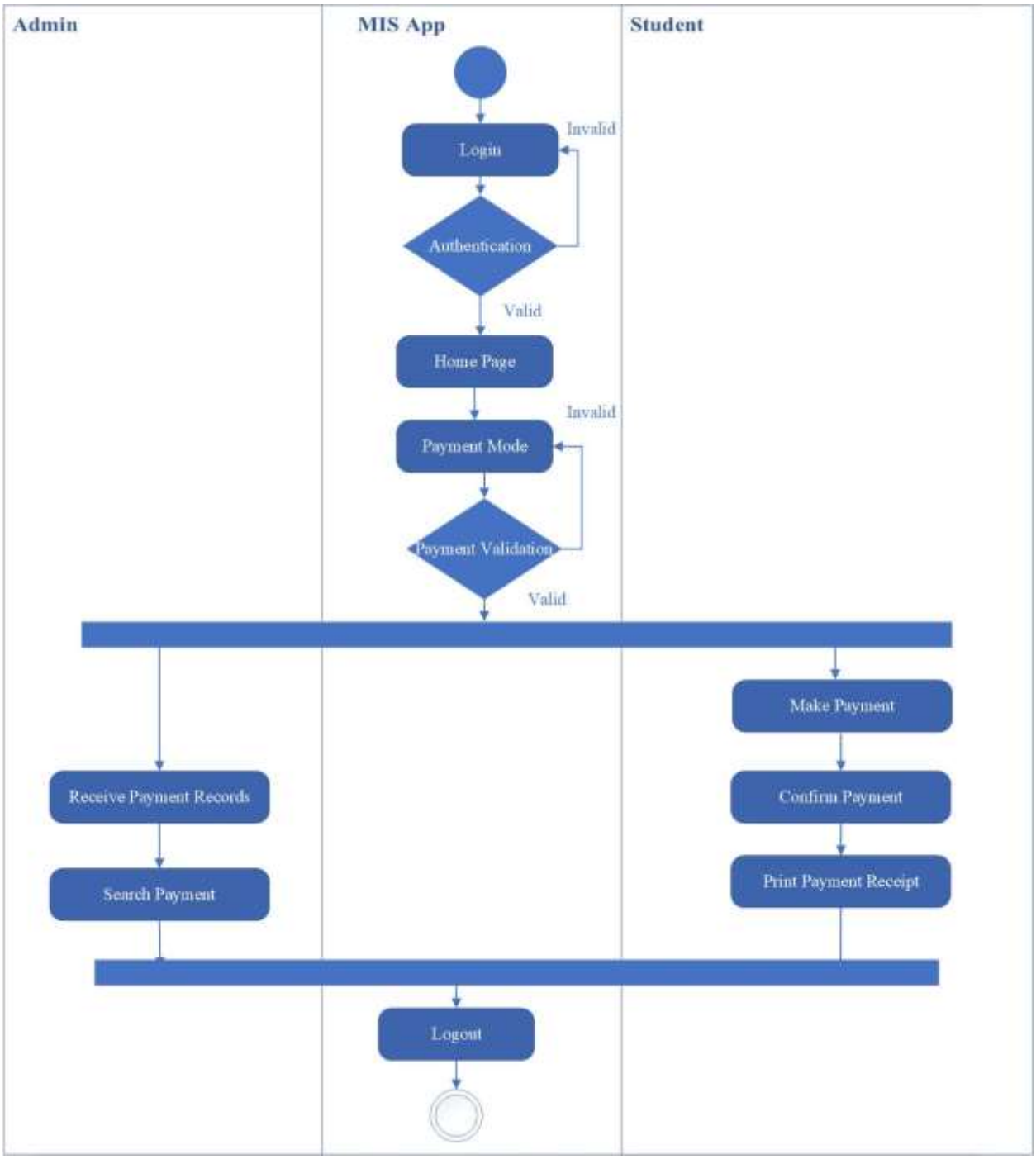


Figure 3-10 : Activity Diagram - Course Fee Payment

3.5 Database Design

Database design is completed through data modeling. The structure of operational systems utilized in the client/server system is specified by database design.

3.5.1 ER Diagram

The database tables were normalized to 3rd normal form to avoid redundancies. The following Entity Relationship diagram in Figure 3-11 the structure of the relational database configuration of the system.

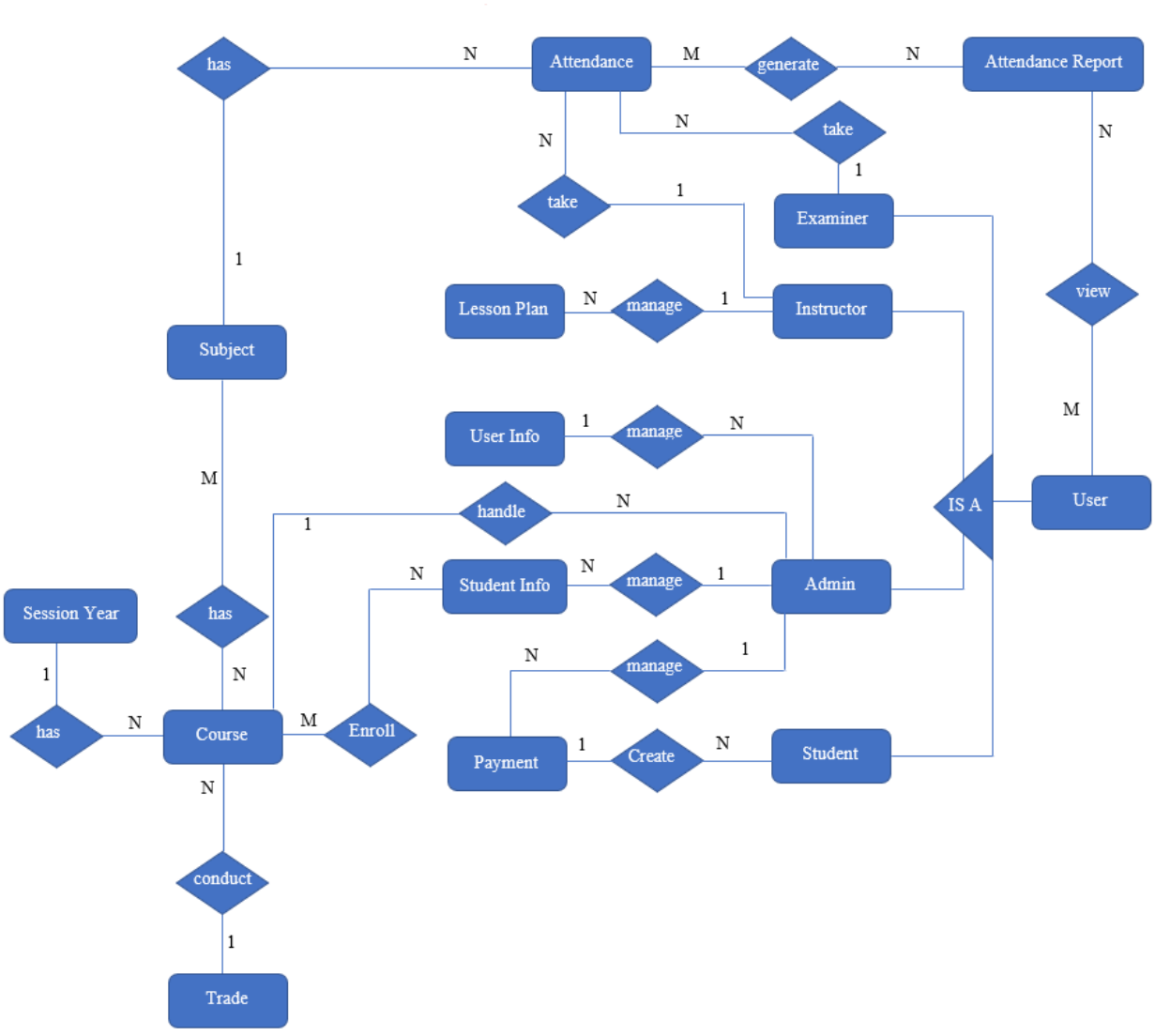


Figure 3-11 : ER diagram of the proposed system

3.5.2 Class Diagram

Class diagrams support carrying out conceptual and domain modeling. A conceptual model represents objects and classes from a particular system user's perspective. The following class diagram in Figure 3-12 depicts the overall class diagram of the system.

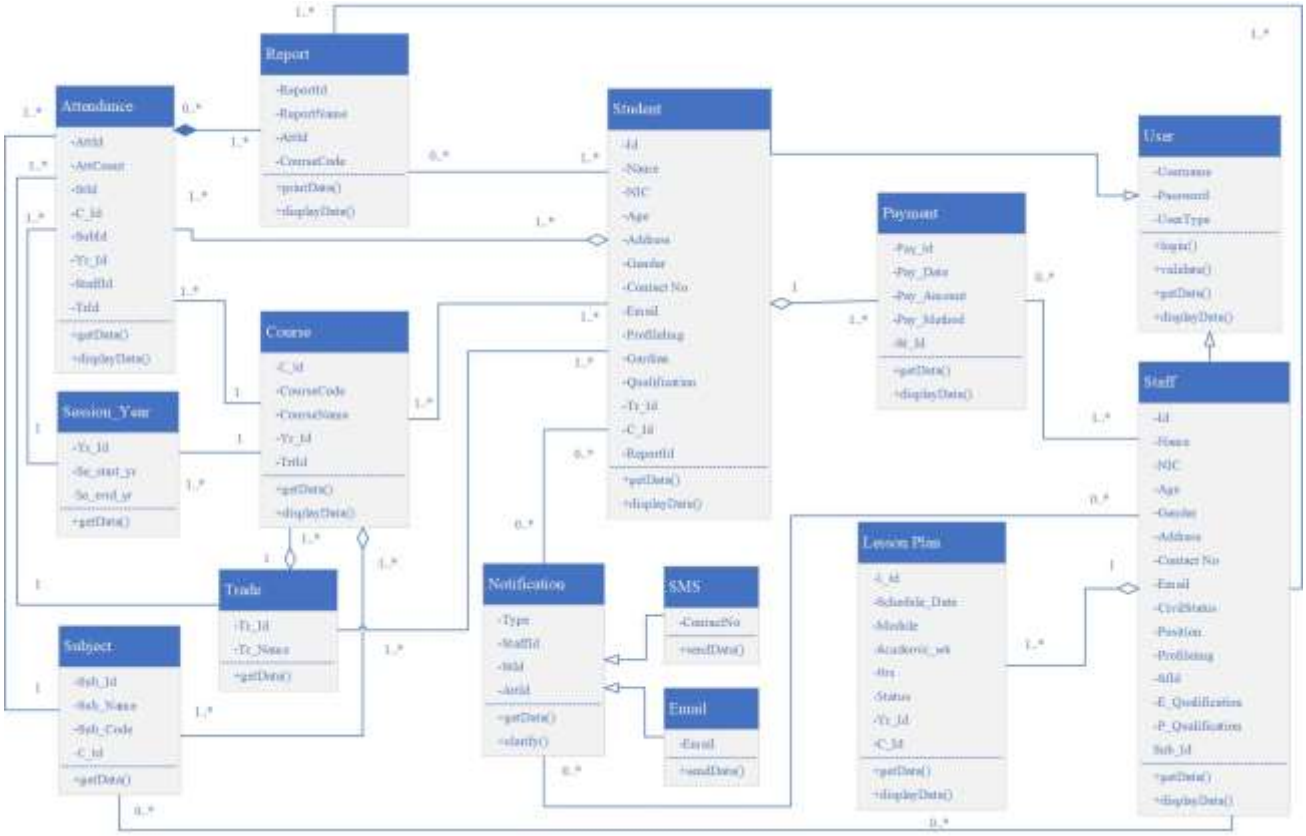


Figure 3-12 : Class diagram of the proposed system

3.5.3 Database Diagram

Database Design: The following database diagrams specify the structure of the database in a detailed manner where, primary keys, foreign keys of tables and the relationship between them are distinguished.

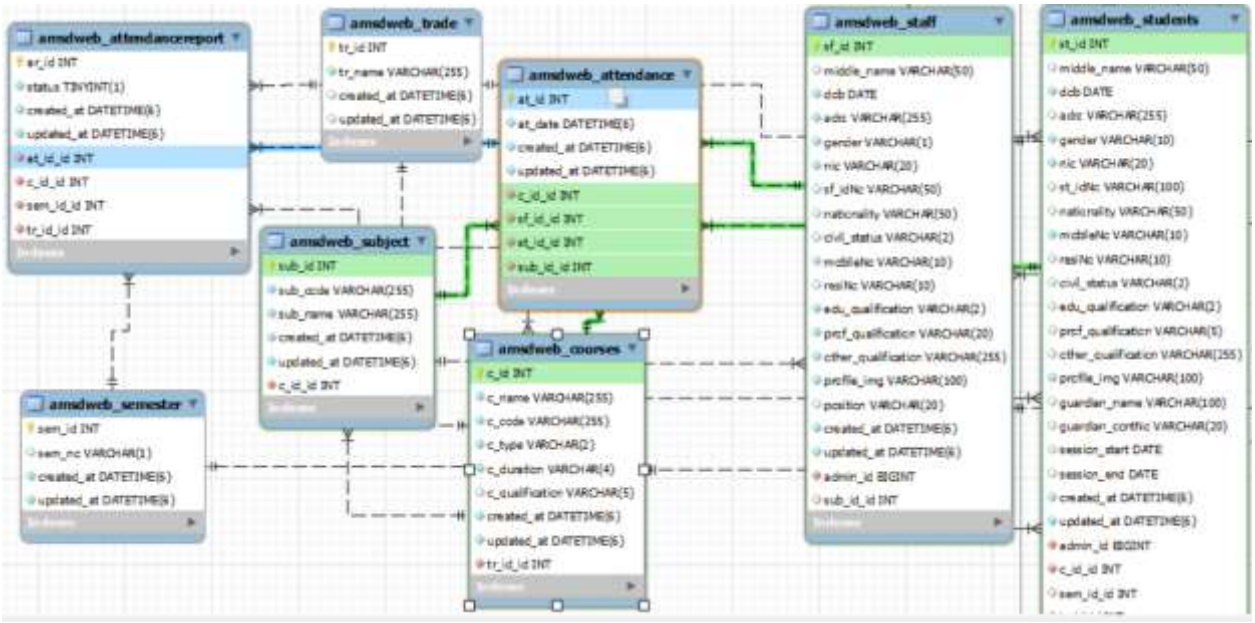


figure 3-13 : DB Diagram for Attendance Module

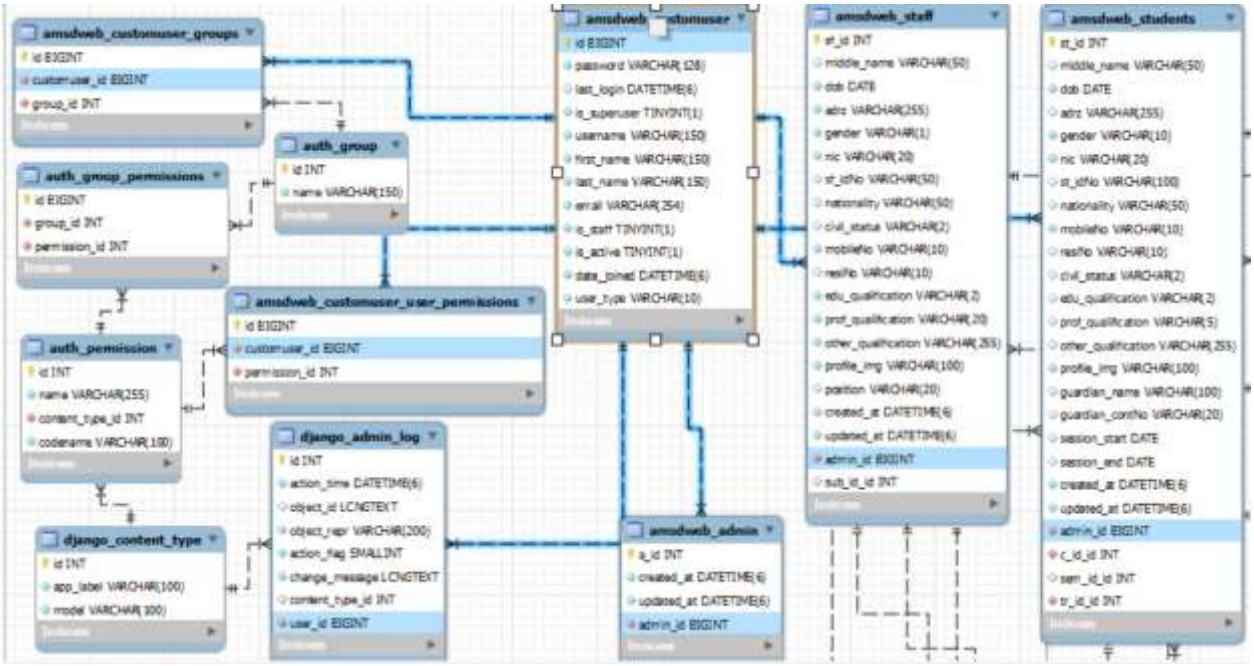


Figure 3-14 : DB Diagram for User Module

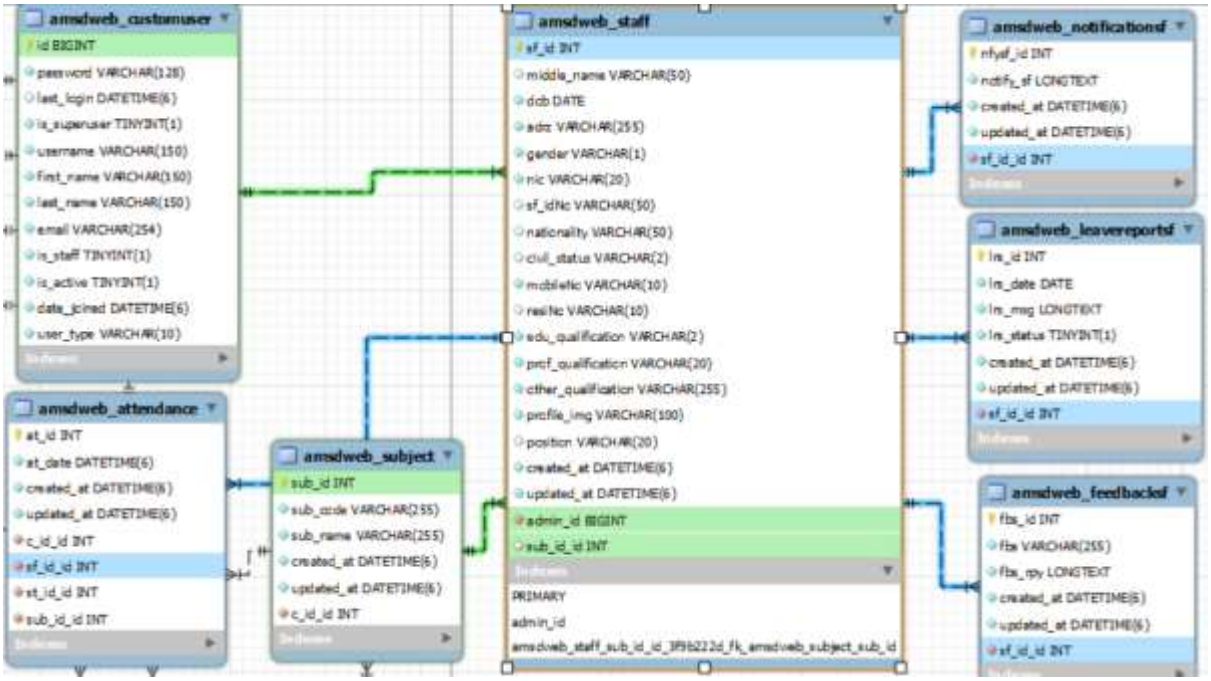


figure 3-15 : DB Diagram for Staff Module

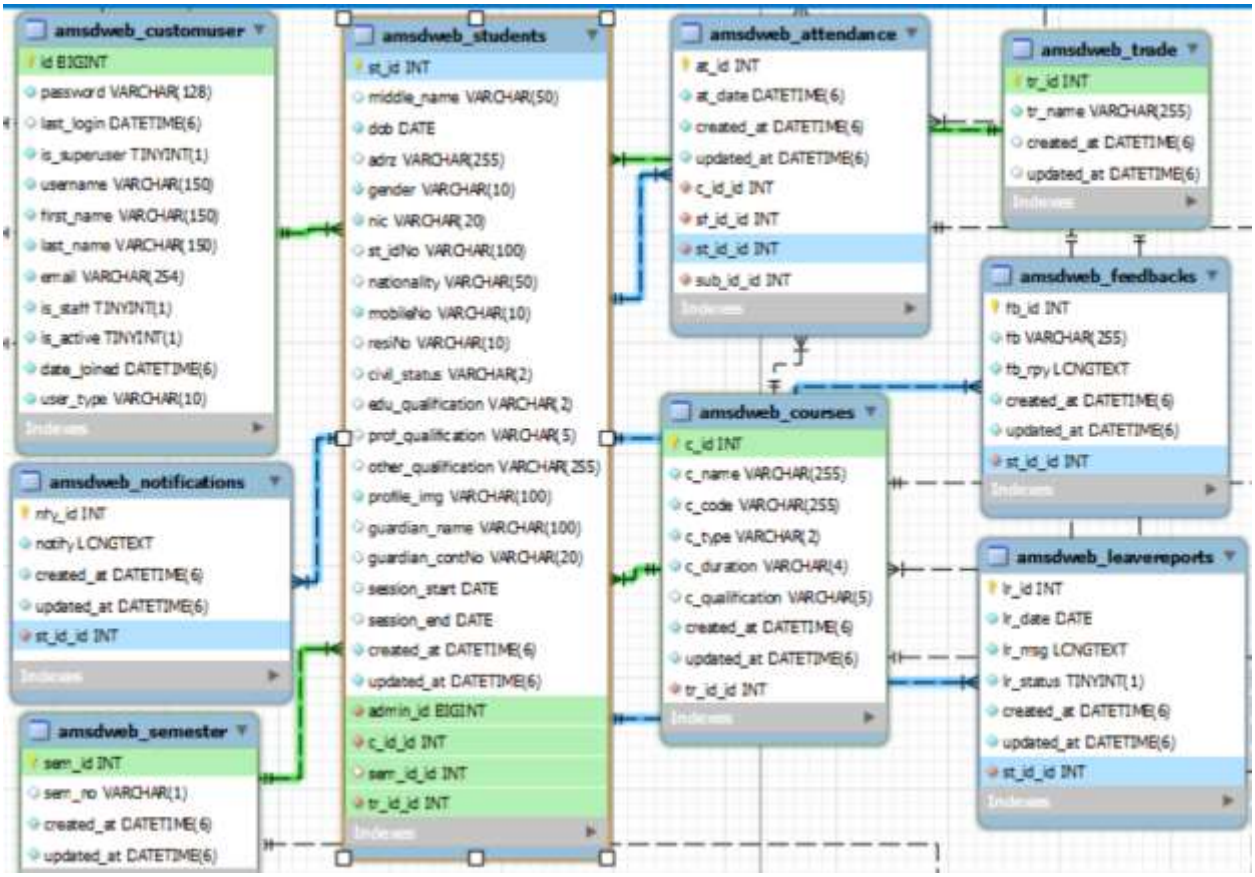


Figure 3-16 : DB Diagram for Student Module

3.6 Summary

The proposed solution's design architecture was the focus of this chapter. UML diagrams are used to represent this high-level architecture.

CHAPTER 4 - IMPLEMENTATION

4.1 Introduction

The process of implementation is used to convert a system's specifications into executable code. During this phase, convert the design into user-friendly views with front-end interfaces and back-end systematic logic to perform functionalities. The coding and development process began with the selection of a comfortable language and appropriate tools. The codes were written and set up in a comprehensible manner. For the purpose of future implementations, special logic has been commented on in the code.

4.2 Related Technologies and Tools

The web application and core API (Application Program Interface) service (backend) of the suggested software tools and technologies are described in the table below.

Application	Technology	Tools
MIS Web Application (Front End and Back End)	Django	PyCharm Community Edition 2021.3.2
	Face Recognition Libraries	
	OpenCV	
	HTML	
	CSS	
	Java Script	
	Json	
	Ajax	
	Bootstrap	
	MySQL	MySQL Workbench 8.0
	Excel	MS Office

Table 4-1 : Application components and related technologies and tools

4.3 Modules of the System

Gampaha Technical College's Management Information System is divided into the following modules, as shown in Figure 4-1. When the system performs the required functionality, the interoperability of these modules is vital.

