KATHMANDU UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
DHULIKHEL, KAVRE

A PROJECT REPORT
ON
"FACULTY MANAGEMENT SYSTEM"
[ENGG:102]

FOR THE PARTIAL FULFILMENT OF I/II IN COMPUTER ENGINEERING

SUBMITTED BY
Aarogya Ghimire
Basanta Khanal
Ghanshyam Khatri
Shasank Singh Thakuri
Shaswat Poudel

SUBMITTED TO
SUMAN SHRESTHA
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SUBMISSION DATE
9th March 2025

BONAFIDE CERTIFICATE

This project Work on

“Faculty Management System”

is the bona fide work of

Aarogya Ghimire

Basanta Khanal

Ghanashyam khatri

Shasank Singh Thakuri

Shaswat Poudel

who carried out the project work under my supervision.

Project Supervisor

____

Mr. Trailokya Ojha

Assistant Professor

Department of Computer Science and Engineering

Date: March 03, 2025

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to all those who have contributed to the successful completion of this project. First and foremost, we would like to thank our project supervisor, Asst. Prof. Mr. Trailokya Ojha, for his invaluable guidance, support, and constructive feedback throughout the course of this project. His expertise and encouragement have played a significant role in shaping the direction of our work.

We would also like to extend our appreciation to our team members for their dedication, cooperation, and collaborative spirit in completing this project. The collective effort, hard work, and commitment of each member made this project successful.

Additionally, we are grateful to Kathmandu University Department of Computer Science and Engineering for giving us the opportunity to execute the plan into actual program. Thank you all for your contribution and support.

ABSTRACT

"Faculty Management System"

Faculty management system aims to be a reliable source of messaging and information sharing applications for students of not only Kathmandu University but any other university or campus. The application will allow students to see their attendance and get informed about their class time and status. If the lecturer is cancelled then students can see updated routine which will be helpful for the teachers who wants to change, reschedule or add the class

Keywords: Qt, SQL lite, Faculty Management System.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS/ABBREVIATIONS

FMS=Faculty Management System

CHAPTER I. Introduction

FMS is specialized software applied in educational institutions to manage all faculty-related responsibilities. It efficiently offers a way to allocate faculties, data management, and communication. Automating these will reduce the administrative burden, increase transparency, provide secure data, bring better decision-making. Our FMS basically focuses on keeping records of Student Data, Marking Attendance, assigning teacher in respective class, along with managing routine.