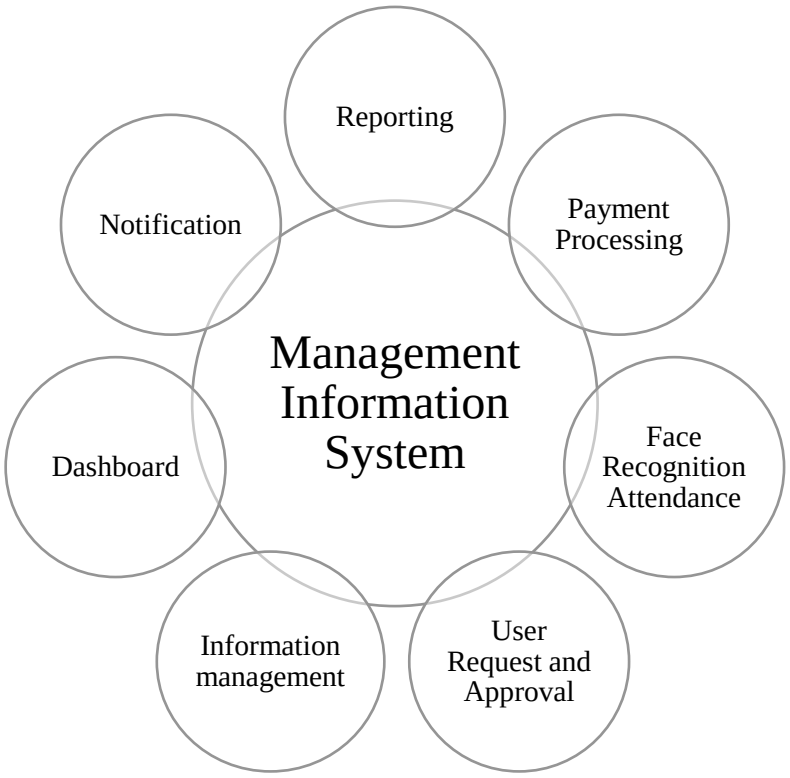


Figure 4-1 : The system module interaction

4.4 Application Development Architecture

As stated in the system architecture section (3.3), the system is built on top of the MVT architecture. The model component represents the data structure and logic of the application. The View component is responsible for handling user requests and returning appropriate responses, and the Template component is responsible for generating the user interface (UI) of the application. Figure 4-2 : Application Development Architectureshows the application development structure of PyCharm 2021.

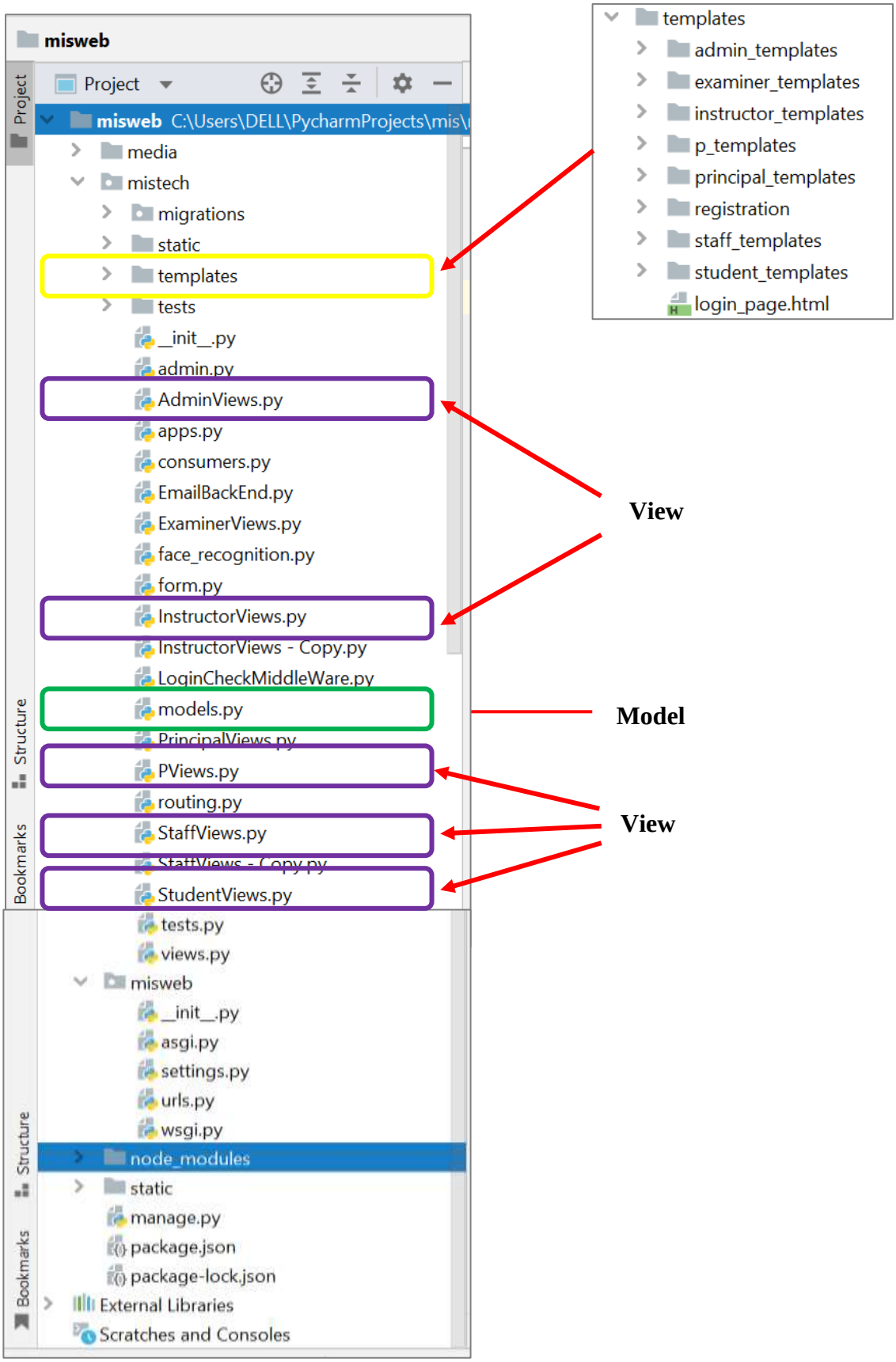


Figure 4-2 : Application Development Architecture

4.5 Middleware

Django middleware is a framework of hooks into Django's request and response processing. It's a lightweight, low-level plugin system that's used to modify incoming requests or outgoing responses globally in a Django application (“Middleware — Django 1.4.22 documentation,” n.d.).

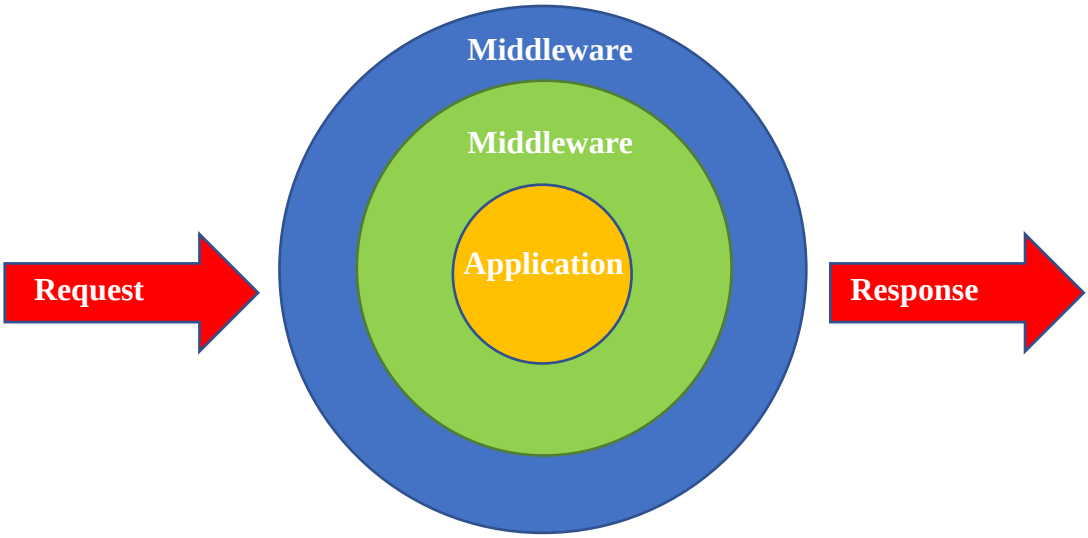


Figure 4-3 : System Middleware

The MIS Tech application's Figure 4-3 : System Middleware intercepts HTTP requests before they reach the view functions. Once intercepted, middleware can handle incoming requests using established rules or logic. Similarly, middleware intercepts the responses generated by the MIS Tech application before they are sent back to the client Figure 4-4 :Login Check Middleware.

LoginCheckMiddleWare.py

```python
from django.http import HttpResponseRedirect
from django.urls import reverse
from django.utils.deprecation import MiddlewareMixin


class LoginCheckMiddleWare(MiddlewareMixin):

    def process_view(self, request, view_func, view_args, view_kwargs):
        modulename = view_func.__module__
        print(modulename)
        user = request.user
        # Exclude password reset URLs from redirection

        if user.is_authenticated:
            if user.user_type == "1":
                if modulename == "mistech.AdminViews":
                    pass
                elif modulename == "mistech.views" or modulename == "django.views.static":
                    pass
                elif modulename == "django.contrib.auth.views" or modulename == "django.contrib.admin.sites":
                    pass
                else:
                    return HttpResponseRedirect(reverse("admin_home"))
            elif user.user_type == "2":
                if modulename == "mistech.StaffViews" or modulename == "django.views.static":
                    pass
                elif modulename == "mistech.views" :
                    pass
                else:
                    return HttpResponseRedirect(reverse("staff_home"))
            elif user.user_type == "4":
                if modulename == "mistech.InstructorViews" or modulename == "django.views.static":
                    pass
                elif modulename == "mistech.views" :
                    pass
```

LoginCheckMiddleWare › process_view() › if user.is_authenticated › if user.user_type == "1" › elif

Figure 4-4 :Login Check Middleware

4.6 Main User Interfaces and Coding

The following section of the document only shows a few main user interfaces to illustrate the system interface structure. Please see Appendix C - User Documentation for the remaining interface designs.

4.6.1 Login Page

The login page is the initial page on this system. It includes two text areas where you may enter your email address and password. It also has a command button called Sign In that initiates the operation. This interface (Figure 4-5 :Login Page) allows the user to sign in and access the MIS system using the proper user email address and password.

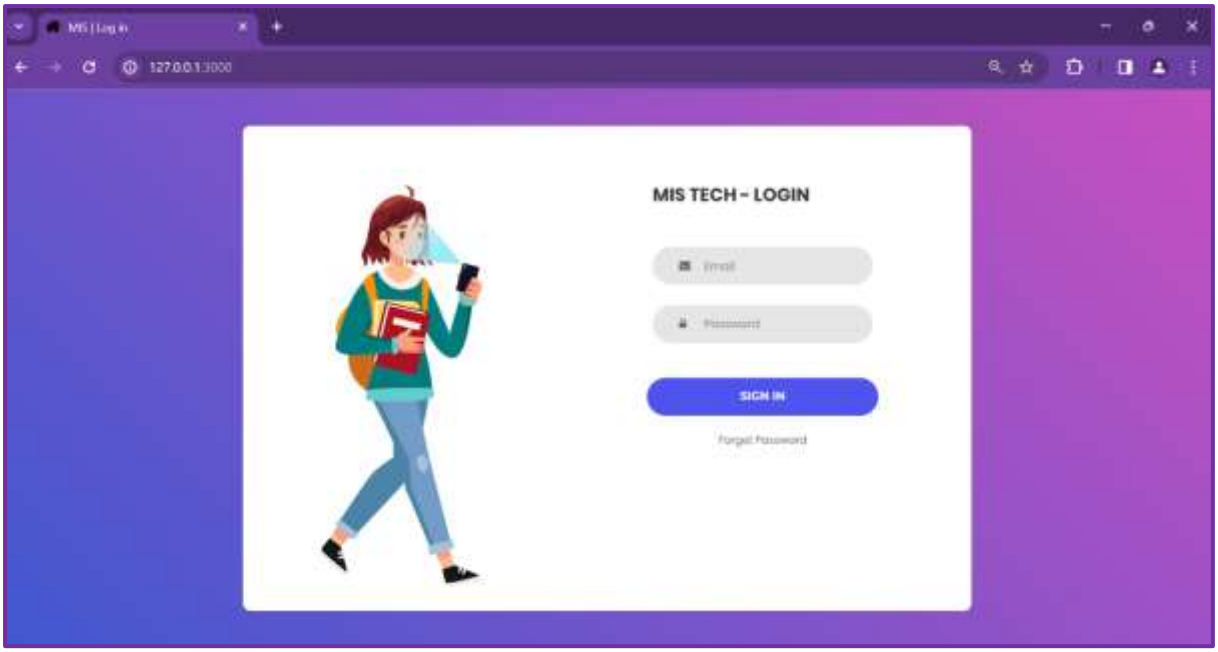


Figure 4-5 :Login Page

If a user forgets their password, they can reset it by clicking the "forget password" link and entering their email address, as seen in Figure 4-6 : Forgot Password Page.

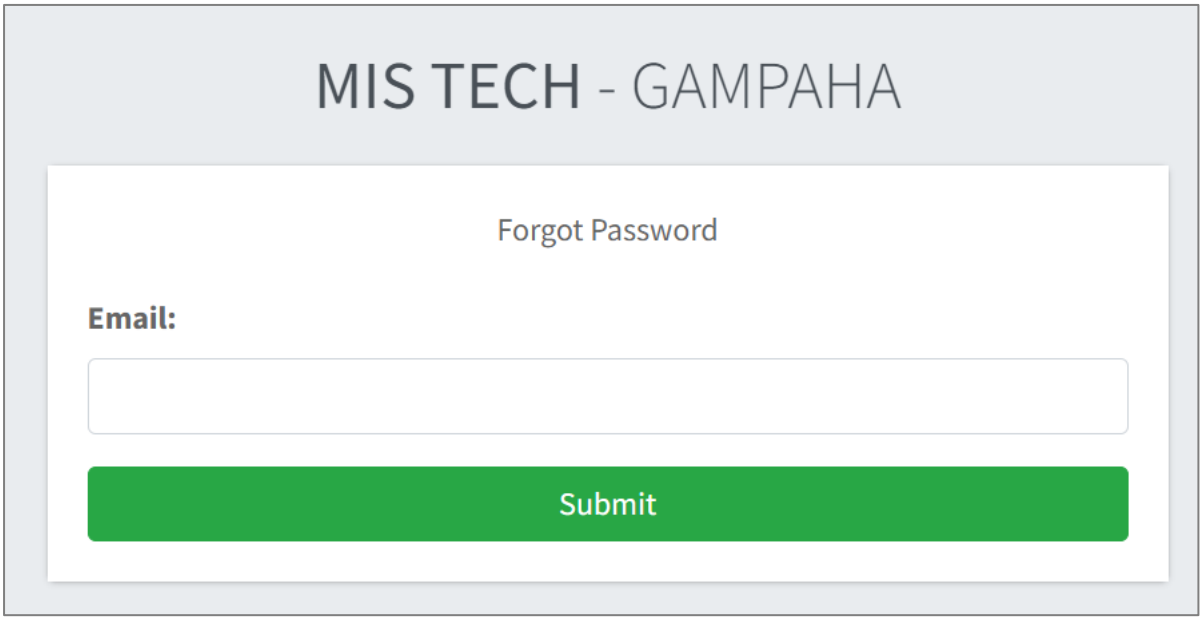


Figure 4-6 : Forgot Password Page

Change the password and add a new password, as illustrated in Figure 4-7 : Change password.

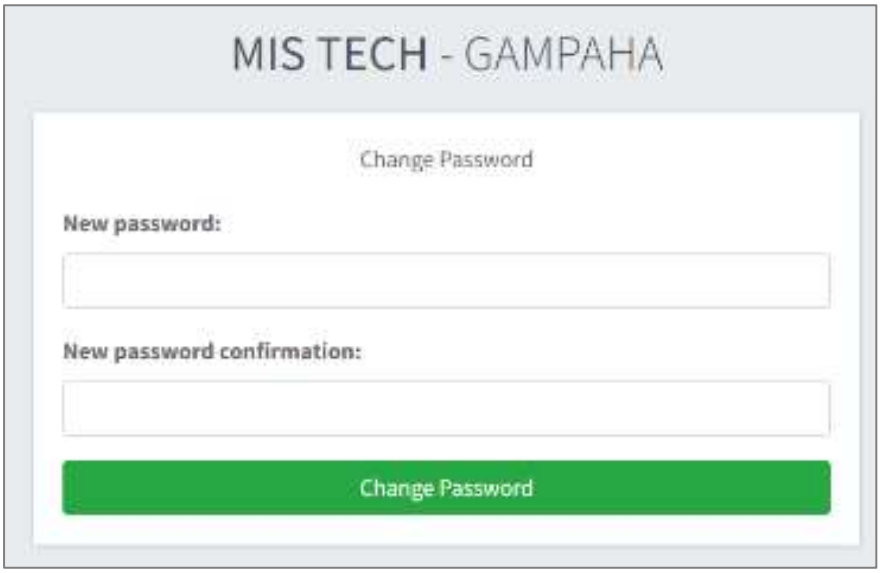


Figure 4-7 : Change password

If both credentials are valid user can view the dashboard shown in the Figure 4-8 : User Dashboard

Figure 4-8 : User Dashboard

4.6.2 Tracking Student Attendance

Instructor or Examiner could be able to take attendance as depict in the Figure 4-9 : Face recognition attendance.

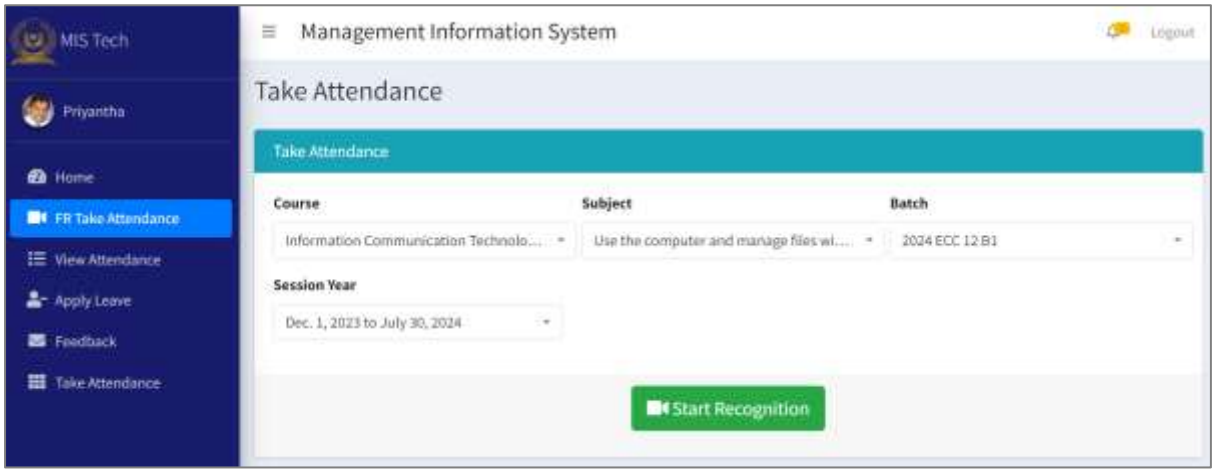


Figure 4-9 : Face recognition attendance

Instructor or Examiner could be able to view students attendance details through the system in the Figure 4-10 : View student attendance.

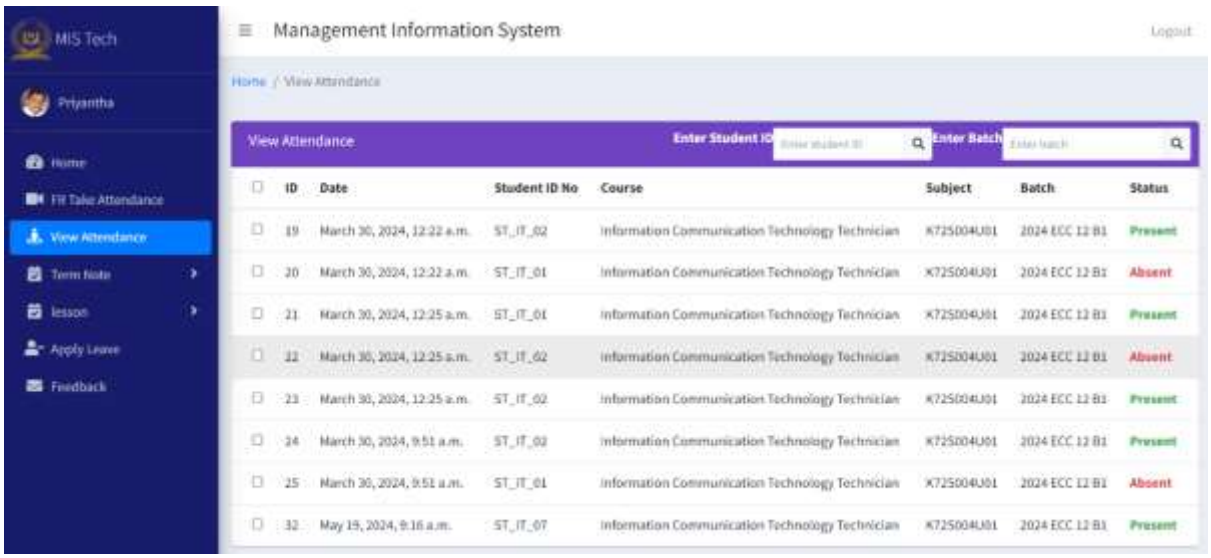


Figure 4-10 : View student attendance

4.6.3 Leave Application and Management

Both instructors and students will be able to apply for leave through the system, as depicted in Figure 4-11 : Leave Application Interface.

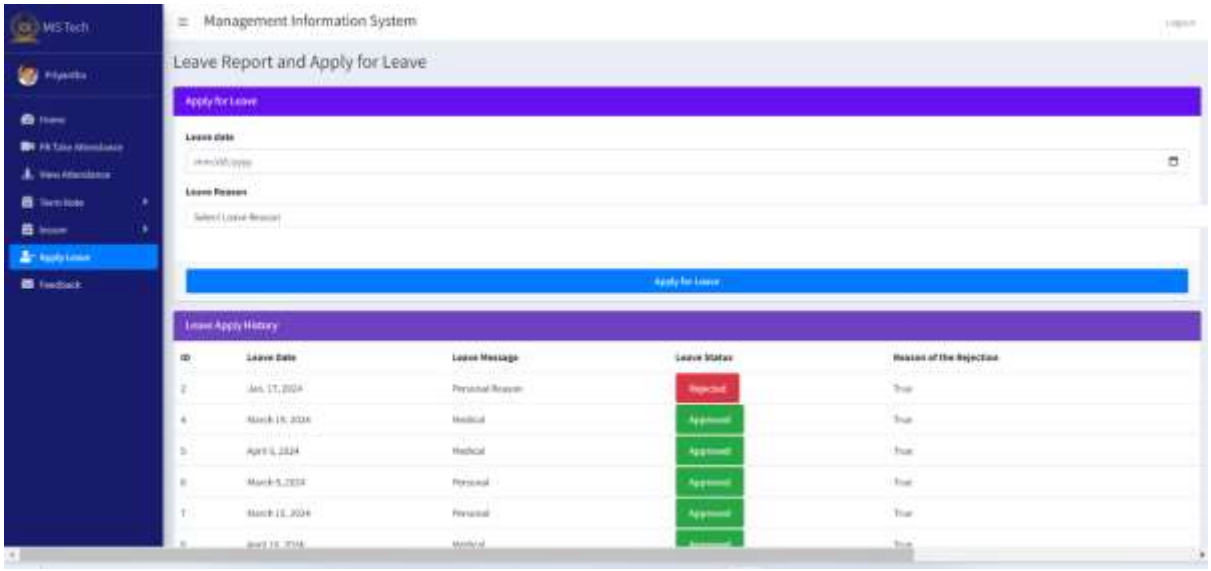


Figure 4-11 : Leave Application Interface

The leave requests can be reviewed and either accepted or rejected by the principal, as shown in Figure 4-12 : Principal's Leave Approval Form.

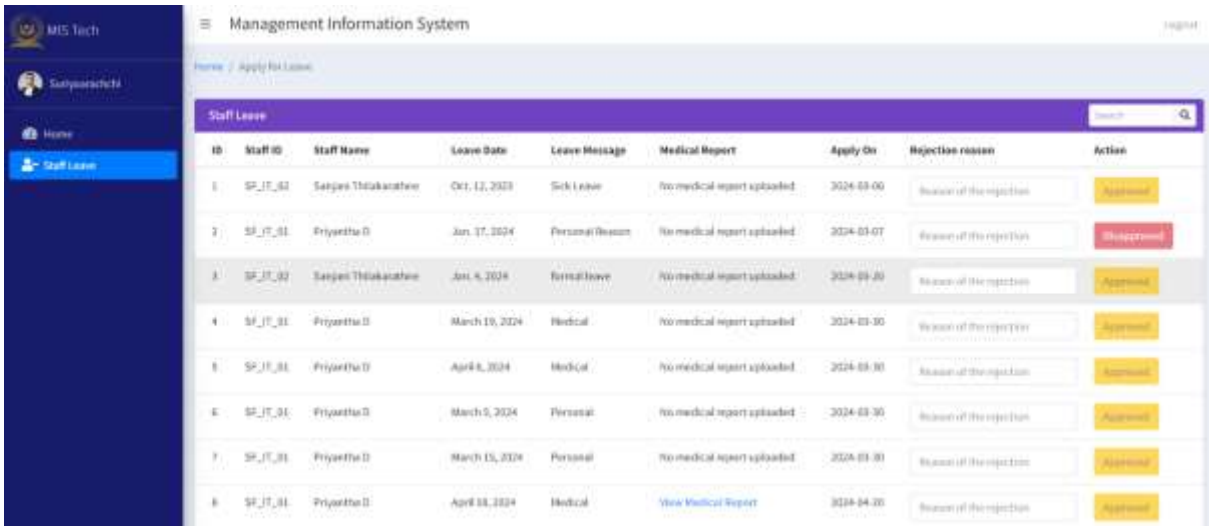


Figure 4-12 : Principal's Leave Approval Form

Additionally, through the principal's dashboard, the total number of students, staff, courses, and subjects can be easily monitored as in Figure 4-13 : Principal's dashboard .

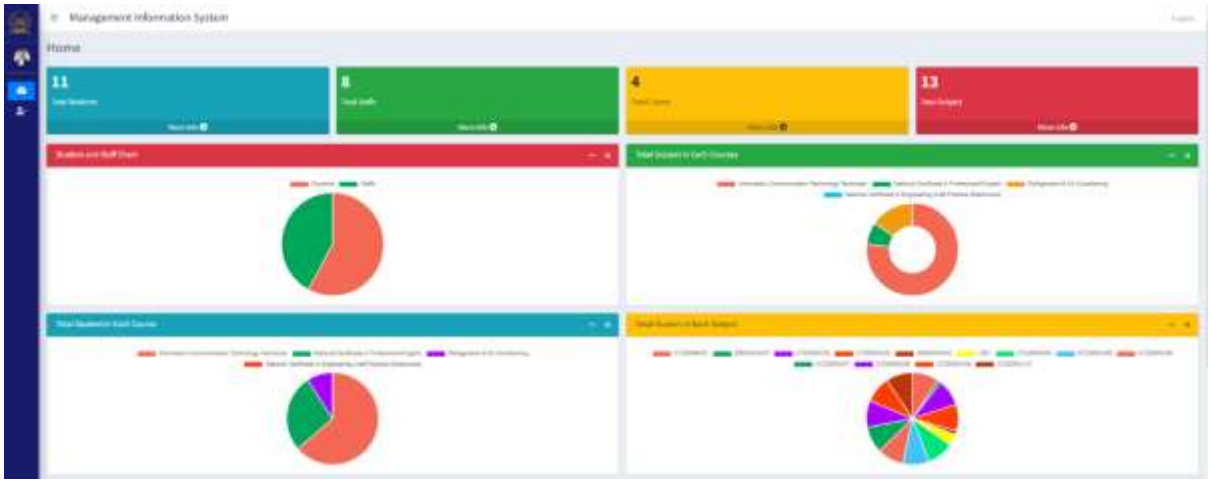


Figure 4-13 : Principal's dashboard

A bar chart, depicted in Figure 4-14 : Staff and Student Attendance vs. Leave, allows the principal to analyze and compare staff leaves versus attendance, and student leaves versus attendance.

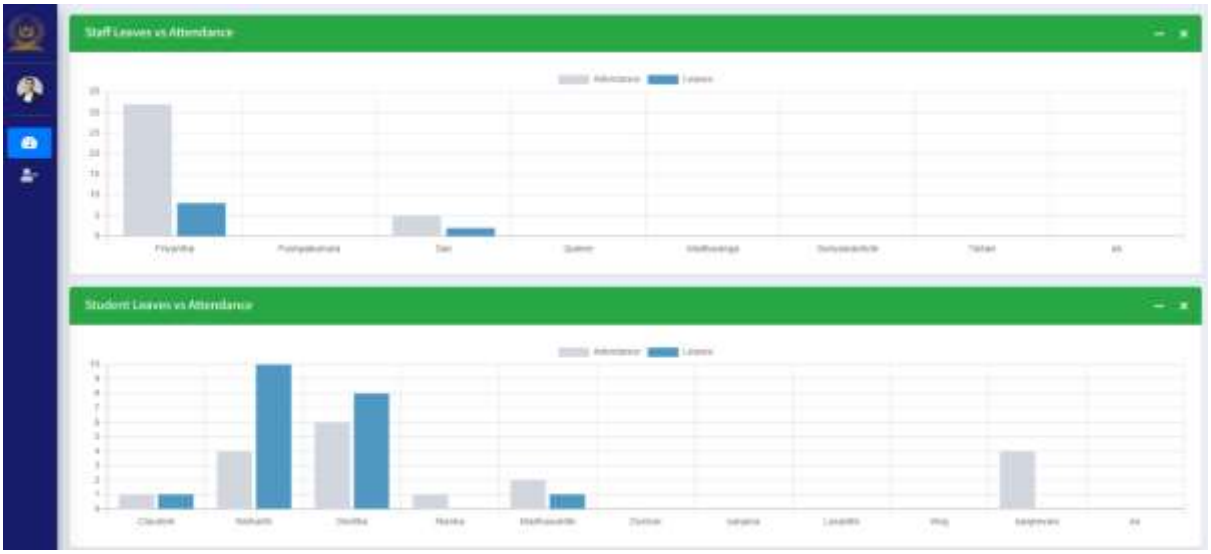


Figure 4-14 : Staff and Student Attendance vs. Leave

4.6.4 Feedback Management

The system allows both instructors and students to send feedback messages, as shown in Figure 4-15 : Feedback Submission.

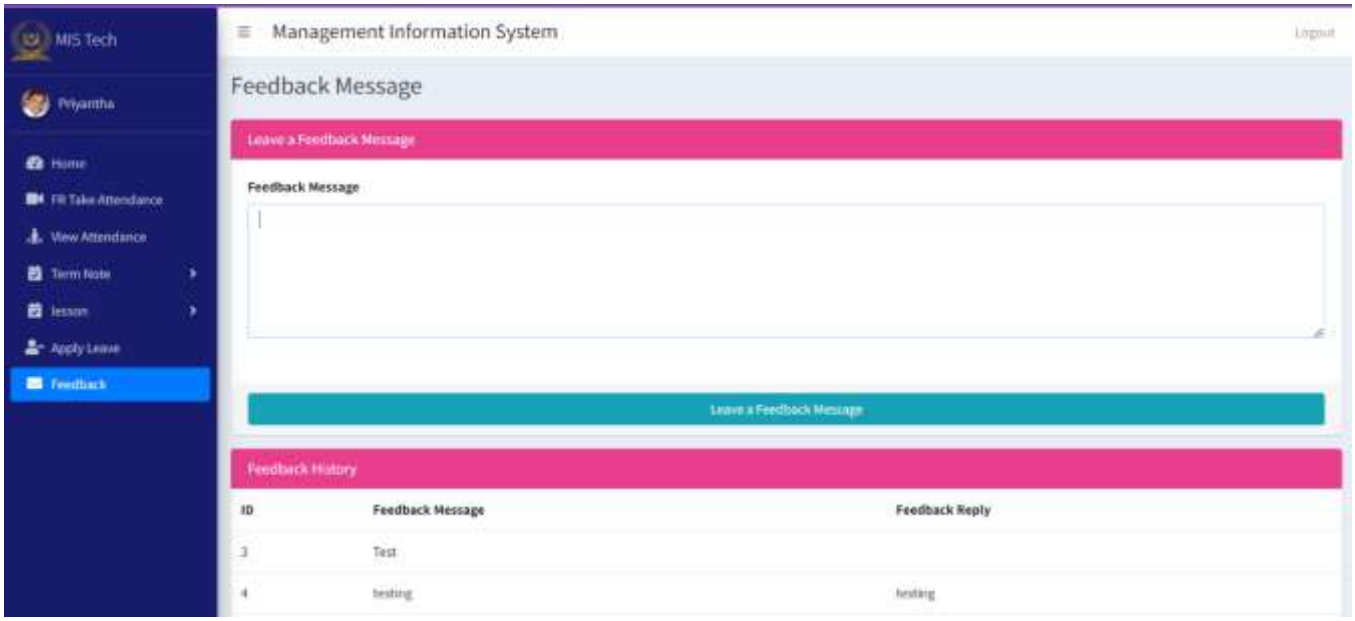


Figure 4-15 : Feedback Submission

4.6.5 Create and Manage Term Notes

Instructors will be able to create and manage term notes, as shown in Figure 4-16 : Term Note Creation, and generate detailed reports based on student performance in Figure 4-17 : View Term Notes.

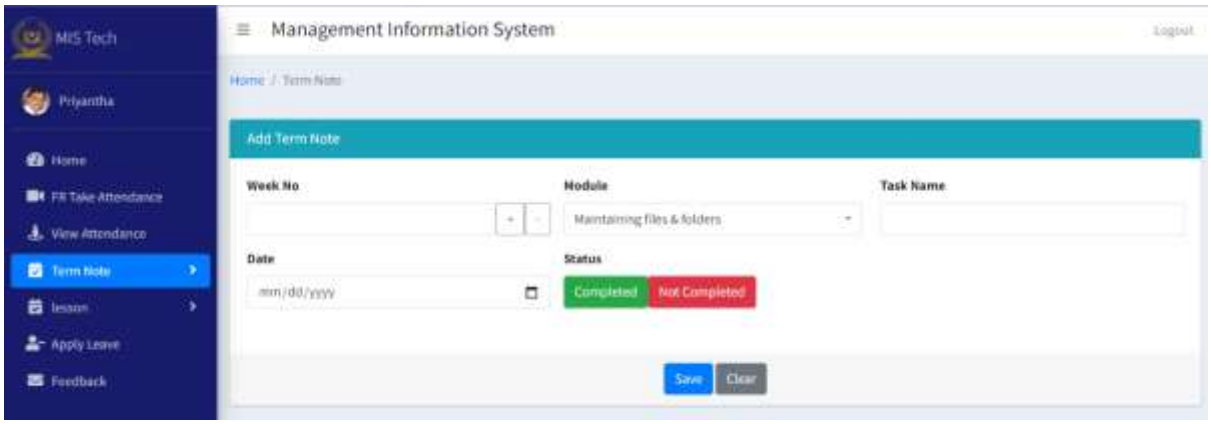


Figure 4-16 : Term Note Creation

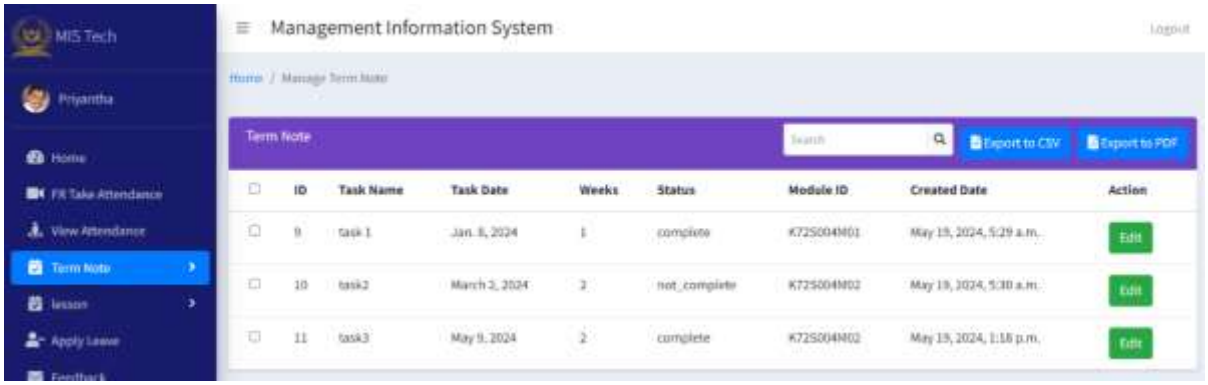


Figure 4-17 : View Term Notes

4.6.6 Lesson Plan Management

Instructors can use the system to create lesson plans, as shown in Figure 4-18 : Lesson Plan Addition, and view their structured plans in Figure 4-19 : View Lesson Plan.

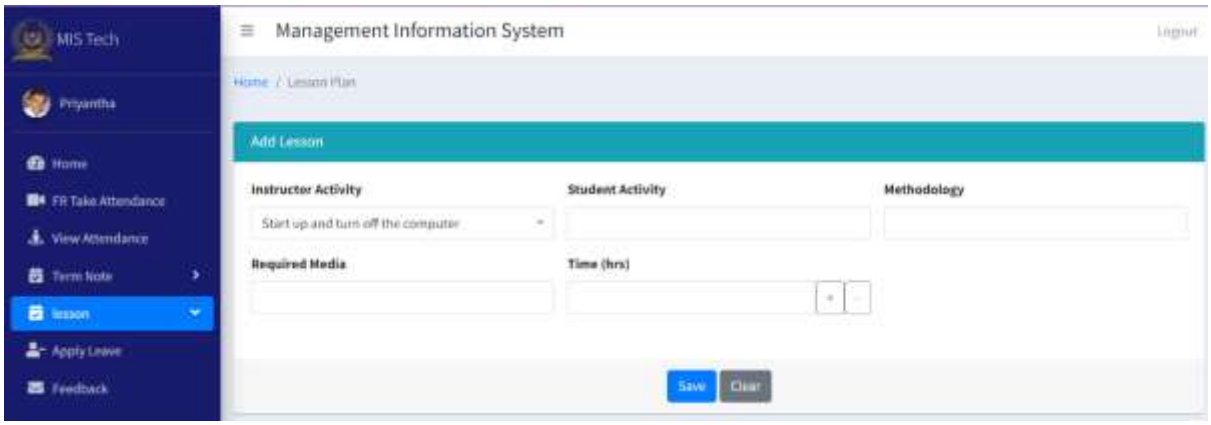


Figure 4-18 : Lesson Plan Addition

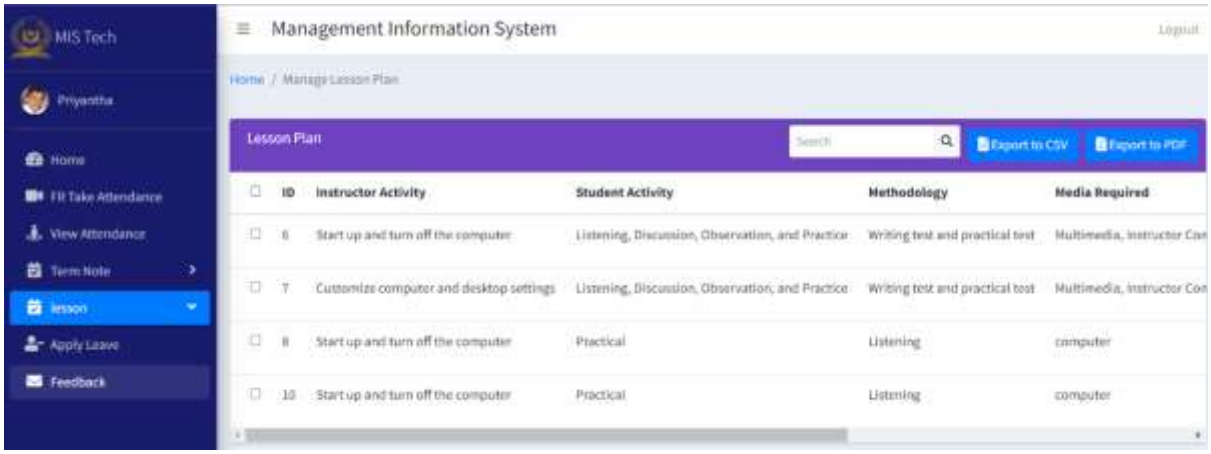


Figure 4-19 : View Lesson Plan

The instructor dashboard allows them to monitor key metrics such as the number of students assigned to them, total attendance recorded, total leave taken, subjects taught, and subjects attended, as shown in Figure 4-20 : Instructor Dashboard Overview.

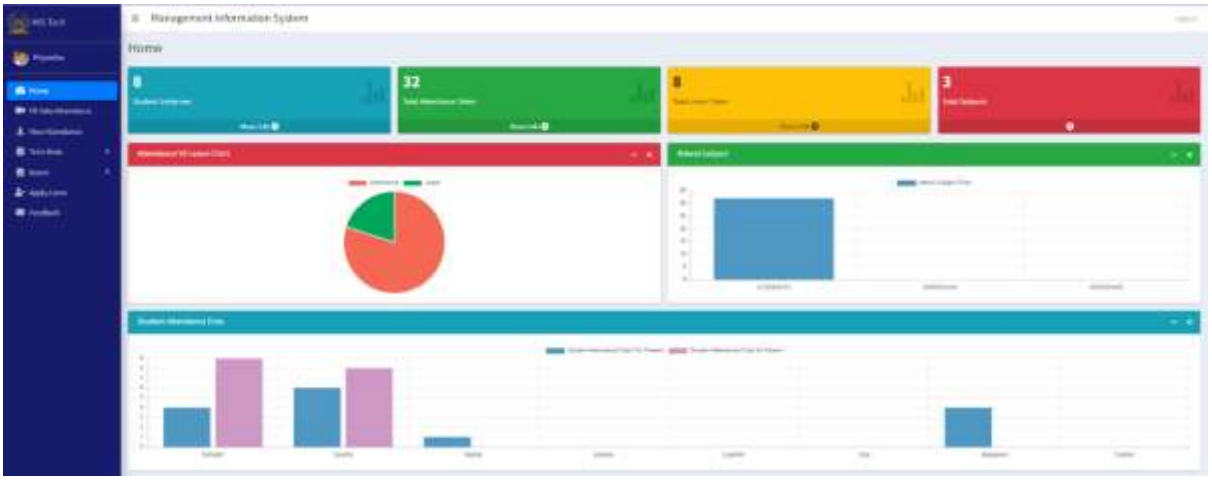


Figure 4-20 : Instructor Dashboard Overview

4.6.7 Course Payment and Transaction Management

Students can securely make course payments through the system using their card details, as shown in Figure 4-21 : Course Payment Interface.

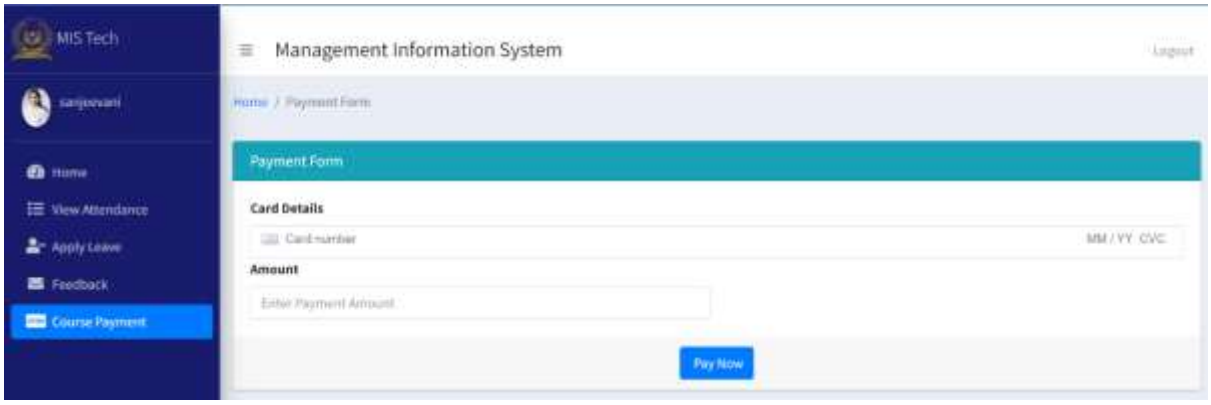


Figure 4-21 : Course Payment Interface

Upon successful payment, a confirmation message will be displayed on the system, as depicted in Figure B: Payment Success Notification, and a payment confirmation email will be automatically sent to the student's registered email address, as shown in Figure 4-23 : Payment Confirmation Email.

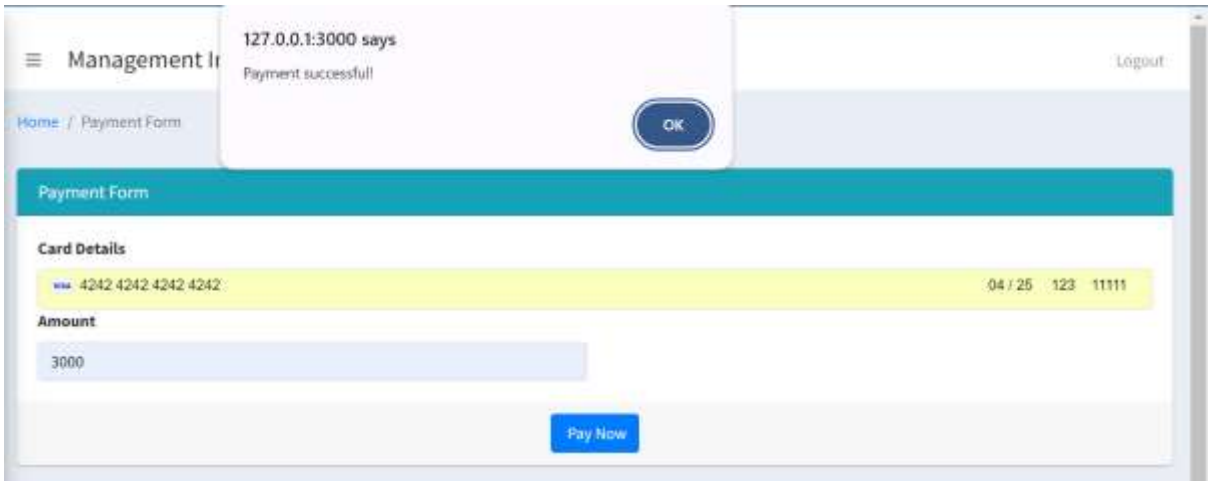


Figure 4-22 : Payment Success Notification

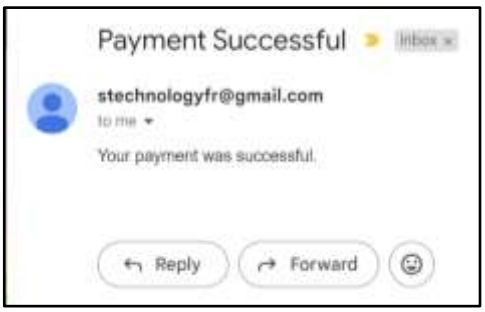


Figure 4-23 : Payment Confirmation Email

All payment transactions can be tracked and viewed on the Stripe transaction page, as illustrated in Figure 4-24 : Stripe Transaction Overview.

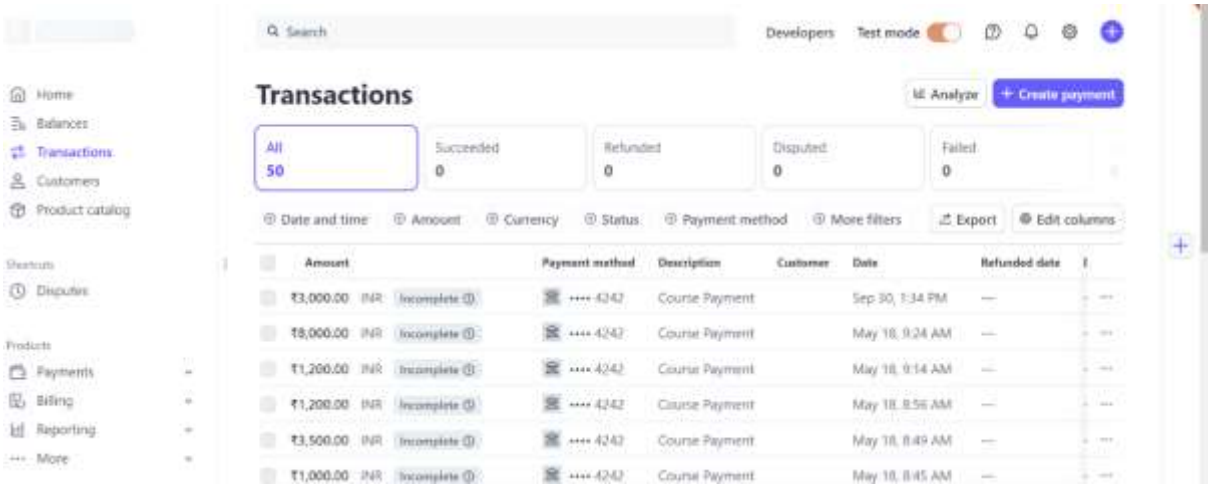


Figure 4-24 : Stripe Transaction Overview

Administrators can access detailed course payment records, as seen in Figure 4-25 : Admin Course Payment Details, and the admin dashboard provides a clear visual comparison of students who have paid versus those who have not, as shown in Figure 4-26 : Paid vs. Unpaid Course Fees Overview .