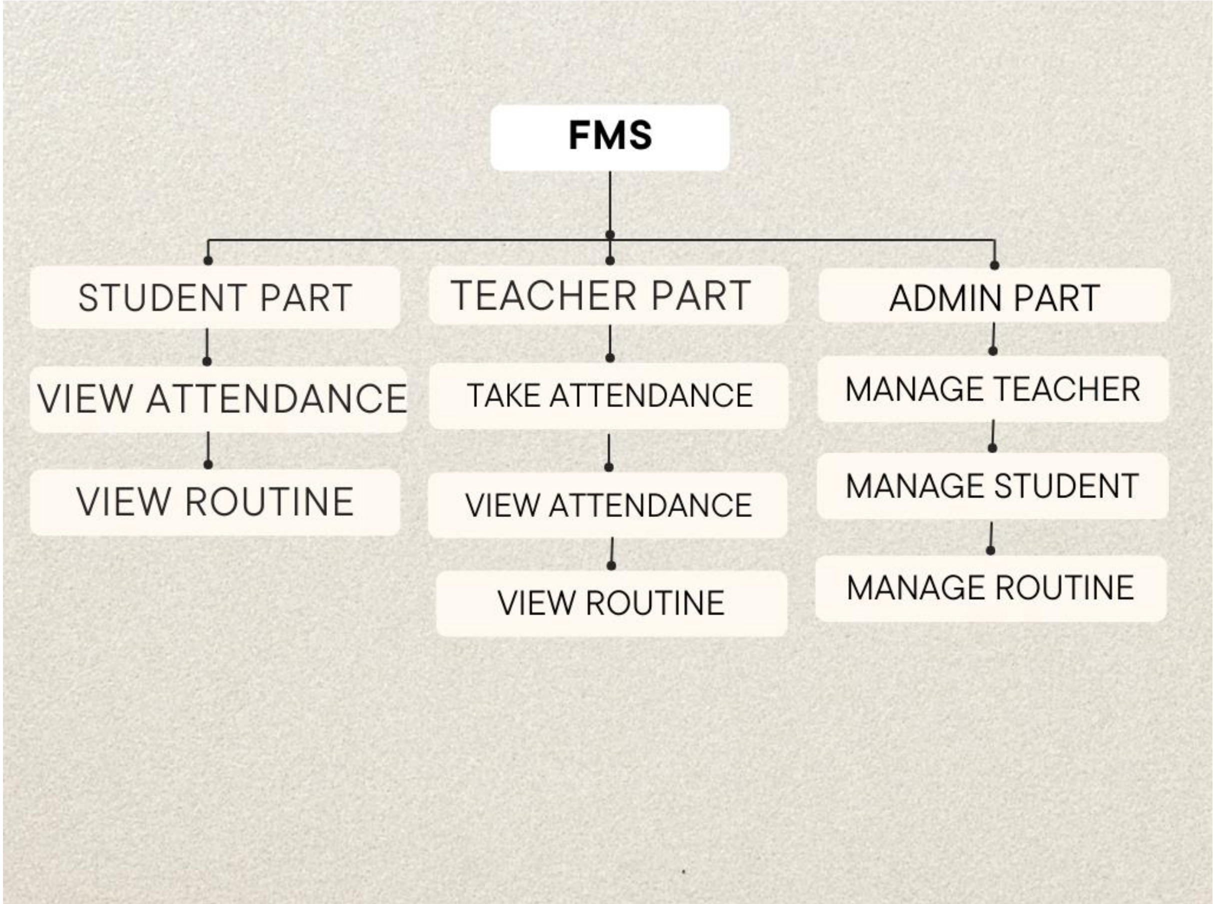


FIGURE 1. 1 REPRESENTATION OF FACULTY MANAGEMENT SYSTEM

1.1. Background

In the 21st century education is growing dynamically; the role of the faculty management system has become more crucial. They offer learning management systems or faculty information systems.

FMS evolved with the advancement in technology as well as evolving needs of academies. Old academies dealt with paper-based records for information of faculty members, attendance, routine, Faculty schedules, workload distribution and evaluations were being managed manually via ledgers, registers, and printed reports. Thereafter by birth and development of internet and system led to the development of centralized faculty management system.

Our project is framed to provide the facilities for the faculty by ensuring the educational institutions such as data security, data and logging data, keeping the routine and updating it. It also helps us to store and manage data in a very efficient way which helps us to ensure that all information is transferred efficiently to everyone who uses it.

Proper management of teacher and student records is the aim of FMS. Students can view their attendance at the same time, and it also helps teachers keep track of student attendance. It also greatly aids in routine management and provides teachers and students with updates

1.2. Objective

- To manage faculty and student records efficiently
- To be able to Track attendance
- To be able to a Manage routine

1.3. Motivation And Significance

In the current system, communication between students, teachers, and administrators primarily relies on a Class Representative (CR). The admin informs the CR about schedules, and students depend on the CR for routine updates. Similarly, teachers notify the CR about class cancellations or changes, which can lead to delays, miscommunication, or information loss. This indirect communication method creates inefficiencies and dependency on a single individual, potentially affecting academic coordination.