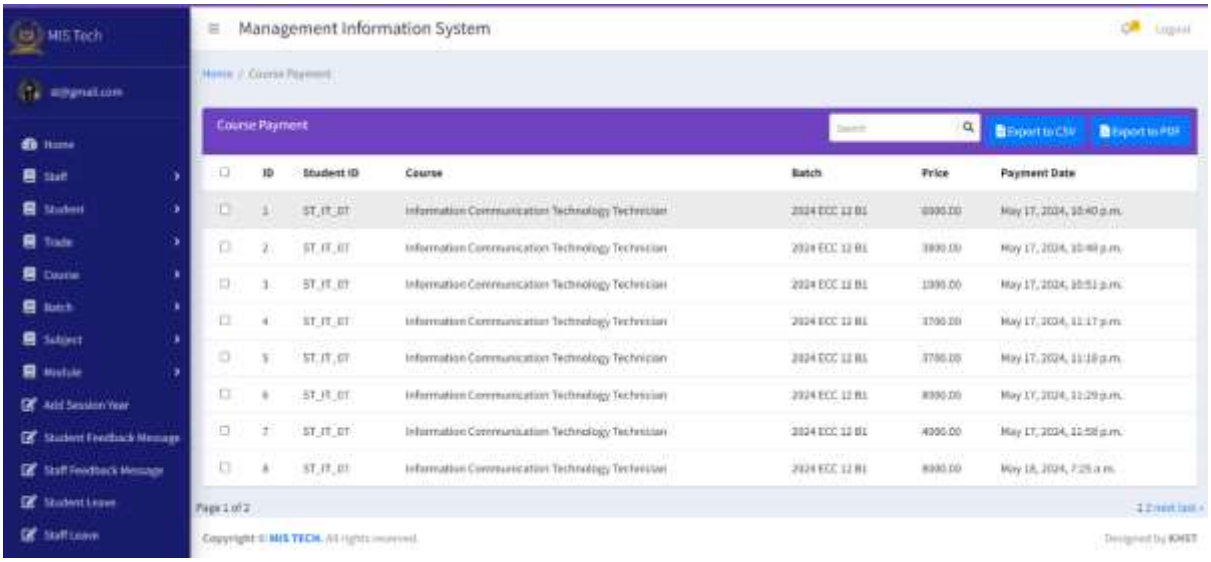


Figure 4-25 : Admin Course Payment Details

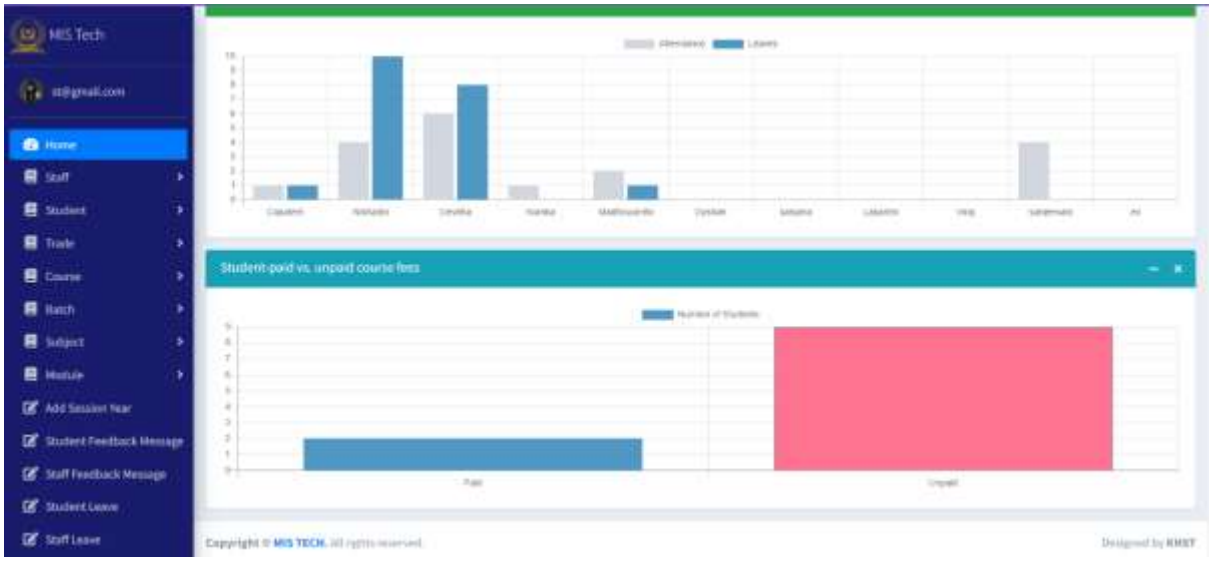


Figure 4-26 : Paid vs. Unpaid Course Fees Overview

4.6.8 Student Overview

Students can conveniently view their attendance records through the system, as depicted in Figure 4-27 : Student Attendance Overview.

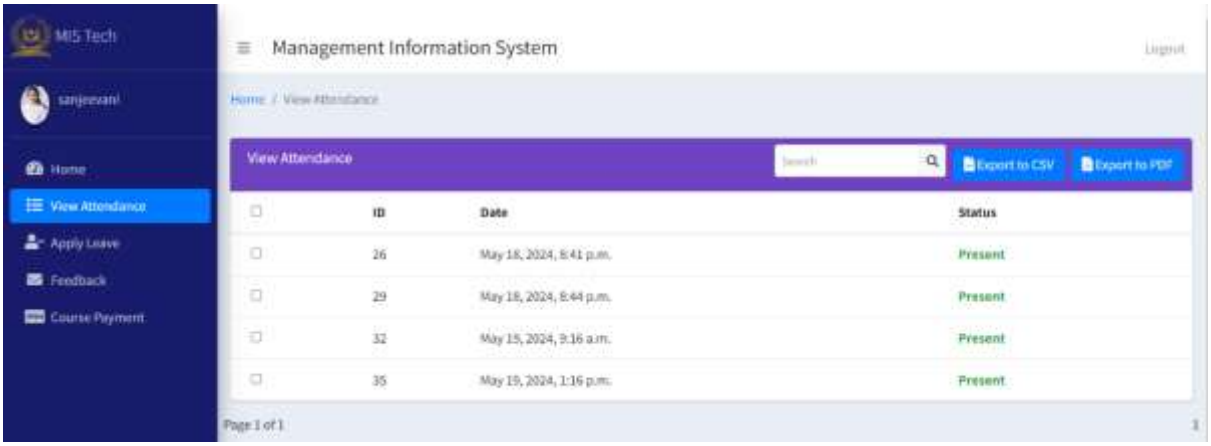


Figure 4-27 : Student Attendance Overview

From their personalized dashboard, students can see a breakdown of their total attendance, including separate counts for absences and presences. The dashboard also features an attendance chart for visual representation and displays their overall attendance percentage, providing a clear and comprehensive overview of their attendance status shown in Figure 4-28 : Student Dashboard.

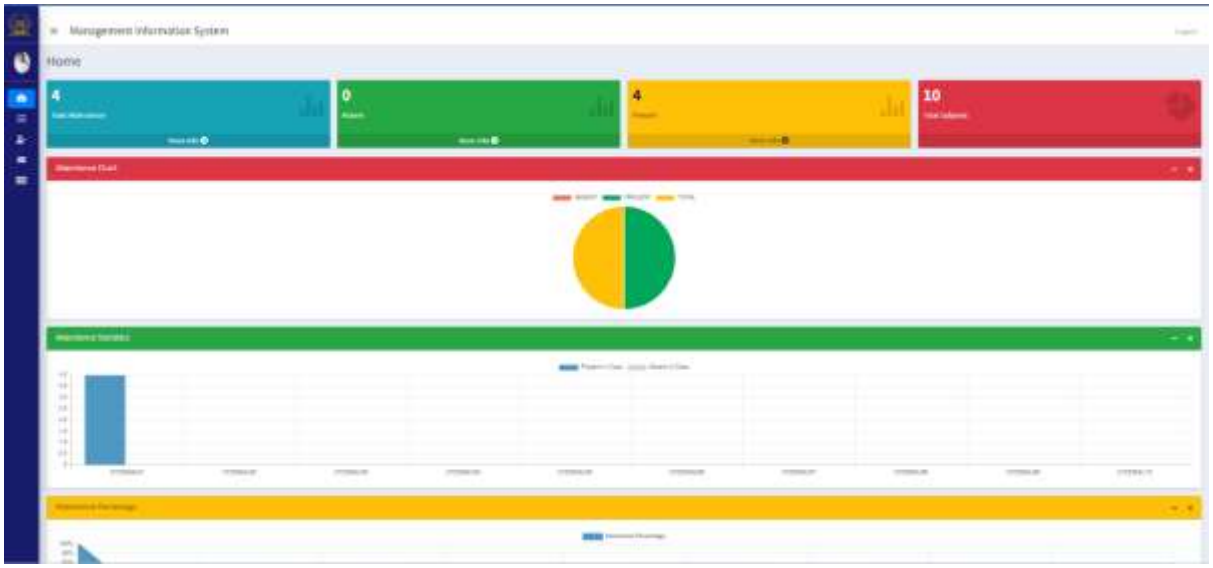


Figure 4-28 : Student Dashboard

Some of the core code structures of the admin portal are shown here. Code structure for the web portal. (Figure 4-29 : Models.py – part I Figure 4-35 : Form.py – Code Structure):

```
from django.contrib.auth.models import AbstractUser
from django.db import models
from django.db.models.signals import post_save
from django.dispatch import receiver

class SessionYr(models.Model):
    yr_id = models.AutoField(primary_key=True)
    session_start = models.DateField()
    session_end = models.DateField()
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)
    object = models.Manager()

class CustomUser(AbstractUser):
    user_ty = ((1, "Admin"), (2, "staff"), (3, "Student"), (4, "Instructor"), (5, "Examiner"), (6, "Principal"))
    user_type = models.CharField(default=1, choices=user_ty, max_length=10)
    profile_img = models.ImageField(upload_to='user_img/', null=True)

class Admin(models.Model):
    id = models.AutoField(primary_key=True)
    admin = models.OneToOneField(CustomUser, on_delete=models.CASCADE)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)
    objects = models.Manager()

class Trade(models.Model):
    tr_id = models.AutoField(primary_key=True)
    tr_name = models.CharField(max_length=255, unique=True)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)
    objects = models.Manager()
```

Figure 4-29 : Models.py – part I

```python
class Batch(models.Model):
    b_id = models.AutoField(primary_key=True)
    b_code = models.CharField(max_length=50, unique=True)
    sem_no = models.CharField(max_length=1)
    c_id = models.ForeignKey(Course, on_delete=models.CASCADE, default=True)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)
    objects = models.Manager()

class Staff(models.Model):
    sf_id = models.AutoField(primary_key=True)
    admin = models.OneToOneField(CustomUser, on_delete=models.CASCADE)
    middle_name = models.CharField(max_length=50, null=True)
    dob = models.DateField(null=False)
    adrz = models.CharField(max_length=255, null=True)
    gender = models.CharField(max_length=10, null=False)
    nic = models.CharField(max_length=20, null=False)
    sf_idNo = models.CharField(max_length=50, unique=True, null=True)
    nationality = models.CharField(max_length=50, null=True)
    civil_status = models.CharField(max_length=10, null=True)
    mobileNo = models.CharField(max_length=11, null=False)
    resiNo = models.CharField(max_length=11, null=True)
    edu_qualification = models.CharField(max_length=10, null=True)
    prof_qualification = models.CharField(max_length=255, null=True)
    other_qualification = models.CharField(max_length=255, null=True)
    position = models.CharField(max_length=20, null=False)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)
    objects = models.Manager()
```

Figure 4-30 : Models.py – part II

```python
class Course(models.Model):
    c_id = models.AutoField(primary_key=True)
    c_name = models.CharField(max_length=255, unique=True)
    c_code = models.CharField(max_length=255, unique=True)
    c_type = models.CharField(max_length=10)
    c_duration = models.CharField(max_length=4)
    c_qualification = models.CharField(max_length=5, null=True)
    tr_id = models.ForeignKey(Trade, on_delete=models.DO_NOTHING)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)
    objects = models.Manager()

class Subject(models.Model):
    sub_id = models.AutoField(primary_key=True)
    sub_code = models.CharField(max_length=255, unique=True)
    sub_name = models.CharField(max_length=255, unique=True)
    c_id = models.ForeignKey(Course, on_delete=models.CASCADE)
    sf_id = models.ManyToManyField('Staff')
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)
    objects = models.Manager()

class Module(models.Model):
    mod_id = models.AutoField(primary_key=True)
    mod_code = models.CharField(max_length=255, unique=True)
    mod_name = models.CharField(max_length=255)
    duration_hours = models.PositiveIntegerField()
    academic_weeks = models.PositiveIntegerField()
    c_id = models.ForeignKey(Course, on_delete=models.CASCADE, default=True)
    sub_id = models.ForeignKey(Subject, on_delete=models.CASCADE)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)
    objects = models.Manager()
```

Figure 4-31 :Models.py – part III

```
class AttendanceE(models.Model):
    ae_id = models.AutoField(primary_key=True)
    ae_date = models.DateTimeField(auto_now_add=True)
    status = models.CharField(max_length=10, null=True)
    ex_name = models.CharField(max_length=200, null=False)
    st_id = models.ForeignKey(Student, on_delete=models.CASCADE)
    yr_id = models.ForeignKey(SessionYr, on_delete=models.CASCADE)
    b_id = models.ForeignKey(Batch, on_delete=models.DO_NOTHING)
    c_id = models.ForeignKey(Course, on_delete=models.DO_NOTHING)
    sub_id = models.ForeignKey(Subject, on_delete=models.DO_NOTHING)
    sf_id = models.ForeignKey(CustomUser, on_delete=models.DO_NOTHING)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)
    objects = models.Manager()

class Payment(models.Model):
    pay_id = models.AutoField(primary_key=True)
    st_id = models.ForeignKey(Student, on_delete=models.CASCADE)
    c_id = models.ForeignKey(Course, on_delete=models.CASCADE)
    b_id = models.ForeignKey(Batch, on_delete=models.CASCADE)
    price = models.DecimalField(max_digits=10, decimal_places=2)
    payment_date = models.DateTimeField(auto_now_add=True)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)
    objects = models.Manager()

class LeaveReportS(models.Model):
    lr_id = models.AutoField(primary_key=True)
    lr_date = models.DateField(null=False)
    lr_msg = models.TextField()
    lr_status = models.IntegerField(default=0)
    st_id = models.ForeignKey(Student, on_delete=models.CASCADE)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)
    objects = models.Manager()

class LeaveReportSf(models.Model):
    lrs_id = models.AutoField(primary_key=True)
    lrs_date = models.DateField(null=False)
    lrs_msg = models.TextField()
    lrs_status = models.IntegerField(default=0)
    sf_id = models.ForeignKey(Staff, on_delete=models.CASCADE)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)
    objects = models.Manager()

class FeedBackS(models.Model):
    fb_id = models.AutoField(primary_key=True)
    fb = models.CharField(max_length=255)
    fb_rpy = models.TextField()
    st_id = models.ForeignKey(Student, on_delete=models.CASCADE)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)
    objects = models.Manager()
```

Figure 4-32 :Models.py – Part IV

```python
def doLogin(request):
    if request.method != "POST":
        return HttpResponse("<h2>Method Not Allowed</h2>")
    else:
        user = EmailBackEnd.authenticate(request, email=request.POST.get("email"),
                                         password=request.POST.get("password"))
        if user != None:
            login(request, user)
            if user.user_type == "1":
                return HttpResponseRedirect('/admin_home')
            elif user.user_type == "3":
                return HttpResponseRedirect(reverse("student_home"))
            elif user.user_type == "4":
                return HttpResponseRedirect(reverse("instructor_home"))
            elif user.user_type == "5":
                return HttpResponseRedirect(reverse("examiner_home"))
            elif user.user_type == "6":
                return HttpResponseRedirect(reverse("principal_home"))
            else:
                return HttpResponseRedirect(reverse("staff_home"))
        else:
            messages.error(request, "Invalid Login Details")
            return HttpResponseRedirect("/")

def GetUserDetails(request):
    if request.user != None:
        return HttpResponse("User : " + request.user.email + "usertype :" + request.user.user_type)
    else:
        return HttpResponse("Please Login First")
```

Figure 4-33 : Login Code Structure

```python
import face_recognition
import numpy as np

# Function to load known face encodings from the database
def load_known_face_encodings(students):
    known_encodings = []
    known_ids = []
    for student in students:
        known_image = face_recognition.load_image_file(student.admin.profile_img.path)
        known_encoding = face_recognition.face_encodings(known_image)[0]
        known_encodings.append(known_encoding)
        # Ensure that student ID is converted to an integer if it's not already
        try:
            student_id = int(student.st_idNo)
        except ValueError:
            print(f"Invalid student ID for student: {student}")
            continue  # Skip this student and proceed to the next one
        known_ids.append(student_id)
    return known_encodings, known_ids


# Function to recognize the student's face
def recognize_student_face(known_encodings, unknown_encoding, tolerance=0.6):
    face_distances = face_recognition.face_distance(known_encodings, unknown_encoding)
    match_indices = [index for index, face_distance in enumerate(face_distances) if face_distance <= tolerance]
    if len(match_indices) > 0:
        return match_indices[0]  # Return the index of the first matched face
    return None  # Return None if no match is found
```

Figure 4-34 : Face Recognition code Structure

```
class AddCourseForm(forms.Form):
    c_name = forms.CharField(label="Course Name", max_length=255,
                             widget=forms.TextInput(attrs={"class": "form-control"}))
    c_code = forms.CharField(label="Course Code", max_length=255,
                             widget=forms.TextInput(attrs={"class": "form-control"}))
    type_choice = (("FT", "Full Time"), ("PT", "Part Time"))
    c_type = forms.ChoiceField(label="Type", choices=type_choice,
                               widget=forms.Select(attrs={"class": "select2 form-control"}))

    duration_choice = (
        ("3M", "3 Months"), ("6M", "6 Months"), ("1Y", "1 Year"), ("1Y6M", "1 Year 6 Months"), ("2Y", "2 Year"),
        ("2Y6M", "2 Year 6 Months"),
        ("3Y", "3 Year"), ("3Y6M", "3 Year 6Months"), ("4Y", "4 Year"))
    c_duration = forms.ChoiceField(label="Duration", choices=duration_choice,
                                   widget=forms.Select(attrs={"class": "select2 form-control"}))

    qul_choice = (("NVQ1", "NVQ 01"), ("NVQ2", "NVQ 02"), ("NVQ3", "NVQ 03"), ("NVQ4", "NVQ 04"), ("NVQ5", "NVQ 05"),
                  ("NVQ6", "NVQ 06"), ("Certi", "Certificate"), ("Dip", "Diploma"))
    c_qualification = forms.ChoiceField(label="Qualification", choices=qul_choice,
                                        widget=forms.Select(attrs={"class": "select2 form-control"}))
    tr_list = []
    try:
        trades = Trade.objects.all()
        for trade in trades:
            td = (trade.tr_id, trade.tr_name)
            tr_list.append(td)
    except:
        tr_list = []
    trade = forms.ChoiceField(label="Trade Name", choices=tr_list,
                              widget=forms.Select(attrs={"class": "select2 form-control"}))
```

Figure 4-35 : Form.py – Code Structure

4.7 Summary

This chapter emphasized the design and implementation of the suggested system. This contains the technology, architecture, and tools utilized to create the proposed system modules.

CHAPTER 5 - TESTING AND EVALUATION

This chapter outlines the testing and evaluation strategy for Gampaha Technical College's web-based management information system. This project used an iterative and incremental process for the software development life cycle; testing is presented toward the conclusion of the project life cycle, and it will conclude with user acceptance. This chapter will discuss how the institute-requested requirements were tested for the developed project.

5.1 Requirements Traceability Matrix

The requirements are acquired from Gampaha Technical College, and the test cases are mapped accordingly. Following that, end-to-end functional testing was carried out. After that, beta testing was carried out. Testing with real data in a real-world scenario helped to determine whether the solution is appropriate for the client.

5.2 Relating Testing Types

The web-based management information system can be tested for functionality, reliability, and user satisfaction by undergoing extensive testing across these numerous types, ensuring its successful deployment and operation within the technical college.

5.2.1 Unit Testing

Identify and fix bugs or issues at an early stage of development, while system testing validates that the system meets its requirements and functions correctly in a real-world environment. Both testing approaches are essential for ensuring the quality, reliability, and functionality of the web-based management information system across its entire lifecycle.

5.2.2 Integration Testing

Integration testing is carried out to identify functional faults with the developed module. Each module must be tested individually in order to identify functional modifications as solutions. Validate the integration of the system's various modules and components to ensure flawless communication and data sharing.

5.2.3 System Testing

Evaluates the entire integrated system. It involves testing end-to-end scenarios such as signing in, marking attendance, accessing lesson plans, making online payments, and generating leave reports. System testing also includes integration testing to ensure seamless communication between different modules and components.

5.2.4 User Acceptance Testing(UAT)

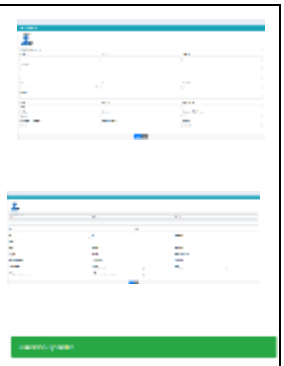
Test the system with end users (students, instructors, and administrators) to ensure it meets their needs and expectations. Obtain feedback on usability, interface design, and the overall user experience in order to make necessary changes before deployment.

5.3 Test Cases

The test case map for the previously mentioned requirement is shown in Table 5-1 : Test Cases below. This section covers the user web portal.

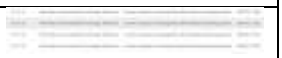
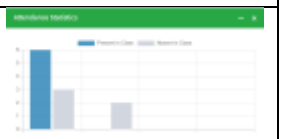
TC No.	Test Scenario	Test Case	Test Steps	Sample Test Data	Expected Result	Actual Result
T1	Validate user can log in to the Web portal with valid credentials	Verify user can log in with the correct email address and password	1. Insert email address 2.Insert Password 3. Click Sign in	Email: admin@gmail.com Password: kba@12#	User should be able to log into the web portal Successfully.	
T2	Validate user cannot log in to the administration portal with invalid credentials.	Verify user cannot log in with valid email but with invalid password	1.Insert email address 2.Insert password 3.Click on Sign in	Email: admin@gmail.com admin@1234	User should not be able to login to the web portal successfully	

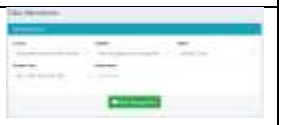
T3		Verify user cannot log in with an invalid email but with a valid password	Insert email address Insert Password Click on sign in	Email: admin12@gmail .com Password: kba@12#	User should not be able to login to the web portal successfully.	
T4		Verify user cannot log in with invalid email address and password	Insert email address Insert Password Click on Sign In	Email: admin1@gmail.com Password: admin@1$	User should not be able to login to the web portal successfully	
T5	Validate user can reset password with forget password option	Verify user can reset the password through forget password option	1.Click on Forgot your password option 2. Enter the registered email address 3. Follow the password reset instructions received via email.		User should be able to reset the password and log in with the new password.	
T6	Validate the user can log out from the web portal	Verify user can log out from the web portal when the clicking Logout button	1. Click on the logout link		User should be able to log out from the web portal successfully	

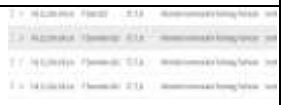
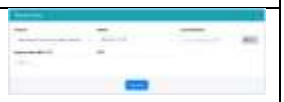
T7	Test admin can add a new user account	Verify the admin can create a new user account	1. Click on the "Add Staff" in the menu under the Staff Menu Or 2. Click on the "Add Student" submenu under the "Student" Menu 3. Fill out all the fields 4. Click on the Save button	1. Upload the user image 2. Fill User Name - Tishan 3. First Name – Tishan 4.Middle Name – Oshanda 5. Last Name – Fernando 6. Email - tishan@gmail.com 7. Password – han#23% 8. DOB: 12-03-1991 9. NIC – 199187656765 10. Address – Veyangoda 11. Gender – Male 12.Staff ID – SF-Civil-01 13. Mobile No – 0768771980	1. The admin should be able to navigate the page. 2. Create new staff or student will appear 3. Admin could be able to insert valid data into all fields successfully 4. If data isn't duplicated, when data saving appears, the successfully saved message	
T8	Validate when the admin inputs an invalid email address	Test when the user input an invalid email address display warning message.	1. Click on the add staff sub menu 2.Input invalid email address 3. Click Save button	Email – tishan	1. The admin should be able to navigate the add staff sub menu 2. The admin should be able to input all fields. 3. When saving display a warning message "please include an "@" in the email address"	

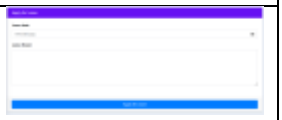
T9	Validate when the user input duplicate user name	Test when the user input a duplicate user name	1. Click on the add staff sub menu 2.Input duplicate user name	User Name – Priyantha	1. The user should be able to navigate the add staff sub menu 2. When the user enters the duplicate user name, it displays "Username Not Available."	
T10	Validate when the user input unique user name	Test when the user input a unique user name	1. Click on the add staff sub menu 2.Input unique user name	User Name – Tishan	1. The user should be able to navigate the add staff sub menu 2. When the user enters the unique user name, it displays "Username Available."	
T11	Test that admin can view added user details	Verify admin should be able to view added user information	1. Click on the staff menu 2. click on the manage staff sub menu		1. The user should be able to navigate to manage staff sub menu 2. Display all the Staff information	
T12	Test that admin can view added user details	Verify admin should be able to view added user information	1. Click on the student menu 2. Click on the manage student sub menu		1. The user should be able to navigate to manage student sub menu 2. Display all the Staff information	

T13	Test that admin can edit user details	Verify admin should be able to edit user informati on	1. Click on the student menu 2. Click on the manage student sub menu 3. Click on the edit button		1. The user should be able to navigate to manage student sub menu 2. Click on the edit button 3. Edit user details. 4. Display "Successfull y Updated" message	
T14	Test that admin can delete the user account	Verify admin should be able to delete user account	1. Click on the student menu 2. Click on the manage student sub menu 3. Click on the delete button		1. The user should be able to navigate to manage student sub menu 2. Click on the edit button 3. Edit user details. 4. Display "Delete Successfully " message	
T15	Verify the instructor can take student attendance	Test take attendan ce	1. Log in as an instructor 2. Navigate to the FR Take attendace Menu		View the FR Take attendance page	
T16	Verify the functionalit y of making attendance using unknown face	Test attendan ce marking with unknown face	1. Log in as an instructor 2. Navigate to the FR Take attendace Menu		Display "Student not found in the system. Please register to the system" message	

T17	Verify the functionality of making attendance using known face	Test attendance marking with known face	1. Log in as an instructor 2. Navigate to the FR Take attendace Menu		Display “Student attendance marked as present” message	
T18	Verify the functionality of making attendance using same face more than once within the given time period	Test attendance marking with same face more than once within given time period	1. Log in as an instructor 2. Navigate to the FR Take attendace Menu		Display “Student attendance has already been taken within the last given minutes” message	
T19	Verify that attendance records are synchronized with the database in real-time.	Test attendance syncing with the database	1. Login as an administrator 2. Monitor the database for real-time updates during attendance marking		Attendance records should be updated promptly in the database as attendees are recognized.	
T20	Test if attendance records are viewed in the View Attendance	Verify real-time attendance updates	1. Login as an instructor 2. Navigate view attendance		Viewed on the View Attendance page	
T21	Test if attendance records are updated in real-time on the instructor's dashboard.	Verify real-time attendance updates.	1. Log in as an instructor. 2. Monitor the attendance dashboard during attendance marking.		Attendance records should reflect real-time updates as attendees are recognized.	