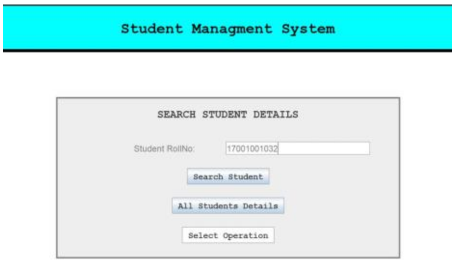
To address these challenges, we propose an integrated system where students, teachers, and administrators can interact directly. By eliminating unnecessary intermediaries, this system will enhance transparency, reduce delays, and ensure that all stakeholders receive real-time updates regarding class schedules, cancellations and others.

CHAPTER II: Related Works/Existing Works

There are similar applications related to our work. Some of them are described as follows:

2.1. Student Management System

A Student Management System is a software solution designed to help educational institutions better manage student data, academic records, administrative operations, and communications. It automates various processes such as admissions, attendance, grades, fee management, and reporting.



Student Managment System

SEARCH STUDENT DETAILS

Student RollNo: 17001001032

FIGURE 2. 1 STUDENT MANAGEMENT SYSTEM

2.2. Library Management System

A Library Management System is a program that automates and regulates library processes such as cataloguing books, monitoring returned and borrowed books, managing member information, and generating reports. It enhances libraries' efficiency through reduction of manual interventions, enhanced accuracy of data, and optimum satisfaction of users.

FIGURE 2. 2 LIBRARY MANAGEMENT SYSTEM

CHAPTER III: Design And Implementation

3.1. Design Pattern

The design and implementation phase of the Faculty Management System (FMS) involved creating a fast, user-friendly, and scalable application using QT Creator in C++. The system was designed in three main user roles: Admin, Teacher, and Student. Each role has specific features, and the system was developed collaboratively by a team of five members using Git for version control and collaboration. Here’s the brief visual representation of system:

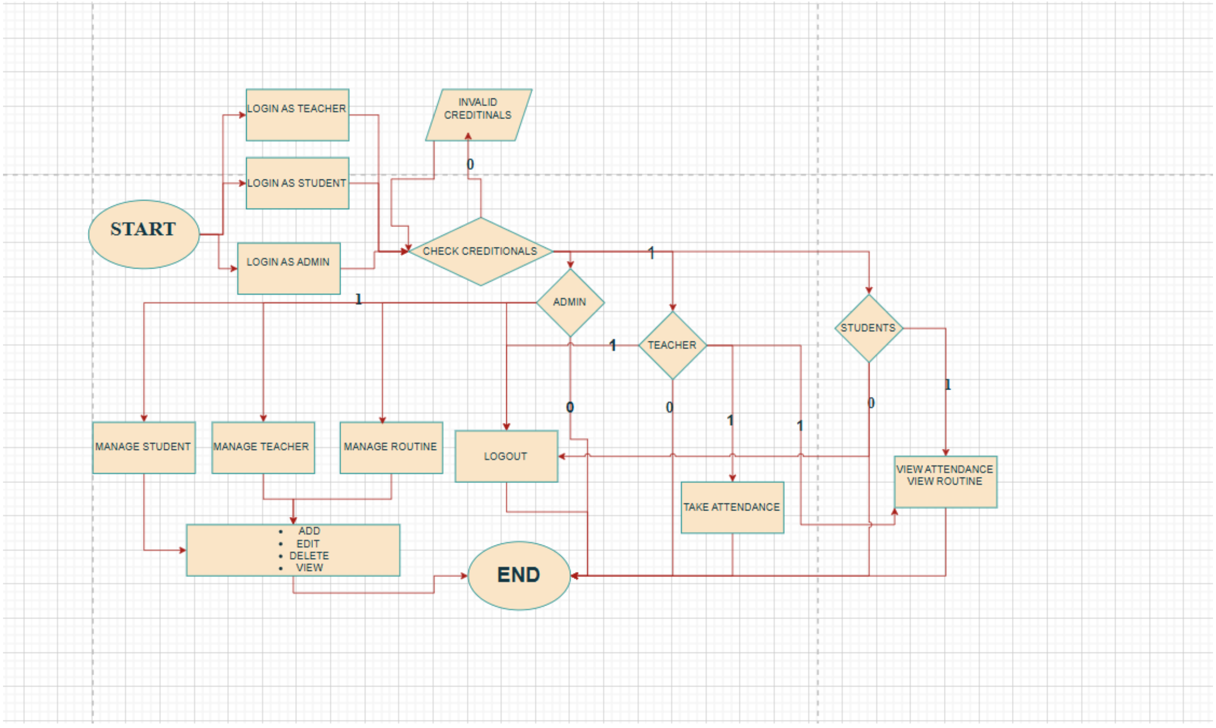


FIGURE 3. 1 FLOW CHART OF FACULTY MANAGEMENT SYSTEM

3.1.1. System Design

We took a modular approach to design the system, breaking it down into components based on user roles and their specific functionalities. Here’s how we approached the design phase:

3.1.1.1. Use Case Diagram

- **Admin**: Responsible for managing routines, adding or removing teachers and students.
- **Teacher**: Handles tasks like taking attendance, scheduling or cancelling classes (only for subjects they are allocated).
- **Student**: Can view their attendance.

The following is the visual representation of User -Case:

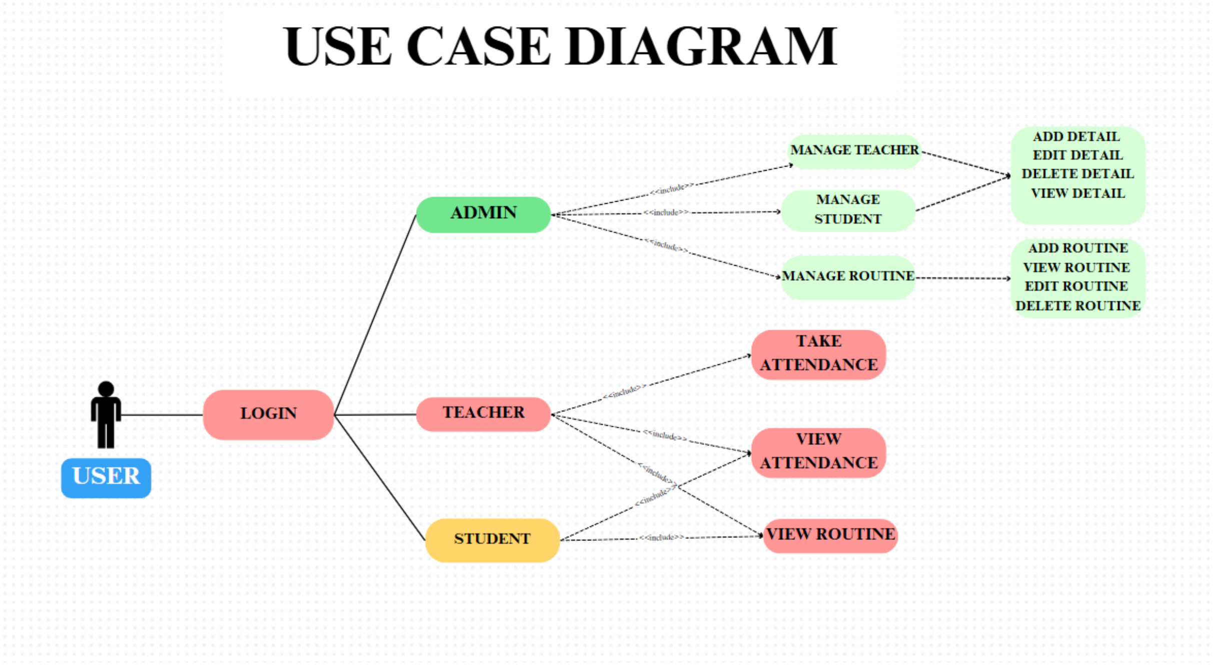


FIGURE 3. 2 USER-CASE DIAGRAM

3.1.2. Database Design

We chose SQLite as our database system. The database included several tables**:**

- **Users**: Stores login credentials and user roles.
- **Students**: Contains student details like name, ID, attendance, and marks.
- **Teachers**: Stores teacher details and the subjects they teach.
- **Classes**: Tracks class schedules and attendance records.
- **Routine**: Manages the timetable for each class.