T22	Verify that instructors can generate attendance reports.	Test attendance reporting functionality.	1.Log in as an instructor. 2. Access the attendance reporting section. 3. Select the desired time period for the report. 4. Generate the attendance report.		Attendance report should be generated accurately based on the specified time period.	
T23	Verify the examiner can take student attendance	Test take attendance	1. Log in as an Examiner 2. Navigate to the FR Take attendace Menu		View the FR Take attendance page	
T24	Verify the functionality of making attendance using unknown face	Test attendance marking with unknown face	1. Log in as an examiner 2. Navigate to the FR Take attendace Menu		Display “Student not found in the system. Please register to the system” message	
T25	Verify the functionality of making attendance using known face	Test attendance marking with known face	1. Log in as an examiner 2. Navigate to the FR Take attendace Menu		Display “Student attendance marked as present” message	
T26	Verify the functionality of making attendance using same face more than once within the given time period	Test attendance marking with same face more than once within given time period	1. Log in as an examiner 2. Navigate to the FR Take attendace Menu		Display “Student attendance has already been taken within the last given minutes” message	

T27	Verify that attendance records are synchronized with the database in real-time.	Test attendance syncing with the database	1. Login as an administrator 2. Monitor the database for real-time updates during exam attendance marking		Attendance records should be updated promptly in the database as attendees are recognized.	
T28	Test if attendance records are viewed in the View Attendance	Verify real-time attendance updates	1. Login as an instructor 2. Navigate view attendance		Viewed on the View Attendance page	
T29	Verify that examiner can generate exam attendance reports.	Test attendance reporting functionality.	1.Log in as an examiner. 2. Access the attendance reporting section. 3. Select the desired time period for the report. 4. Generate the attendance report.		Attendance report should be generated accurately based on the specified time period.	
T30	Verify that students can successfully make payments for enrolled courses. Test Steps:	Test course payment process for students.	1.Log in as a student. 2.Navigate to the course payment section. 3.Select the course for payment. 4.Proceed to the payment gateway. 5.Complete the payment process.		Display “Student should receive a confirmation of successful course payment.” message	
T31	Test if students can apply for leaves through the system.	Verify leave application process	1.Log in as a student. 2.Navigate to the leave application section.		Leave application should be successfully submitted for review by	

		for students.	3.Fill out the leave application form with details. 4.Submit the leave application.		the concerned authority.	
T32	Verify that staff members can approve leave applications submitted by students.	Test leave approval process for staff.	1.Log in as a staff member with leave approval authority. 2.Access the leave approval dashboard. 3.Review pending leave applications. 4.Approve or reject leave applications as appropriate.		Leave applications should be processed promptly with clear status updates for students.	
T33	Test if instructor can apply for leaves through the system.	Verify leave application process for instructor.	1.Log in as an instructor. 2.Navigate to the leave application section. 3.Fill out the leave application form with details. 4.Submit the leave application.		Leave application should be successfully submitted for review by the concerned authority.	
T34	Verify that principal can approve leave applications submitted by instructor.	Test leave approval process for principal.	1.Log in as a principal with leave approval authority. 2.Access the leave approval dashboard.		Leave applications should be processed promptly with clear status updates for instructors.	

			3.Review pending leave applications. 4.Approve or reject leave applications as appropriate.			
T35	Verify send the feedback message	Test Send the feedback message	1. Login as student or instructor 2.Fill out the feedback message details		Display “Successfully send feed back” and get notification	

Table 5-1 : Test Cases

5.4 Test Results

Table 5.2 refers to the actual test results of the above test cases.

Requirement Description	TC No	Test Status
Administrator: The administrator can access the login page with authorized credentials and log out from the logged user.	T1 T2 T3 T4 T5 T6	Pass Pass Pass Pass Pass Pass
Administrator: The admin can add a new user account with following inputs. 1. Upload the user image 2. Fill User Name 3. First Name 4.Middle Name 5. Last Name 6. Email 7. Password 8. DOB 9. NIC	T7	Pass Pass Pass Pass

10. Address 11. Gender 12.Staff ID 13. Mobile No		
Web Portal: The system validates the below fields 1. Email address 2. User Name	T8 T9 T10	Pass Pass Pass
Administrator: The administrator can view the user account details	T11 T12	Pass Pass
Administrator: The administrator can edit the user account details	T13	Pass
Administrator: The administrator can delete the user account	T14	Pass
Instructor: The Instructor can take students attendance	T15 T16 T17 T18 T19 T20 T21 T22	Pass Pass Pass Pass Pass Pass Pass Pass
Administrator: The Administrator can test the attendance syncing with database	T19	Pass
Examiner: The Examiner can take students attendance	T23 T24 T25 T26 T28 T29	Pass Pass Pass Pass Pass Pass
Administrator: The Administrator can test the exam attendance syncing with database	T27	Pass

Student: The student can pay course fee through the online	T30	Pass
Leave Reports for students and staff	T31 T32 T33 T34	Pass Pass Pass Pass
Feedback message	T35	Pass

Table 5-2 : Test Result

5.5 User Evaluation

The developed system is evaluated by users to determine whether or not it meets the specifications. Following User Acceptance Testing (UAT) for the questioner process via Google Form Survey, the main actors in the process were chosen as users. Figure 5-1 :User Evaluation Form – Part IFigure 5-2 :User Evaluation Form – Part II refer to evaluation forms sent to users to collect their feedback regarding the developed system.

The evaluation included the following staff members from Gampaha Technical College: For each department section, employees have been selected from across the Technical College.

Administrator – 1

Principal – 1

Academic Staff -6

Students - 4

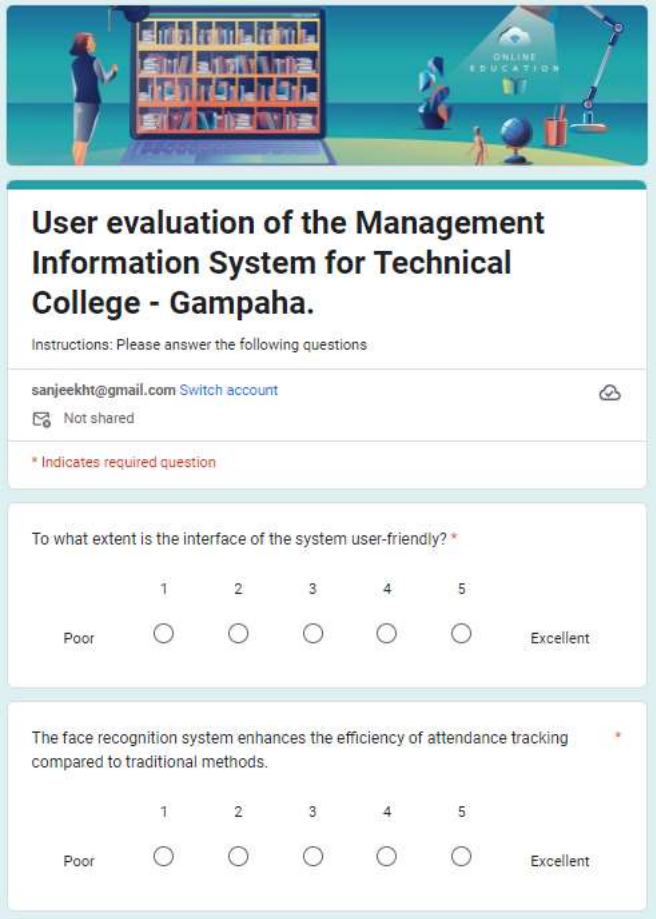


Figure 5-1 :User Evaluation Form – Part I

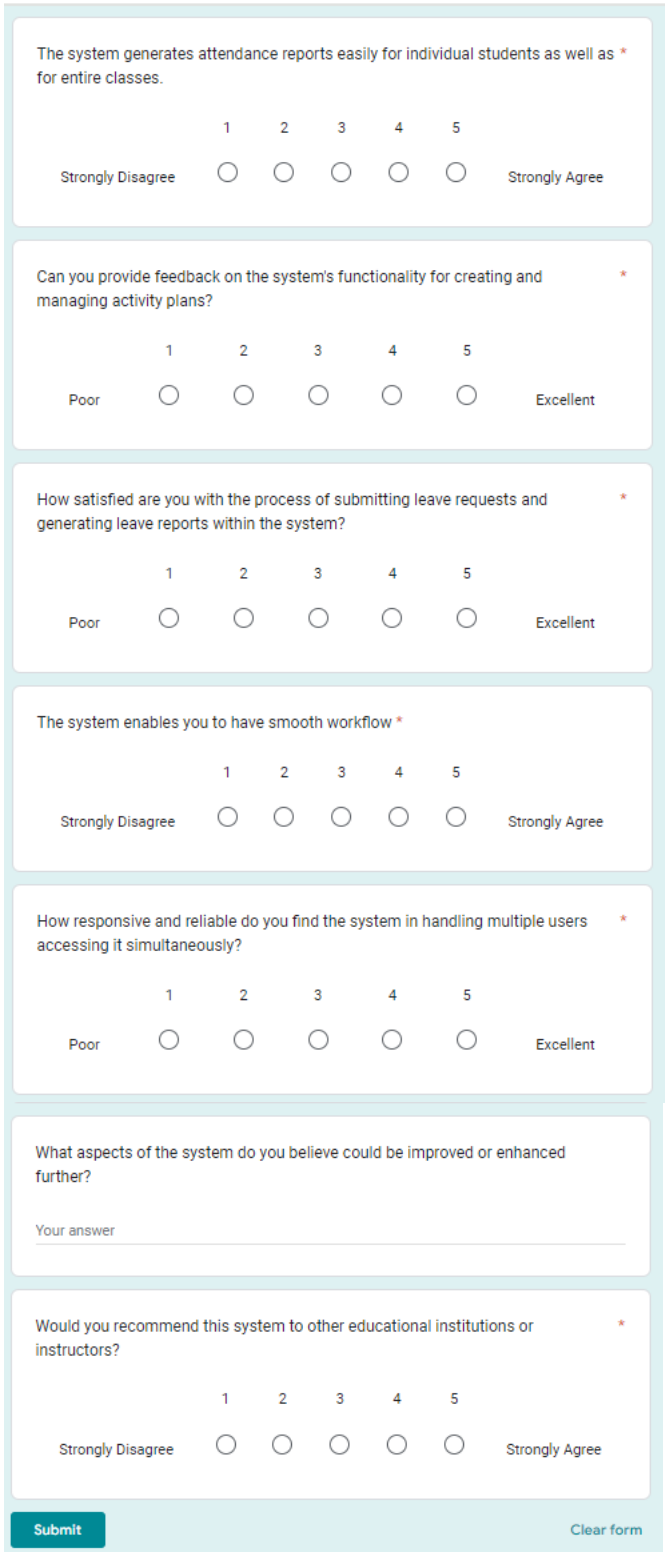


Figure 5-2 :User Evaluation Form – Part II

5.5.1 The Result of the User Acceptance Testing

The survey was conducted among administrators, staff, and students at Gampaha Technical College; survey results are appended to Figures 5.3-5.11.