The following shows the relation between entities:

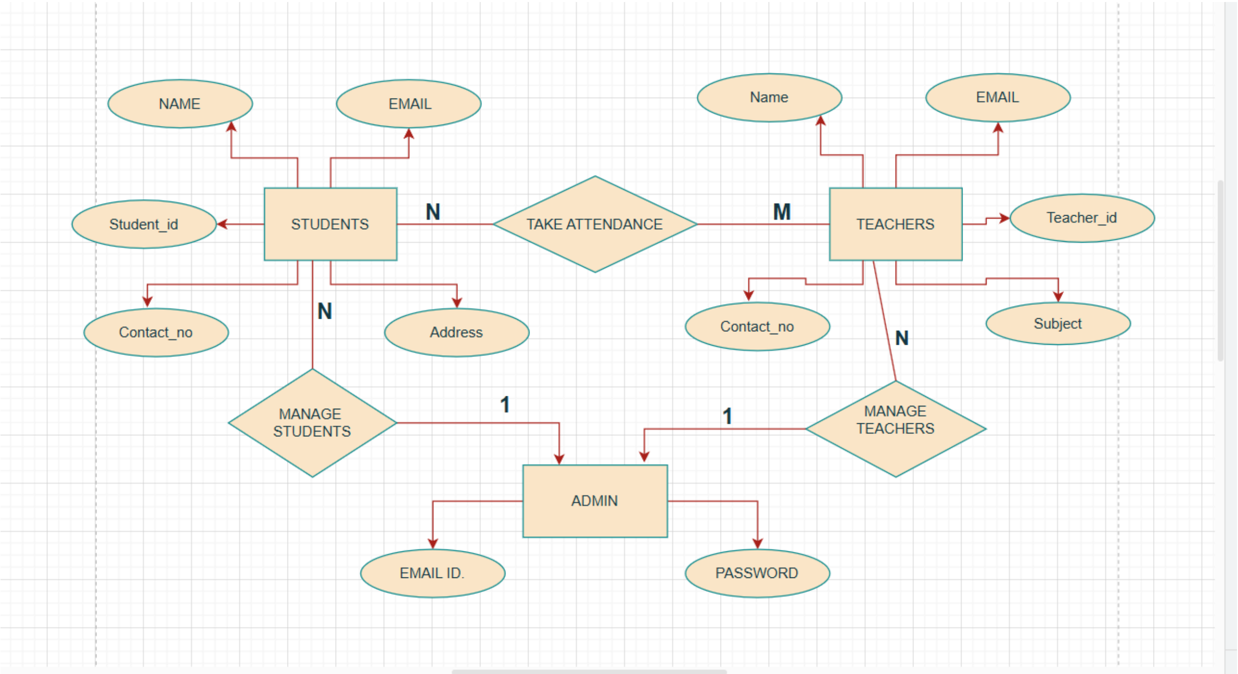


FIGURE 3. 3 ENTITY RELATIONSHIP DIAGRAM

3.2. System Requirement Specification

The software and hardware requirement are as follows:

3.2.1. Software requirement

The software requirement includes short description of the software that are required for the projects.

3.2.1.1. Front-End and Back-End Tools

- Qt C++

Qt is a cross-platform application framework. Cross platform application, which means you write the application once and then it works on different system. Qt can also be understood as a cross-platform application development framework with a strong emphasis on a graphical user interface.

3.2.1.2. Database

- SQ LITE

SQ Lite is a highly efficient, serverless and self-contained SQL database engine that stands out for its simplicity and ease of integration. Designed to be embedded with applications, SQ Lite eliminates the need for separate database server processes and complex configurations.