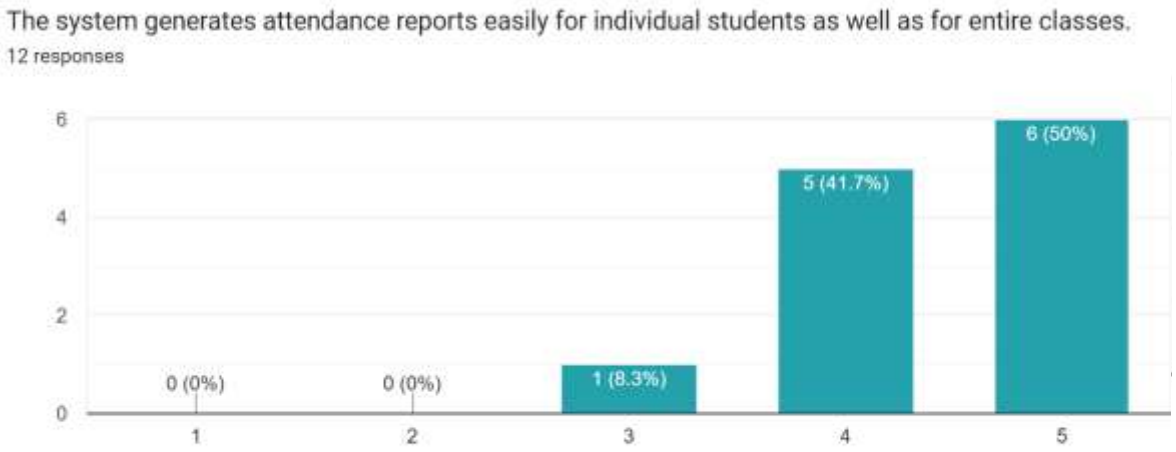


Figure 5-3 :Survey Feedback – I

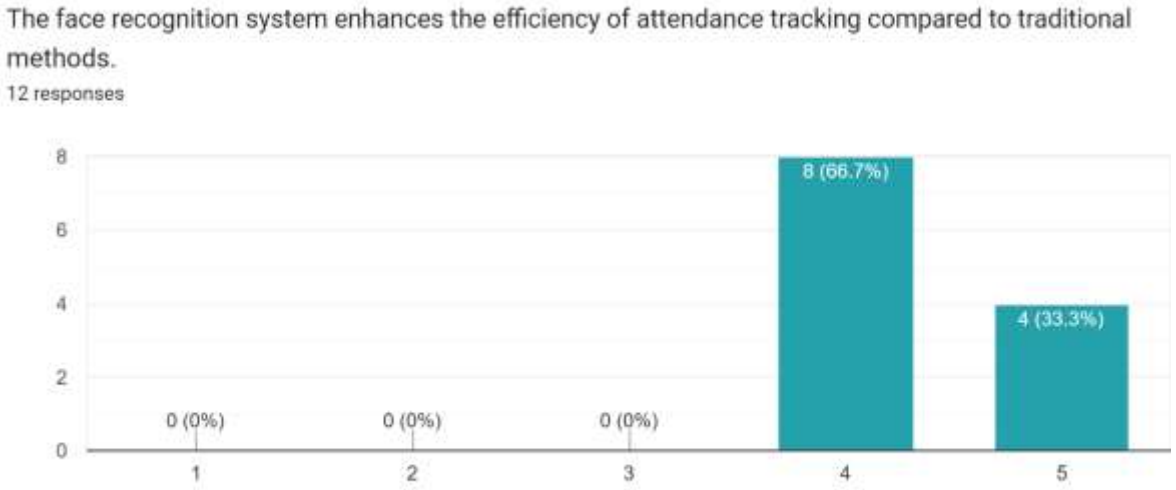


Figure 5-4 : Survey Feedback – II

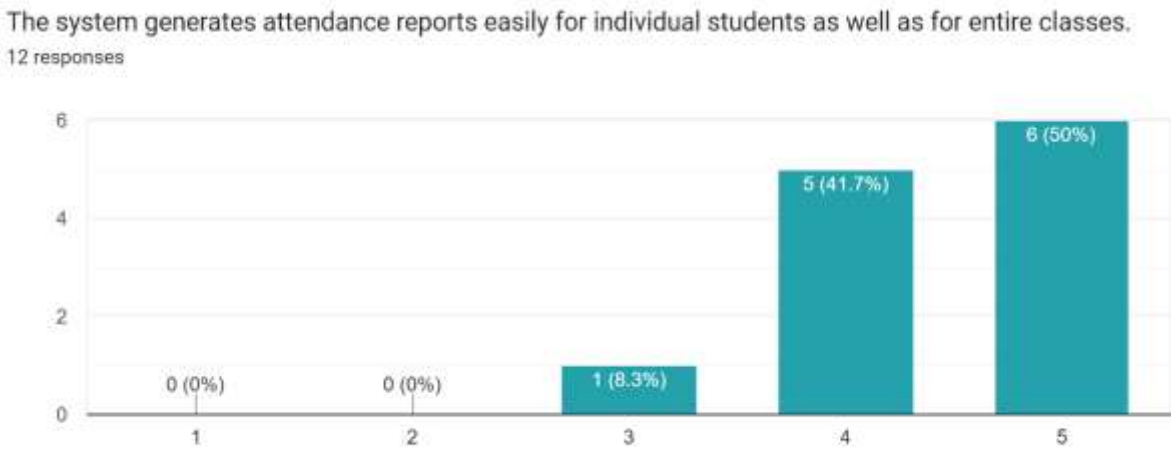


Figure 5-5 : Survey Feedback – III

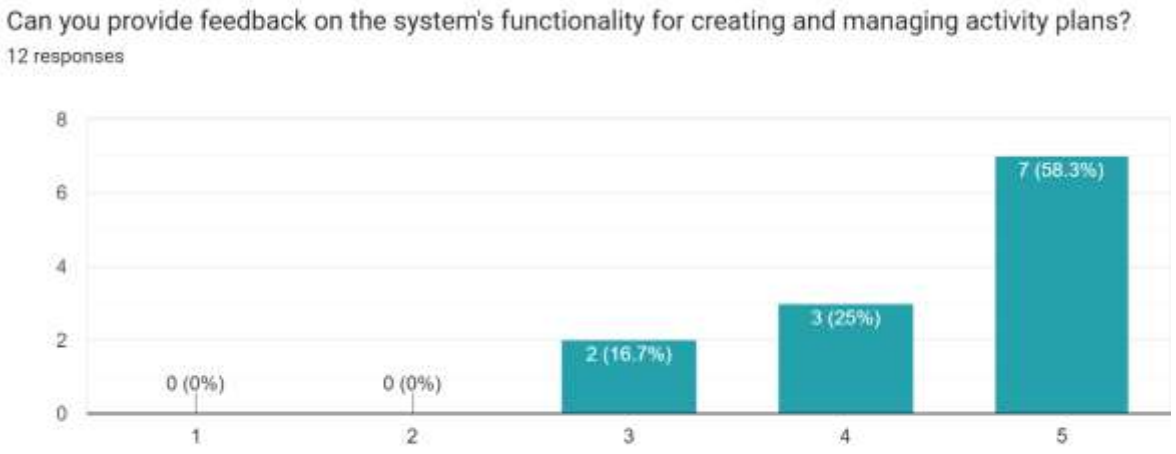


Figure 5-6 : Survey Feedback – IV

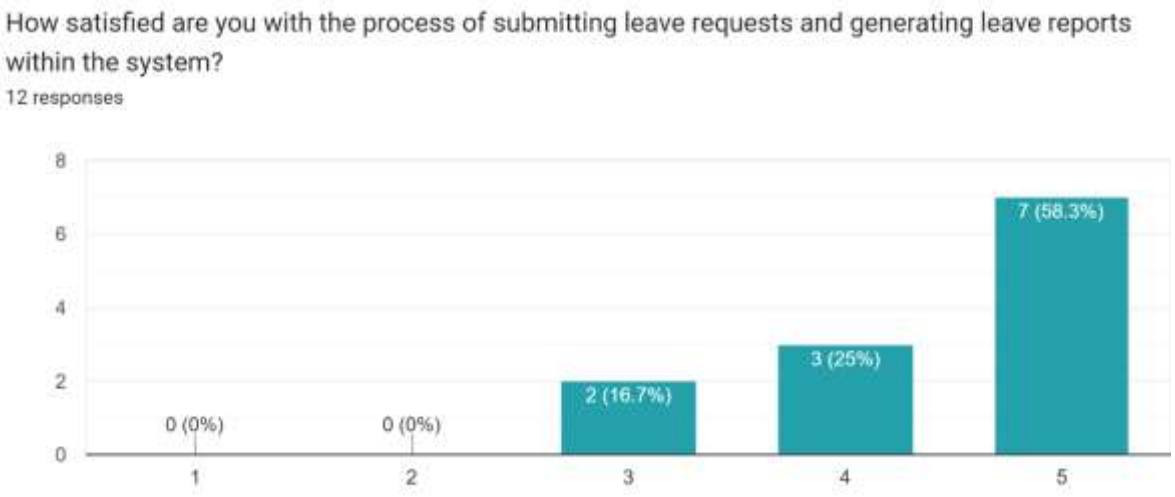


Figure 5-7 : Survey Feedback – V

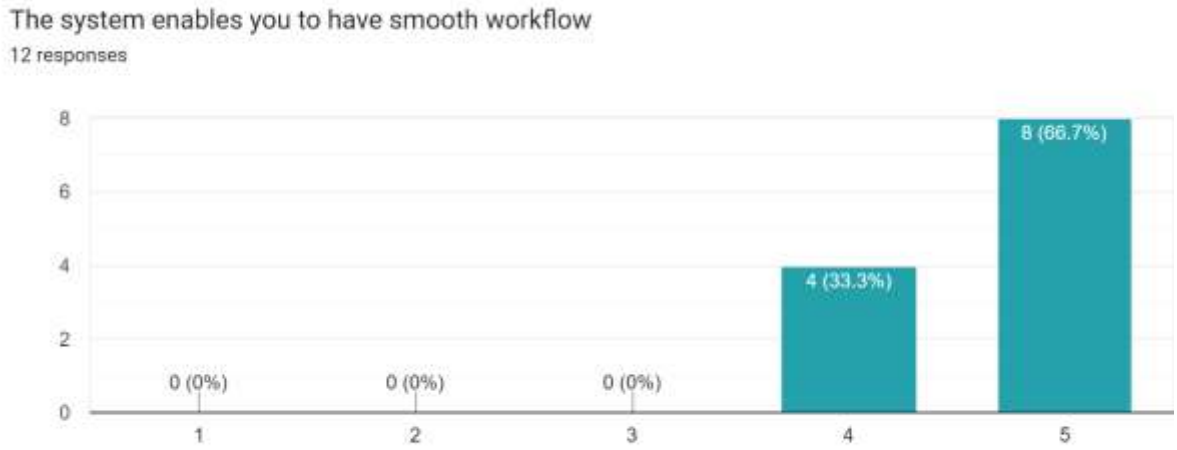


Figure 5-8 :Survey Feedback – VI

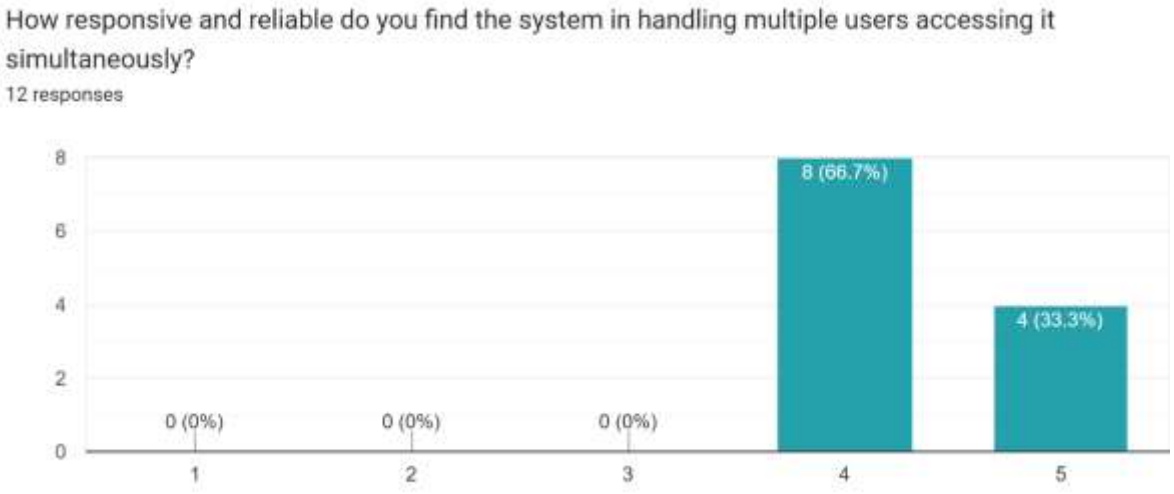


Figure 5-9 : Survey Feedback – VII

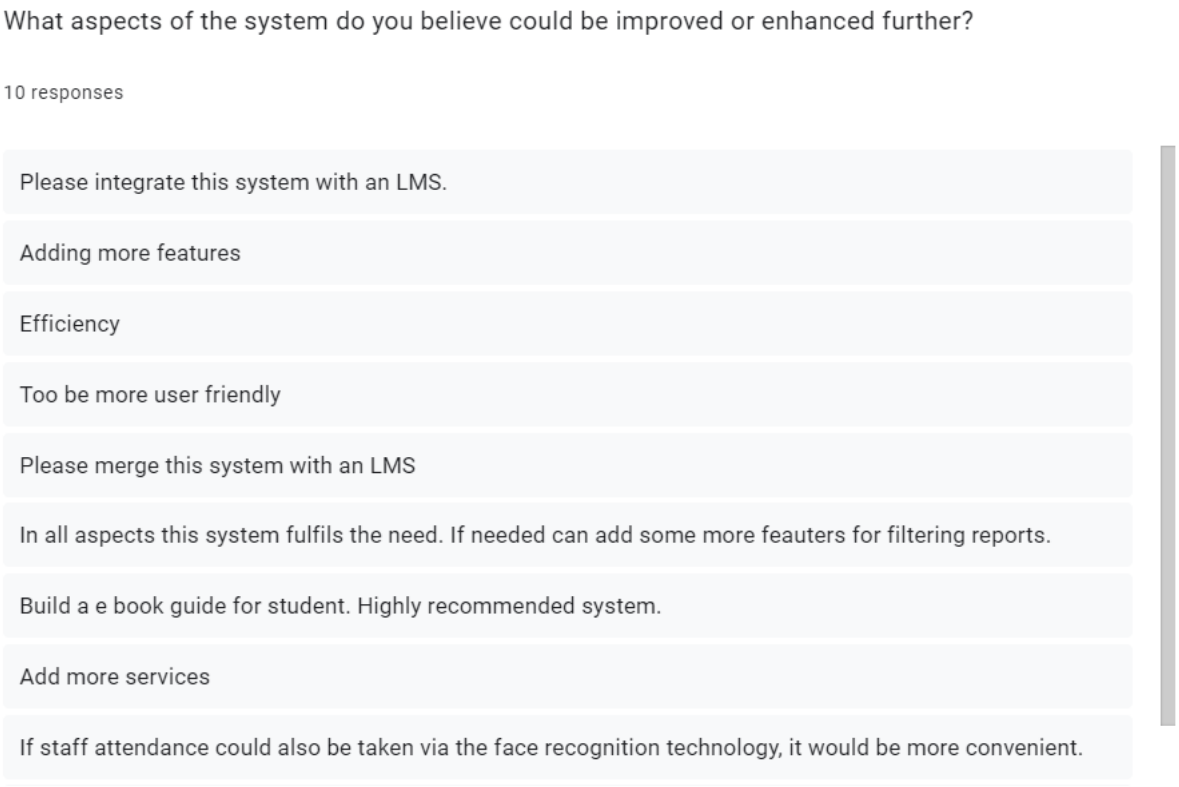


Figure 5-10 : Survey Feedback – VIII

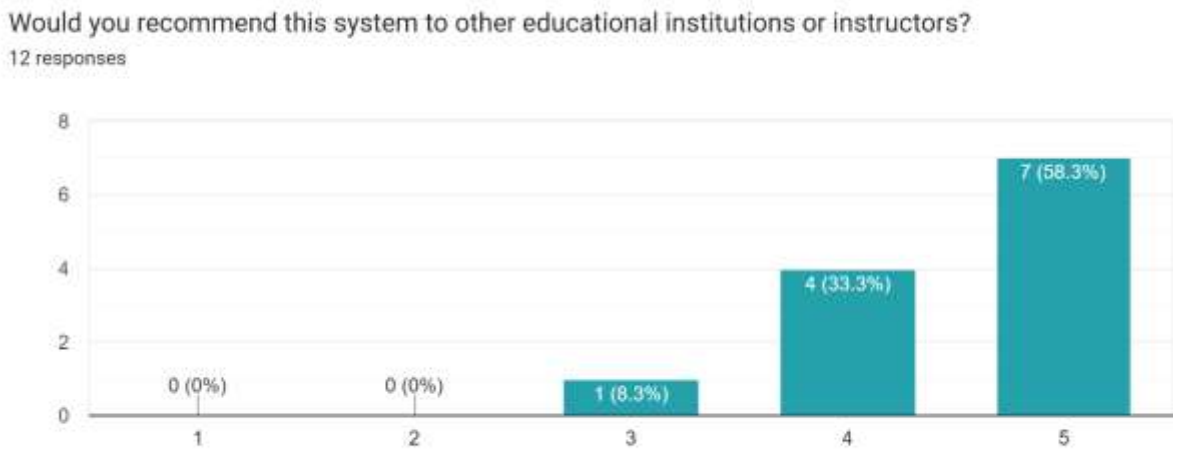


Figure 5-11 : Survey Feedback – IX

5.6 Summary

This chapter is about the testing and evaluation process of the entire system. The evaluation helped to measure how far the system helped to meet the project objectives of the business process. Moreover, the chapter discussed the techniques used for the testing and evaluation according to user feedback. The next chapter will give the future work and the brief on the entire project.

CHAPTER 6 - CONCLUSION

This chapter defines the project's overall conclusion. This focuses mostly on what has been discovered and how those apply to various scenarios. This also contains the lessons learned, project processes, project outcomes, and future improvements.

6.1 Overview of the developed system

The current method of managing student and staff information at the technical colleges involves a significant amount of paper usage, which is not sustainable due to the shortage of paper caused by the nation's economic crisis and the government's decision to cut back on providing paper to government institutes. In the Gampaha Technical College, the lecturer or instructor should develop the essential fundamental plans (year plan, assessment plan, term notes, lesson plan) prior to the start of the teaching-learning process in the classroom in accordance with the pertinent curriculum material. The developed plans should be implemented by the due date and time, and the instructor should prepare the data and papers needed to measure their efficacy and efficiency. In one semester, one instructor or lecturer teaches many courses as well as a variety of subjects. Making lesson plans for every subject taught as well as putting together pertinent records (daily teaching logs, monthly delivery reports), papers (skill standards, curriculum, training guides, and teaching guides), and tasks takes a lot of time and effort. Planning, writing, and reporting are essential components of the teaching methods. However, it will have an impact on an instructor’s effectiveness if they are too busy to remember or disregard them during the process.

Also, students' attendance is considered a critical aspect for both students and instructors/lecturers in technical colleges. Students who attend technical colleges on a regular basis enhance the quality of tertiary educational technology. Manual student attendance marking is currently plagued by numerous problems and is a time-consuming process.

The project aims to address these problems by developing a digital solution for managing student and staff information, reducing the reliance on paper, and improving the efficiency of administrative tasks.

The project proceeded with requirement gathering, which included multiple interviews and conversations with employees, the head of the department, academic staff, and some Gampaha Technical College students, covering all institutes and departments to obtain all of their information. The project utilized iterative and incremental methods as the SDLC methodology.

According to the requirements of the ICT sector, the final product must be hosted on their internal server. The implementation utilizes the Django framework and MySQL as database services. Once the implementation and functional end-to-end testing were complete, stakeholders conducted user acceptance testing (UAT). A survey was conducted among stakeholders involved in the requirement-gathering stage using Google Forms, and the results were examined. According to their response, the system was satisfactory, so they adopted it as a pilot project for a management information system.

6.2 Lessons Learnt

The knowledge gained throughout the project was extremely useful. From feasibility studies to the end of development, this procedure provided unparalleled experience in many aspects. This was a great opportunity for me to learn about a variety of backend and frontend technologies, including Django, Python, MySQL, face recognition libraries, and online payment mechanisms. Additionally, extra effort was made to become familiar with the MVT design approach. Furthermore, working on the project encourages me to strengthen both my technical and intellectual talents by collaborating with numerous people from various professions. Effective time management and planning are essential skills to acquire throughout this project. Additionally, business process flaws can be addressed through a computerized system. This provided me with an opportunity to work through the entire SDLC process.

6.3 Future Improvement

According to the user acceptance test conducted during the evaluation phase, the user response was optimistic and inspiring. The following features are planned to be added to the newly developed system as future enhancements:

Enhanced Face Recognition Technology:

- Invest in research and development to improve the accuracy and reliability of face recognition algorithms.
- Explore the integration of advanced facial recognition features, such as emotion detection and fatigue monitoring, to provide additional insights into student engagement and well-being.

Mobile Compatibility:

- Develop a mobile application version of the system to enable users to access key functionalities on their smartphones and tablets.
- Ensure cross-platform compatibility and responsiveness to provide a seamless user experience across devices.

Predictive Analytics and Reporting:

- Implement predictive analytics capabilities to forecast attendance patterns, identify potential issues, and proactively address them.
- Enhance reporting functionalities to generate customizable reports and visualizations for administrators, instructors, and students, providing valuable insights into attendance, course performance, and leave trends.

Integration with Learning Management Systems (LMS):

- Integrate the system with popular learning management systems to streamline data sharing and course management processes.
- Enable seamless integration of course materials, assignments, and assessments between the management system and the LMS, providing a unified learning experience for students and instructors.

REFERENCES

- Al-Badri, R., Hasani, S.A., 2019. Class Attendance Record Based Face Recognition Using Rasperry Pi 7, 5.
- Arulogun, O.T., Olatunbosun, A., Fakolujo, O.A., Olaniyi, O.M., 2013. RFID-based students attendance management system 1–9.
- B, L., 2022. Things to Know About Django Beforehand [WWW Document]. Medium. URL https://levelup.gitconnected.com/things-to-know-about-django-beforehand-f93e4e3319ed (accessed 6.23.23).
- Bandara, W.B.K., Chandike, B.M.T., Samaraweera, G.C., 2022. Evaluation of Student Dropout Rate in Alternative Higher Education Sector of Sri Lanka: A Case Study 17, 59. https://doi.org/10.4038/ouslj.v17i2.7580
- Bootstrap · The most popular HTML, CSS, and JS library in the world. [WWW Document], n.d. URL https://getbootstrap.com/ (accessed 6.22.23).
- Chung, L., Leite, J., 2009. On Non-Functional Requirements in Software Engineering. pp. 363–379. https://doi.org/10.1007/978-3-642-02463-4_19
- Divyansh, sh, n.d. Face-recognition [WWW Document]. URL https://github.com/Divyansh6799/Face-recognition-in-python-django (accessed 6.23.23).
- Django CSS [WWW Document], n.d. URL https://pythonprogramming.net/css-django-tutorial/ (accessed 6.23.23).
- Django JavaScript [WWW Document], n.d. . Django Project. URL https://docs.djangoproject.com/en/dev/internals/contributing/writing-code/javascript/ (accessed 6.23.23).
- Django overview | Django [WWW Document], n.d. URL https://www.djangoproject.com/start/overview/ (accessed 9.3.22).
- Django Project MVT Structure, 2019. . GeeksforGeeks. URL https://www.geeksforgeeks.org/django-project-mvt-structure/ (accessed 8.31.22).
- Django Template [WWW Document], n.d. . Django Project. URL https://docs.djangoproject.com/en/4.2/ref/templates/language/ (accessed 6.23.23).
- Django [WWW Document], n.d. . Django Project. URL https://docs.djangoproject.com/en/4.2/misc/design-philosophies/ (accessed 6.23.23).
- django-excel [WWW Document], n.d. URL https://django.pyexcel.org/en/latest/ (accessed 6.23.23).
- Geitgey, A., n.d. face-recognition: Recognize faces from Python or from the command line.
- Imene, F., Imhanzenobe, J., 2020. Information technology and the accountant today: What has really changed? JAT 12, 48–60. https://doi.org/10.5897/JAT2019.0358
- Khatun, A., Haque, A.F., Ahmed, S., Rahman, M.M., 2015. Design and implementation of iris recognition based attendance management system, in: 2015 International Conference on Electrical Engineering and Information Communication Technology (ICEEICT). Presented at the 2015 International Conference on Electrical Engineering and Information Communication Technology (ICEEICT), IEEE, pp. 1–6. https://doi.org/10.1109/ICEEICT.2015.7307458
- Larman, C., 2012. Applying UML and patterns [WWW Document]. GitHub. URL https://github.com/AraNaldinho/Applying-uml-and-patterns-3rd-editiom/blob/master/Applying%20UML%20and%20Patterns%203rd%20Edition.pdf (accessed 8.11.23).

- Li, S., Kot, A.C., 2012. An Improved Scheme for Full Fingerprint Reconstruction. IEEE Transactions on Information Forensics and Security 7, 1906–1912. https://doi.org/10.1109/TIFS.2012.2212012
- Middleware — Django 1.4.22 documentation [WWW Document], n.d. URL https://django.readthedocs.io/en/1.4.X/topics/http/middleware.html (accessed 3.4.24).
- MS Visio [WWW Document], 2012. . groovyPost. URL https://www.groovypost.com/reviews/microsoft-visio-explained/ (accessed 6.23.23).
- MuthuKalyani.K, VeeraMuthu.A, 2013. smart application for AMS using Face Recognition [WWW Document]. URL https://www.semanticscholar.org/paper/smart-application-for-AMS-using-Face-Recognition-MuthuKalyani.K-VeeraMuthu.A/99db2f02be6044afdd1c0a45e5fb8aa12ff5d71d (accessed 7.28.22).
- Nige, B., 2022. Mastering Django: Structure. Mastering Django. URL https://masteringdjango.com/django-tutorials/mastering-django-structure/ (accessed 2.17.23).
- OpenCV, 2023. . Wikipedia.
- Popy, H.P., 2020. Online Course Management System.
- Pressman, R.S., 2000. Software engineering: a practitioner's approach, 5th ed. ed, McGraw-Hill series in computer science. McGraw Hill, Boston, Mass.
- Sommerville, I., 2011. Software engineering, 9th ed. ed. Pearson, Boston.
- Team, D., 2019. Django Bootstrap | An Essential Framework to beat your competitors! [WWW Document]. DataFlair. URL https://data-flair.training/blogs/django-bootstrap/ (accessed 6.23.23).
- Team, G.L., 2021. OpenCV Tutorial: A Guide to Learn OpenCV in Python [WWW Document]. Great Learning Blog: Free Resources what Matters to shape your Career! URL https://www.mygreatlearning.com/blog/opencv-tutorial-in-python/ (accessed 2.17.23).
- Vyas, K., 2023. Advantages of Using MySQL [WWW Document]. Datamation. URL https://www.datamation.com/storage/8-major-advantages-of-using-mysql/ (accessed 6.23.23).

APPENDIX A - SYSTEM DOCUMENTATION

The system documentation could be referred to if there are any changes or enhancements to be made in the application. The system administrator must follow the steps outlined in this handbook to configure the implementation environment.

To develop the system locally, the software listed below must be installed.

PyCharm 2021.3.2

- opencv-python4.7.0.72
- Django4.2.2 5.0.3
- Pillow 9.5.0 10.2.0
- PyPDF2 1.28.6 3.0.1
- PyYAML 6.0.1 6.0.1
- Strip 0.0.1.dev14
- dlib 19.24.2 19.24.2
- face-api 0.2.8 0.2.8
- face-recognition 1.3.0 1.3.0
- face-recognition-models 0.3.0
- mysql-connector 2.2.9 2.2.9
- mysqlclient 2.2.0 2.2.4
- numpy 1.25.0 1.26.4
- pytz 2023.3
- qrcode 7.4.2
- reportlab 4.0.7
- sqlparse 0.4.4
- stripe 8.1.0

MySQL Workbench 8.0

Minimum requirements of hardware and software for the installation are as follows:

Supported Operating Systems	• Windows 8 or later
Hardware	1.8 GHz or faster processor. Quad-core or better recommended ● 2 GB of RAM; 8 GB of RAM recommended (2.5 GB minimum if running on a virtual machine) ● Hard disk space: Minimum of 1GB up to 20 GB of available space, depending on features installed; ● Hard disk speed: to improve performance, install Windows and Visual Studio on a solid-state drive (SSD).
Additional Requirements	A web browser (Chrome/Firefox preferred)

Table A - 1 : Minimum hardware and software requirement

Set the database connection in the PyCharm settings.py application as shown in Figure A - 1 : Database Settings.py.

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'mis_tech',
        'USER': 'mistech',
        'PASSWORD': 'mistech',
        'HOST': 'localhost',
        'PORT': '3306',
    }
}
```

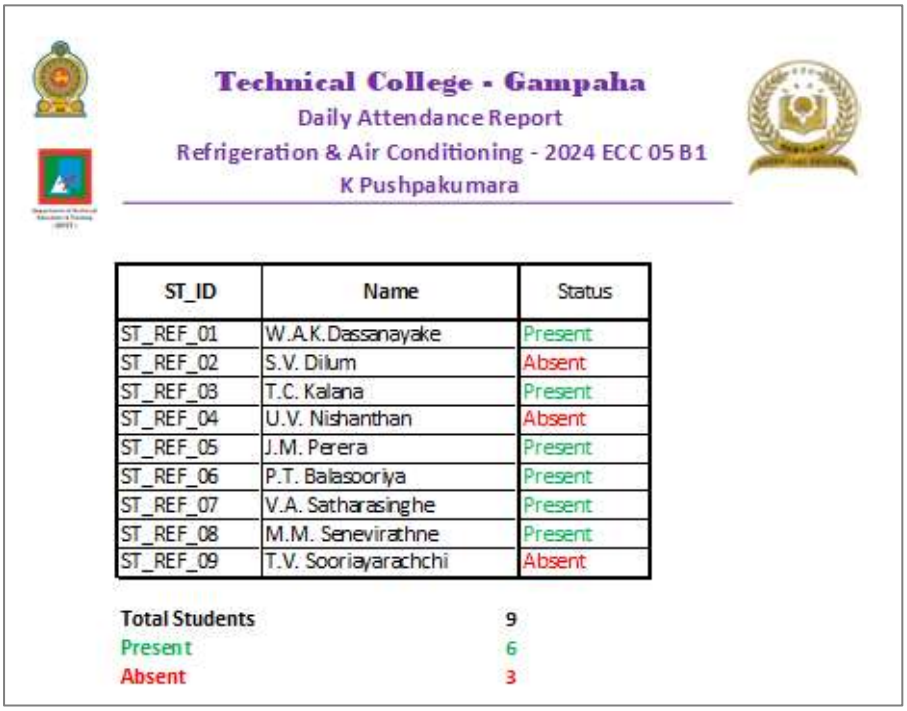
Figure A - 1 : Database Settings.py

APPENDIX B – MIS REPORTS

MIS refers to Management Information System reports, which provide management with summary information. Academic staff members' progress can be monitored according to a MIS report that highlights the institute's daily activities. The Web-based management information system for technical college Gampaha provides a variety of reports connected to everyday activities, and each MIS report module is outlined with convincing evidence.

Daily Attendance Report

The Instructor could be able to generate a daily attendance report after tracking attendance. Figure B - 1 : Daily Attendance Report depicts the daily attendance record for the students.



Technical College - Gampaha

Daily Attendance Report

Refrigeration & Air Conditioning - 2024 ECC 05 B1

K Pushpakumara

ST_ID	Name	Status
ST_REF_01	W.A.K.Dassanayake	Present
ST_REF_02	S.V. Dilum	Absent
ST_REF_03	T.C. Kalana	Present
ST_REF_04	U.V. Nishanthan	Absent
ST_REF_05	J.M. Perera	Present
ST_REF_06	P.T. Balasooriya	Present
ST_REF_07	V.A. Satharasinghe	Present
ST_REF_08	M.M. Senevirathne	Present
ST_REF_09	T.V. Sooriyarachchi	Absent

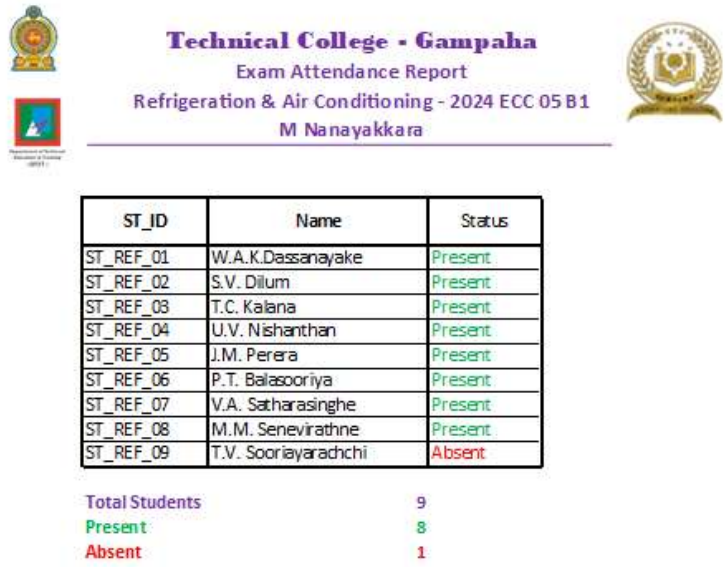
Total Students	9
Present	6
Absent	3

Figure B - 1 : Daily Attendance Report

Exam Attendance Report

The exam attendance report serves as a vital tool for educational institutions to monitor and evaluate student attendance during important examination periods. This report Figure B - 2 : Exam Attendance provides a comprehensive analysis of attendance data gathered before,

during, and after exams, offering valuable insights into student engagement and participation levels.



Technical College - Gampaha

Exam Attendance Report

Refrigeration & Air Conditioning - 2024 ECC 05 B1

M Nanayakkara

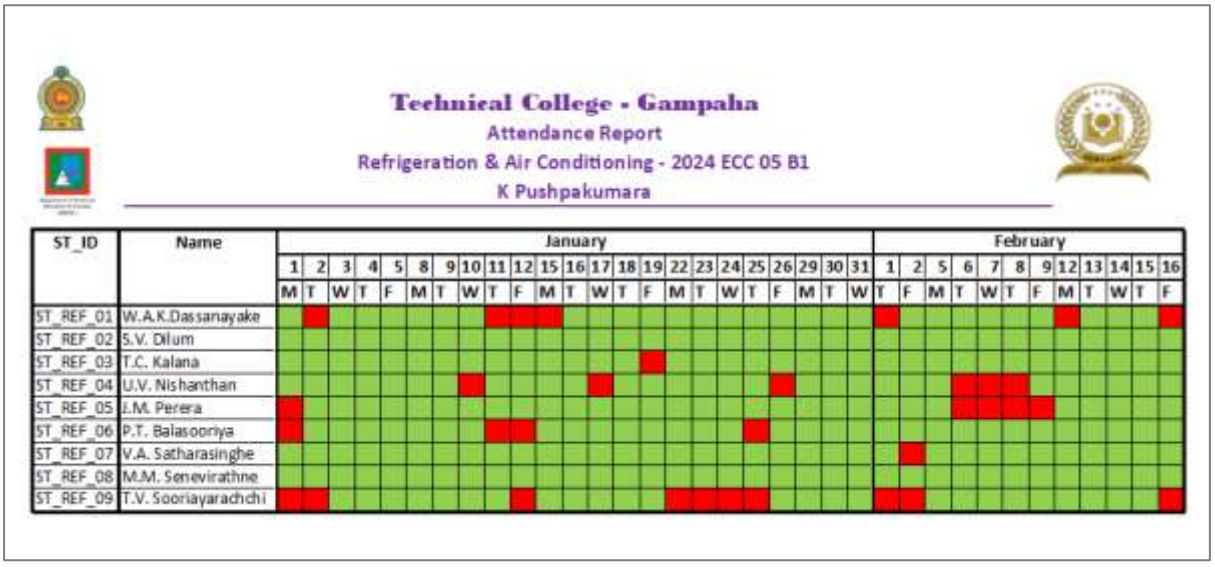
ST_ID	Name	Status
ST_REF_01	W.A.K.Dassanayake	Present
ST_REF_02	S.V. Dilum	Present
ST_REF_03	T.C. Kalana	Present
ST_REF_04	U.V. Nishanthan	Present
ST_REF_05	J.M. Perera	Present
ST_REF_06	P.T. Balasooriya	Present
ST_REF_07	V.A. Satharasinghe	Present
ST_REF_08	M.M. Senevirathne	Present
ST_REF_09	T.V. Sooriayarachchi	Absent

Total Students	9
Present	8
Absent	1

Figure B - 2 : Exam Attendance

Attendance Report

The report indicates the with given date range could be able to generate attendance report as user request. The Figure B - 3 : Attendance Summary Reportshows the filtered attendance by given date range.



Technical College - Gampaha

Attendance Report

Refrigeration & Air Conditioning - 2024 ECC 05 B1

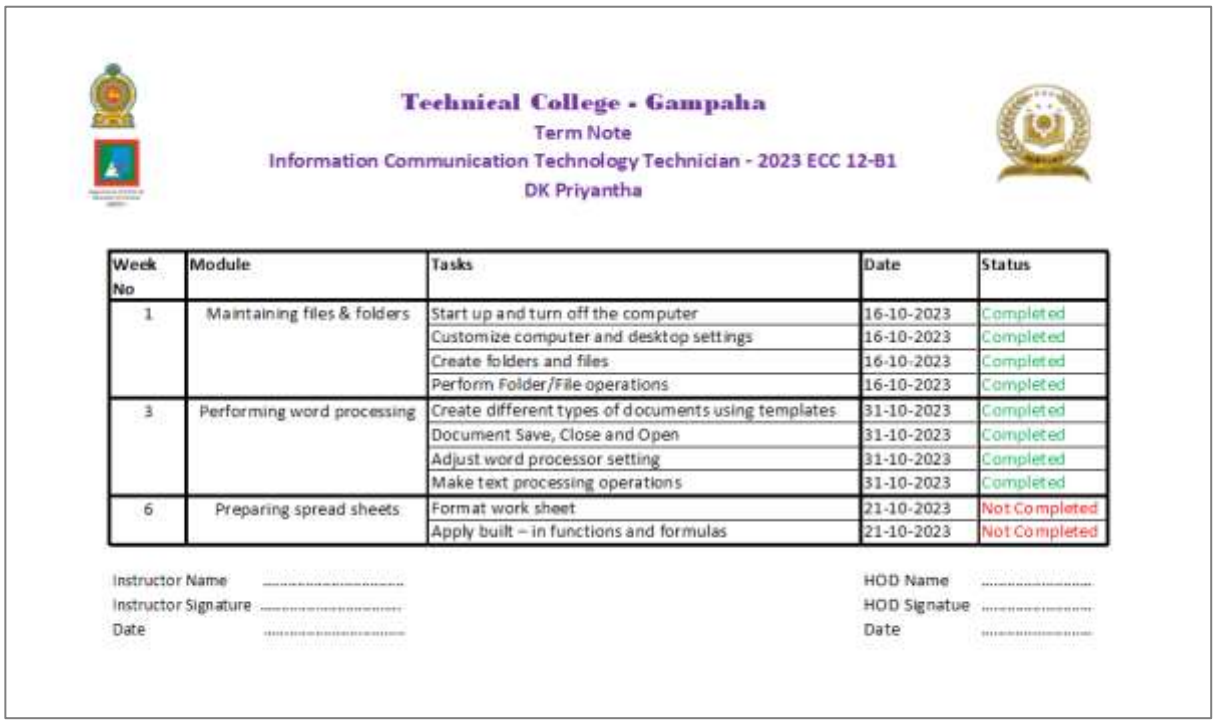
K Pushpakumara

ST_ID	Name	January																										February												
		1	2	3	4	5	8	9	10	11	12	15	16	17	18	19	22	23	24	25	26	29	30	31	1	2	5	6	7	8	9	12	13	14	15	16				
		M	T	W	T	F	M	T	W	T	F	M	T	W	T	F	M	T	W	T	F	M	T	W	T	F	M	T	W	T	F	M	T	W	T	F				
ST_REF_01	W.A.K.Dassanayake																																							
ST_REF_02	S.V. Dilum																																							
ST_REF_03	T.C. Kalana																																							
ST_REF_04	U.V. Nishanthan																																							
ST_REF_05	J.M. Perera																																							
ST_REF_06	P.T. Balasooriya																																							
ST_REF_07	V.A. Satharasinghe																																							
ST_REF_08	M.M. Senevirathne																																							
ST_REF_09	T.V. Sooriayarachchi																																							

Figure B - 3 : Attendance Summary Report

Term Notes

According to the year plan, the instructor should prepare this report at the start of each academic term. The instructor creates the term note using the curriculum, academic calendar, and personal timetable. The Figure B - 4 : Term Note



Technical College - Gampaha

Term Note

Information Communication Technology Technician - 2023 ECC 12-B1

DK Priyantha

Week No	Module	Tasks	Date	Status
1	Maintaining files & folders	Start up and turn off the computer	16-10-2023	Completed
		Customize computer and desktop settings	16-10-2023	Completed
		Create folders and files	16-10-2023	Completed
		Perform Folder/File operations	16-10-2023	Completed
3	Performing word processing	Create different types of documents using templates	31-10-2023	Completed
		Document Save, Close and Open	31-10-2023	Completed
		Adjust word processor setting	31-10-2023	Completed
		Make text processing operations	31-10-2023	Completed
6	Preparing spread sheets	Format work sheet	21-10-2023	Not Completed
		Apply built – in functions and formulas	21-10-2023	Not Completed

Instructor Name .....

Instructor Signature .....

Date .....

HOD Name .....

HOD Signatue .....

Date .....

Figure B - 4 : Term Note

Lesson Plan

The lesson plan document defines how students should participate in the learning process, the teaching approach, and the time range to demonstrate a unit in a subject. The Figure B - 5 : Lesson Plan report.

Technical College - Gampaha

Lesson Plan

Information Communication Technology Technician - 2023 ECC 12-B1

DK Priyantha

Instructor Activity	Student Activity	Methodology	Media Required	Time
Start up and turn off the computer	Listening, Discussion, Observation, and Practice	Writing test and practical test.	Multimedia, Instructor Computer, and Computers for each student	1hr
Customize computer and desktop settings	Listening, Discussion, Observation, and Practice	Writing test and practical test.	Multimedia, Instructor Computer, and Computers for each student	5hrs
Create folders and files	Listening, Discussion, Observation, and Practice	Writing test and practical test.	Multimedia, Instructor Computer, and Computers for each student	1hr
Perform Folder/File operations	Listening, Discussion, Observation, and Practice	Writing test and practical test.	Multimedia, Instructor Computer, and Computers for each student	2hrs
Set attributes of files folders	Listening, Discussion, Observation, and Practice	Practical test.	Multimedia, Instructor Computer, and Computers for each student	2hr
Compress and extract folder/ files	Listening, Discussion, Observation, and Practice	Practical test.	Multimedia, Instructor Computer, and Computers for each student	3hrs
Backup and restore folders/ files	Listening, Discussion, Observation, and Practice	Practical test.	Multimedia, Instructor Computer, and Computers for each student	5hrs

Figure B - 5 : Lesson Plan

Year Plan

The instructor should develop this plan at the start of the semester. They schedule the time it will take to teach each topic in accordance with the curriculum . A curriculum defines the fundamental knowledge that the student should acquire during the course. It defines the subject, the modules within the subject, and the module units. It also explains how much time is required to teach. As a result, when preparing the course, the instructor should evaluate the subject, module, unit, and time required for it (Figure B - 6 : Year Plan report.).

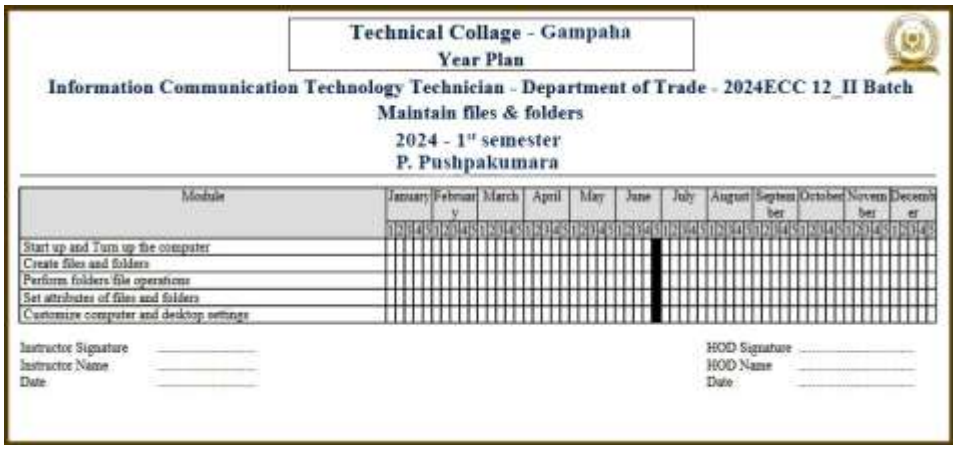


Figure B - 6 : Year Plan report.

APPENDIX C - USER MANUAL

This comprehensive guide will help you navigate and utilize the various features and functionalities of the system efficiently. The system encompasses a range of capabilities, including face recognition attendance, tracking of students' attendance, managing instructors' lesson plans, term notes, activity plan creation, online course payment, and leave reports for both students and staff. The application's user documentation will assist all first-time users in identifying the steps they must take.

SIGN IN

Step 1 & 2: Sign in to the system by using valid user name and password. Figure C - 1 :Sign inRefers to the steps the sign in process (Figure C - 1 :Sign in)

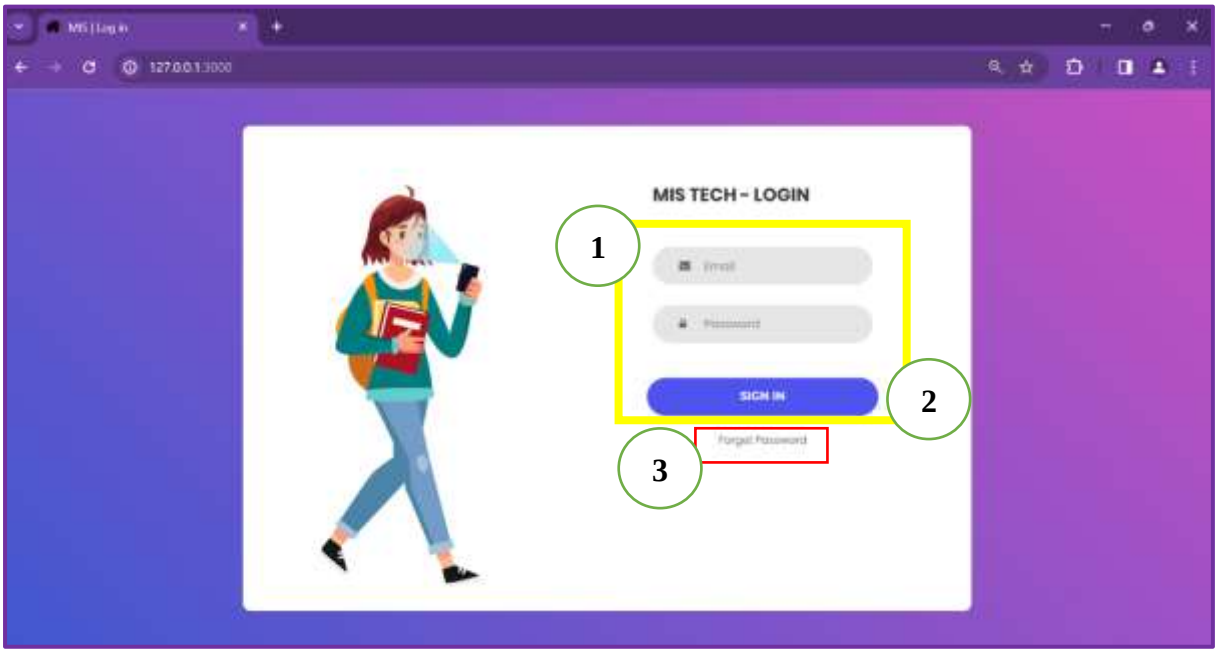


Figure C - 1 :Sign in

Step 3: If user forgot the password Click on the forgot password link below.

Step 4: Enter an email address and submit it.

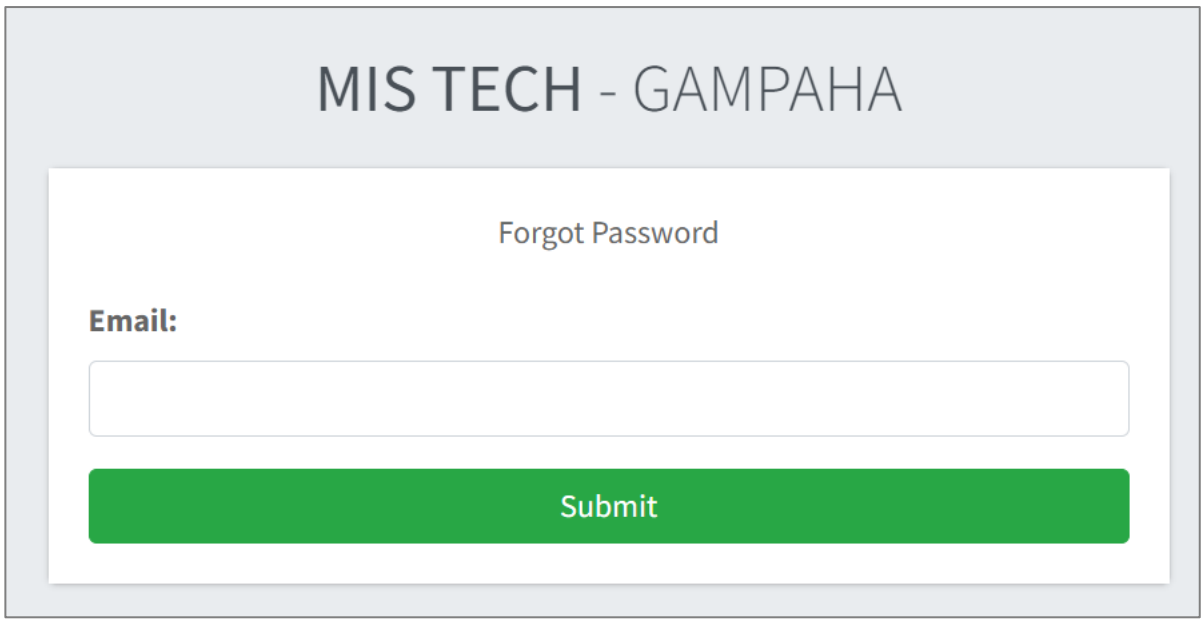


Figure C - 2 : Forgot Password

Step 5: You will receive the message below

Figure C - 3 : Message

Step 6: Click on the received link or copy and paste on the browser URL.

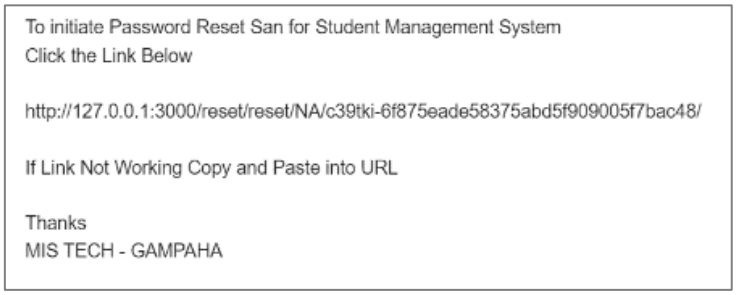


Figure C - 4 : Email Message

Step 7 : Using new password try to sign in again if the credentials are valid, the users will be greeted with a dashboard displaying relevant information and navigation options.

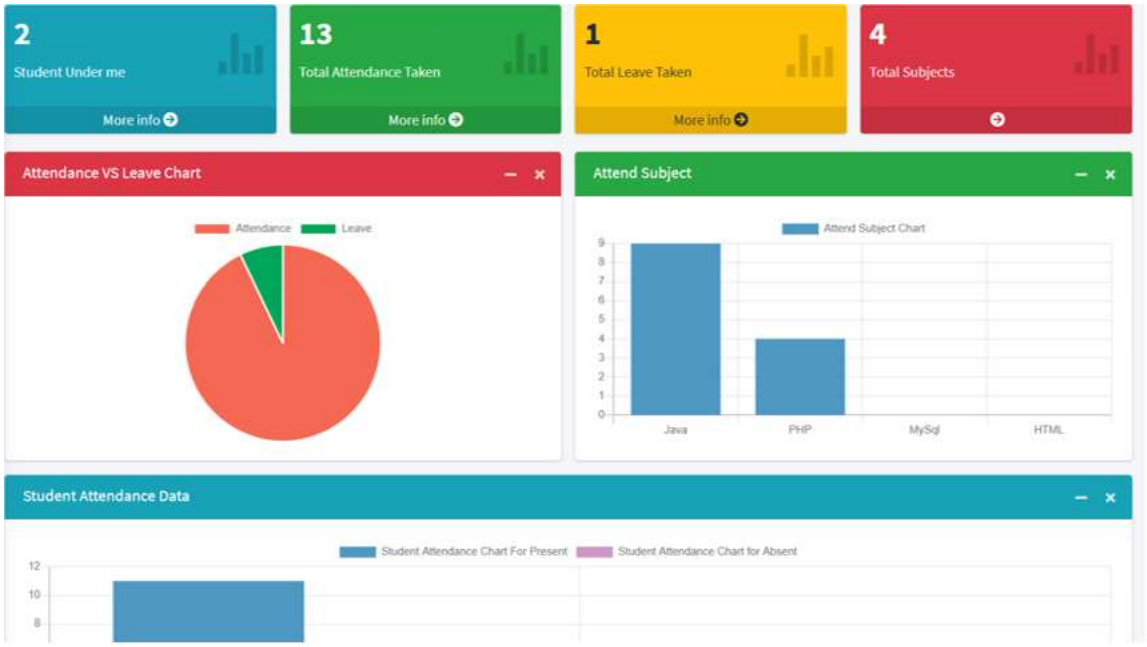


Figure C - 5 : Dashboard

Step 8: After signing in, you will be able to edit your profile.

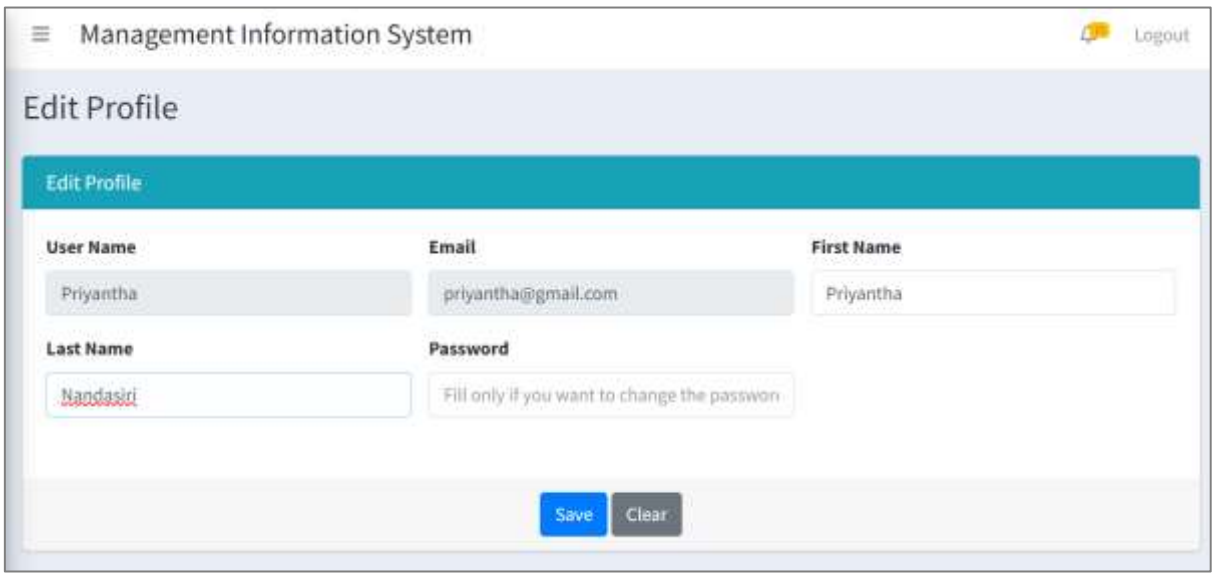


Figure C - 6 : Edit Profile

Administrator: Manage Information

Step 1: Select the slide bar, click on the menu which need to manage.

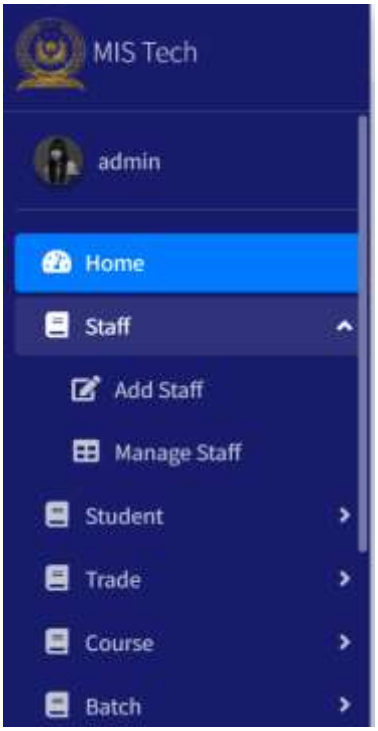


Figure C - 7 : Slide bar

Step 2: Click on Add Staff submenu

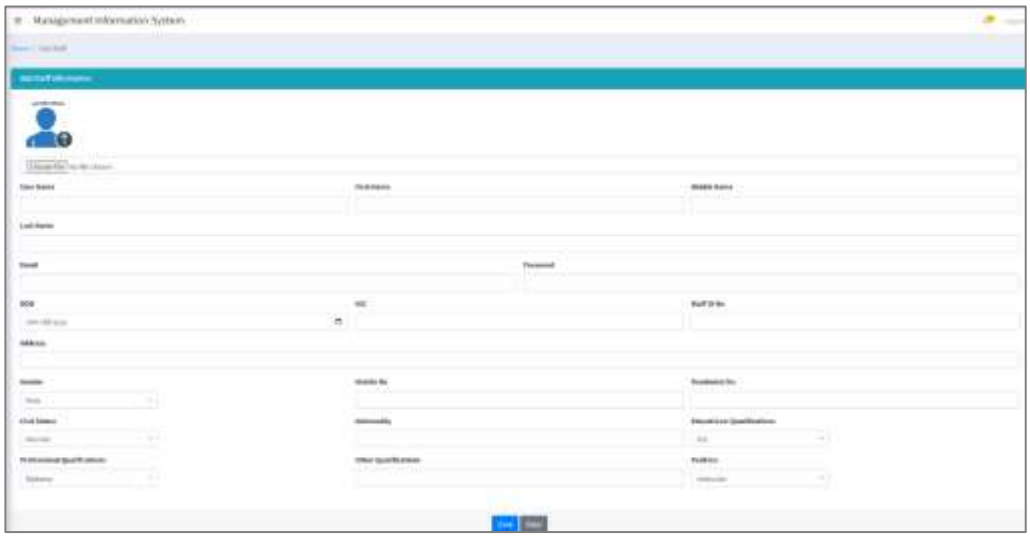


Figure C - 8 :Add Staff

Step 3: View Staff Data

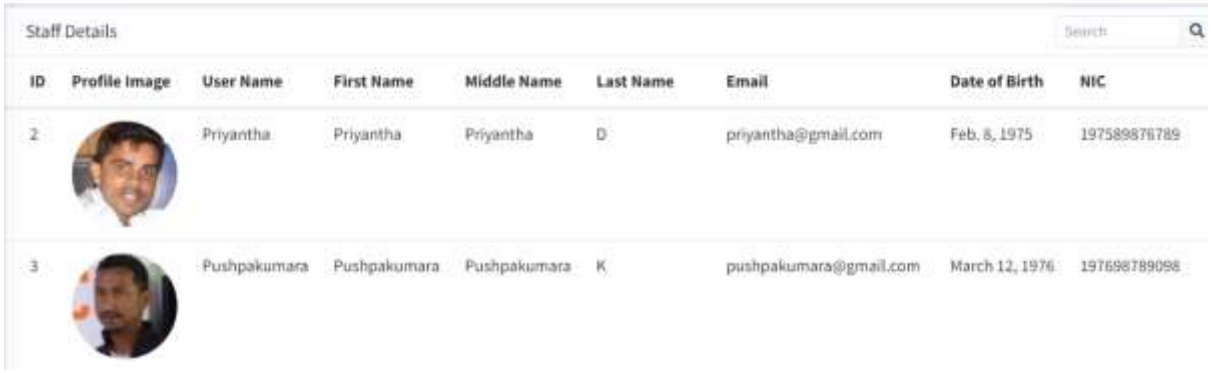


Figure C - 9 : Staff Details

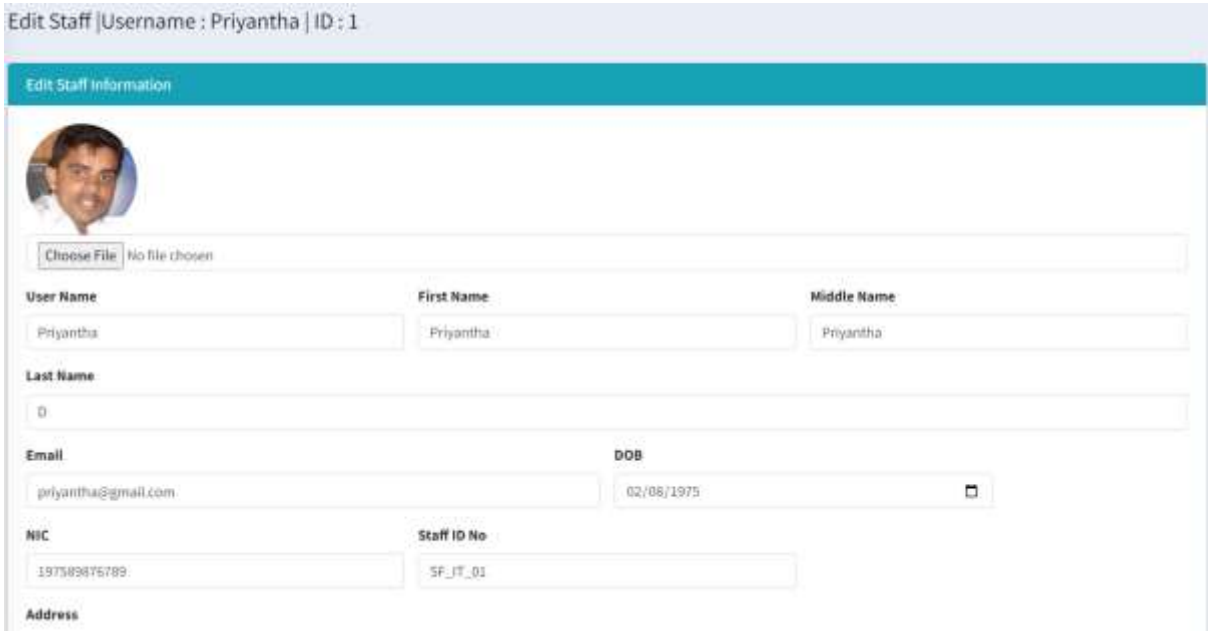


Figure C - 10 : Edit Staff

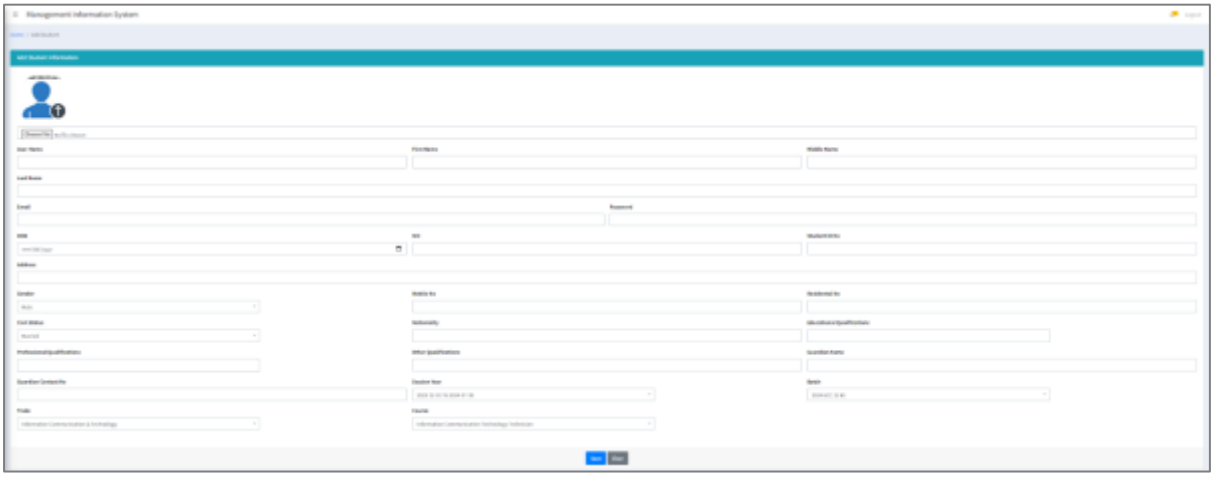


Figure C - 11 : Add Student

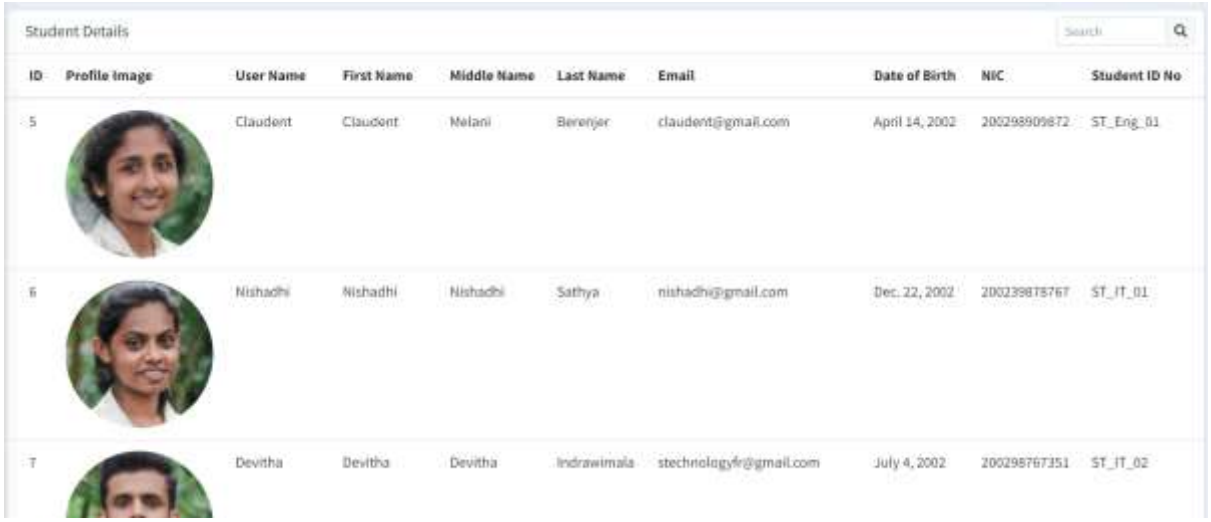


Figure C - 12 : Student Details

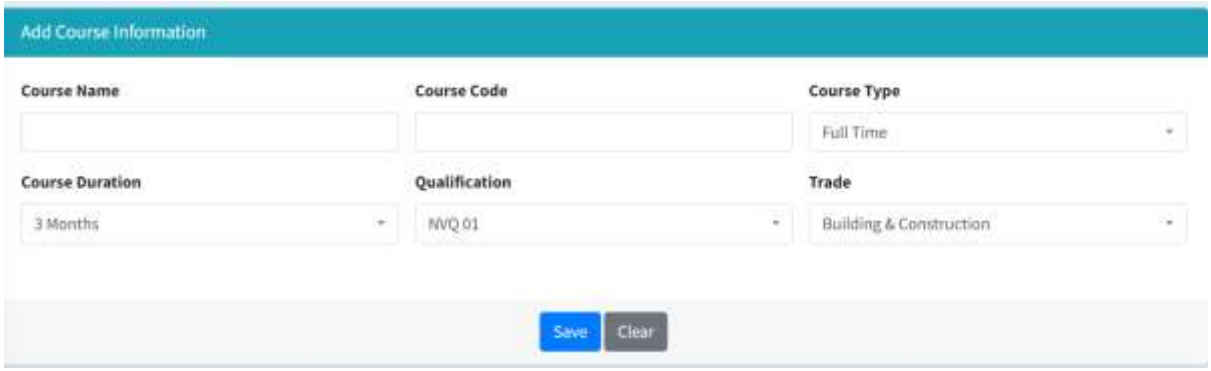


Figure C - 13 : Add Course Details

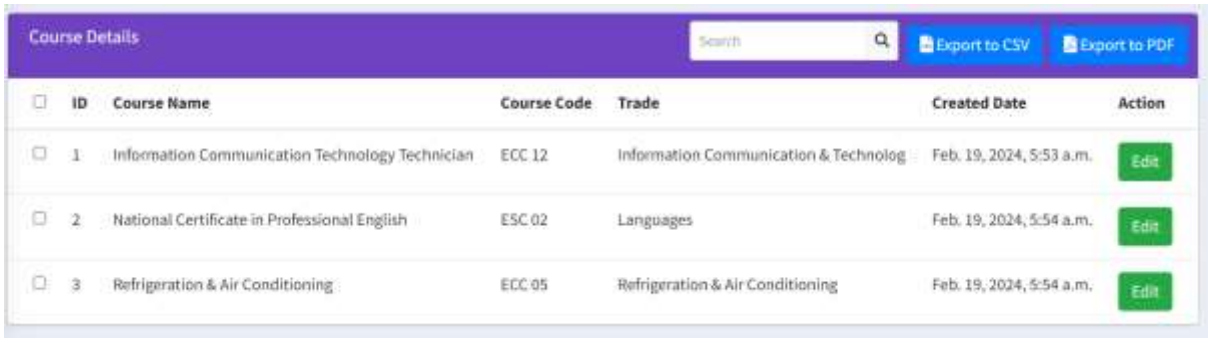


Figure C - 14 : View Course Details

Instructor : Take attendace

Step 1: Sing In to the system

Step 2: Click on FR Take Attendance menu

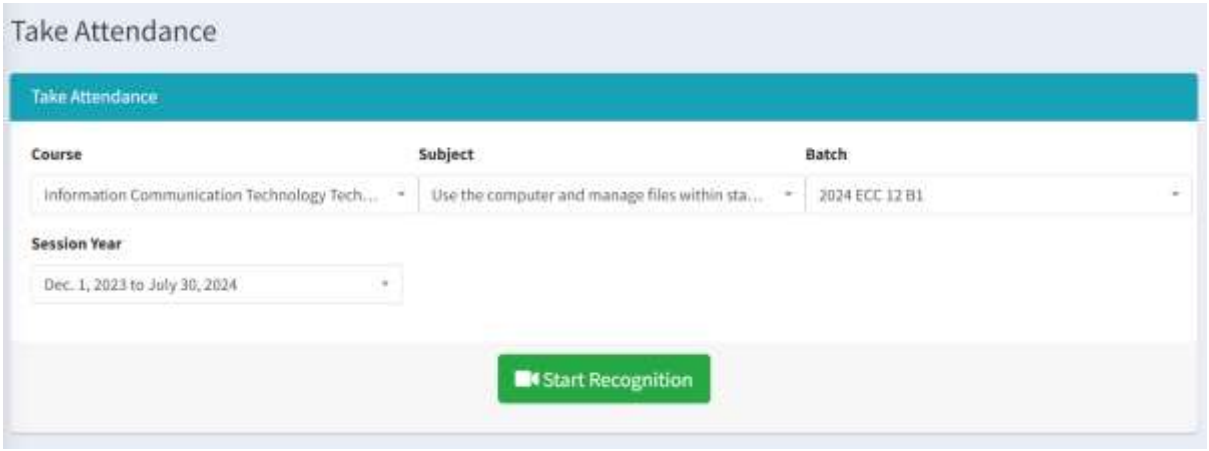


Figure C - 15 :Take Attendance

Step 3: View Attendance

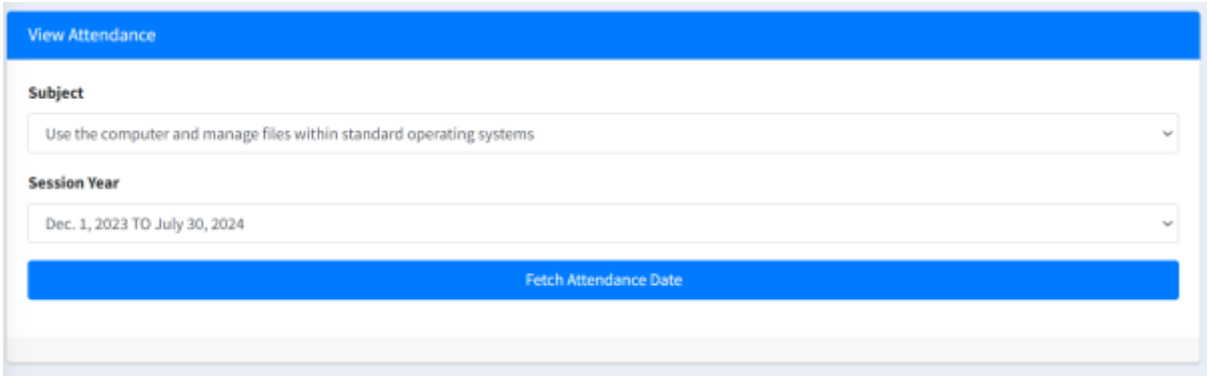


Figure C - 16 :View Attendance

Examiner : Take Attendace

Step 1: Select the FR Take Attendance

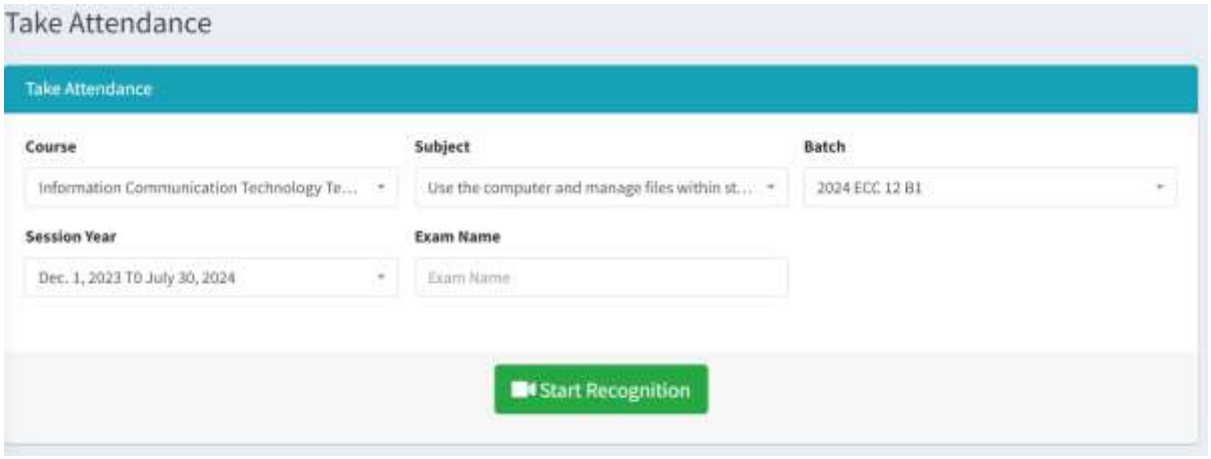


Figure C - 17 : Exam Take Attendance

Step 2: View Attendance

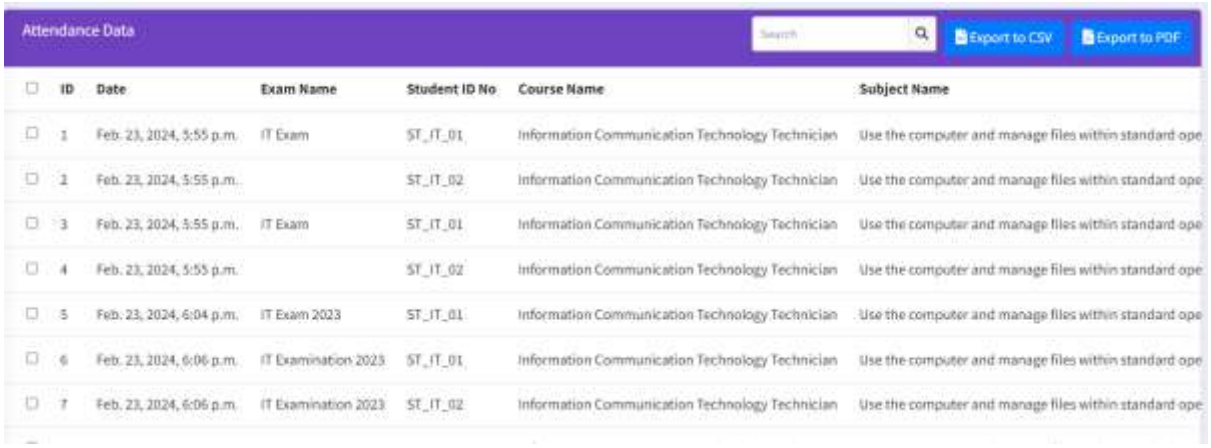


Figure C - 18 :View Attendance

Student: Online Course Payment

Step 1: Login to the system

Step 2: Select course payment menu in the slide bar

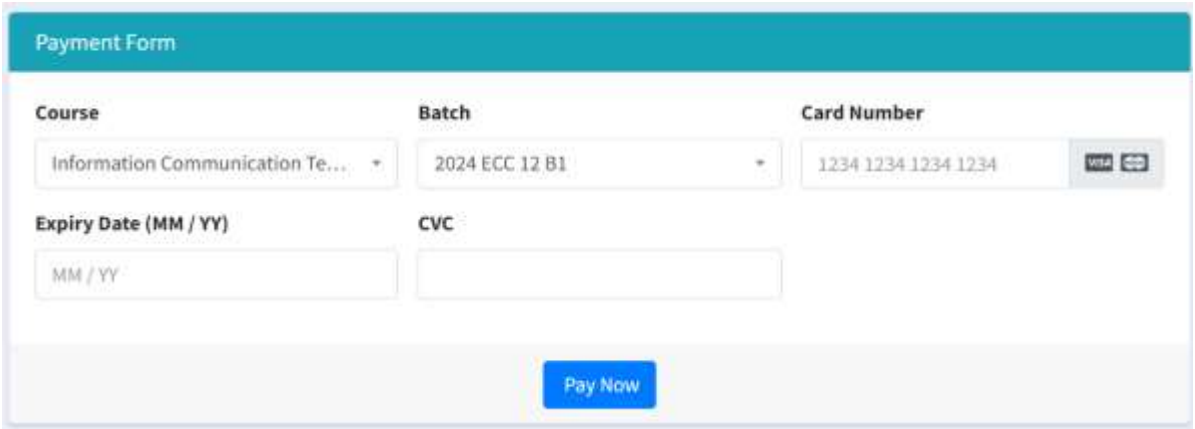


Figure C - 19 : Course payment

Instructor: Apply Leave

Step 1: Login to the system

Step 2: Select the Apply Leave menu

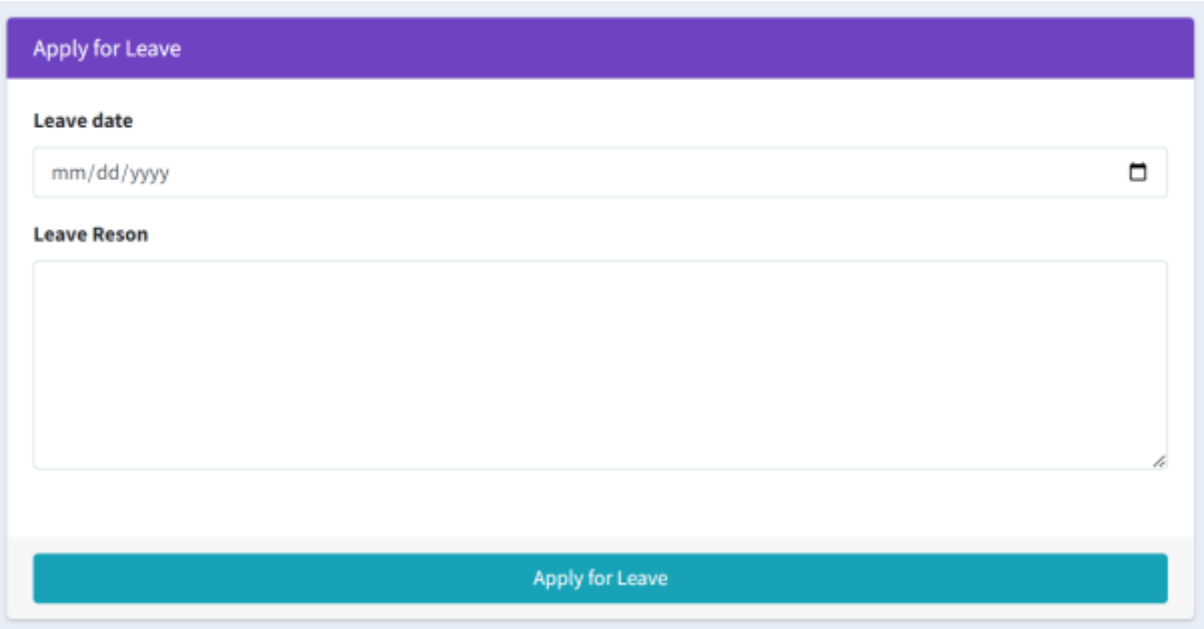


Figure C - 20 : Apply for Leave

Step 3: View leave apply history

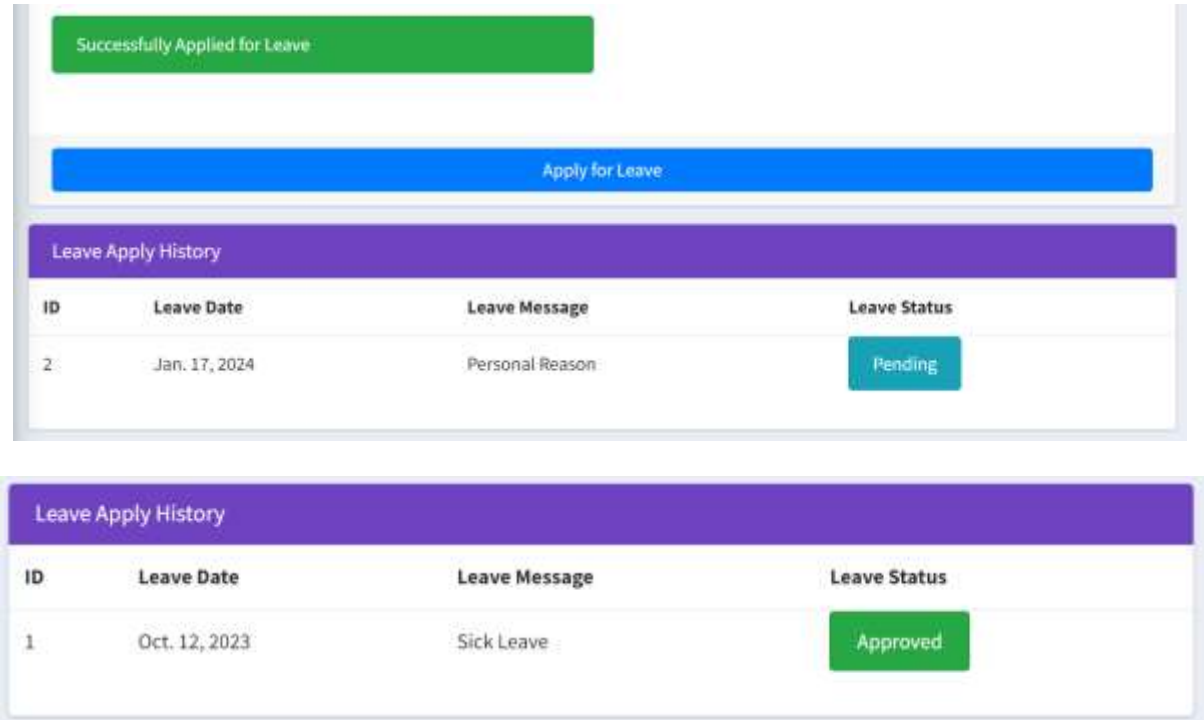


Figure C - 21 : Leave apply history

Step 4: Select feedback menu. Send feedback message

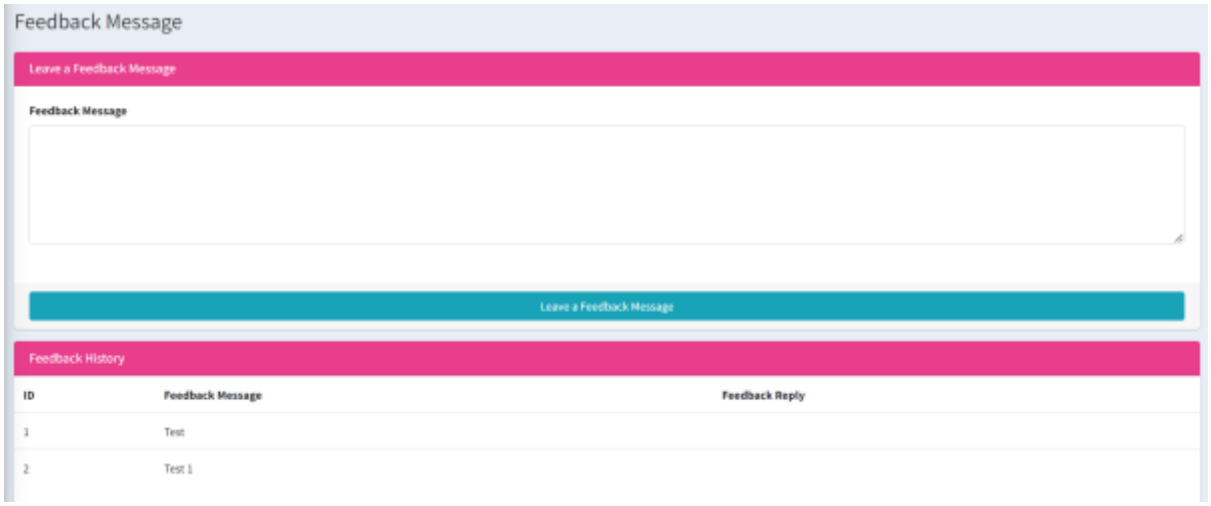


Figure C - 22 : Feedback

Principal – Approve or Disapprove leaves

Step 1: Login to the system

Step 2: select Staff Leave menu

Figure C - 23 : Leave Approve or Disapprove form

Figure C - 24 : Staff leave Approve or Disapprove form

APPENDIX E – INSTITUTE APPROVAL LETTER

Ministry of Education

Department of Technical Education & Training

To whom It May Concern

I am writing to confirm the project work completed by Ms. K.H.S. Thilakarathne for the Gampaha Technical College.

The MIS system was handed over to the college staff and tested. They demonstrated that their approach is extremely efficient and effective. We look forward to future collaboration with the Department of Technical Education and Training, as well as liaising with the Department's information systems.

This system will be hosted in the Gampaha Technical College's hosting environment, and Ms. K.H.S. Thilakarathne has been requested to keep all MIS system data and information strictly confidential when submitting it to the University of Colombo School of Computing.

Thank You

Implementation of ICT section

B.D.P. Nandasiri

Technical College

Gampaha

2024.03.06

B.D.P. Nandasiri
H.O.D. Commerce Section
Gampaha Technical College