3.2.2. Hardware tools

Any desktop or laptop devices can run the application mentioned above.

CHAPTER IV: Discussion On Achievements

During the development of the Faculty Management System (FMS), we successfully implemented core functionalities, allowing admins to manage student and teacher records, assign classes, and schedule routines, while teachers can take attendance and view routines, and students can check their attendance and routine updates. However, we encountered several challenges, including coding errors like clazy name errors and SQL driver not found, with the most significant issue being parameter count mismatches in SQL queries. Additionally, Git-related issues, such as the .pro file being automatically removed when collaborators pushed changes, previously commented code becoming uncommented after pushes, and deleted slots not being permanently removed, caused significant difficulties in version control. API integration complexities prevented us from adding a notification system, requiring students and teachers to manually check for routine updates. Teachers also lack an in-system communication feature to notify students about changes. Furthermore, the system currently supports only a single class without multi-department functionality. Despite these limitations, the project successfully provides a structured and efficient faculty management system, laying the foundation for future enhancements such as multi-department support, internal communication features, and real-time notifications.

4.1. Features

4.1.1. User Role-Based Access

The Faculty Management System (FMS) implements a user role-based access system to ensure that each user has specific permissions based on their role. Admins have the highest level of access, allowing them to add, edit, delete, and view student and teacher details, as well as manage class routines. Teachers have limited access, enabling them to take attendance and view their assigned routines. Students have the most restricted access, allowing them only to view their attendance records and class routines. This structured approach ensures a clear hierarchy, prevents unauthorized modifications, and maintains data integrity within the system.

4.1.2. Separate Credential Verification

The Faculty Management System (FMS) employs separate credential verification for Admin, Teacher, and Student users to enhance security and ensure role-based access control. Each user type has a dedicated authentication database, preventing unauthorized access to restricted functionalities. Admins, teachers, and students log in using their respective credentials, which are stored separately to maintain data integrity and streamline user management. This approach ensures that each role has access only to the features relevant to them, minimizing security risks and improving system efficiency.

4.1.3. Uncrowded Database Structure

The Faculty Management System (FMS) features an uncrowded database structure by storing different types of data in separate databases. Instead of a single, overloaded database, credentials, student details, teacher records, attendance, and routines are managed independently. This approach improves data organization, enhances performance, and ensures efficient retrieval and management of information while reducing redundancy and complexity.

4.1.4. Routine & Attendance Management

The Faculty Management System (FMS) maintains routine and attendance data in separate databases to ensure efficient management and retrieval. The routine database stores class schedules, allowing admins to modify and assign teachers accordingly, while teachers and students can view their respective routines. The attendance database records student attendance, which teachers can update, and students can access for tracking their records. By keeping these datasets separate, the system avoids data congestion, enhances performance, and simplifies data handling.

CHAPTER V: Conclusion And Recommendation

5.1. Limitations

Our Faculty Management System (FMS) offers many useful features, but it has several limitations. The limitations are highlighted below:

5.1.1. Lack of Notification System

One of the major limitations of the FMS is the absence of a notification system. Due to challenges with API integration, the system cannot automatically notify students and teachers about changes in their routines or attendance. As a result, students and teachers must manually check the system to stay updated on any modifications made by the admin. This lack of real-time alerts can lead to delays in information sharing, affecting the overall user experience.

5.1.2. No In-System Communication

The system also lacks an integrated communication feature, meaning teachers cannot directly inform students through the platform about any changes or important updates. Teachers must resort to external means of communication, such as emails or messaging apps, to notify students, which can result in missed messages or delayed communication. The absence of this feature reduces the efficiency of the system.

5.1.3. Limited Scope (Single-Class Focus)

Currently, the FMS is designed to manage data for only a single class. It does not support multi-department or multi-class functionality, which limits its scalability and applicability in larger educational institutions. As a result, the system cannot be easily adapted for schools or colleges with more complex structures, where multiple departments or classes would need to be managed simultaneously.

5.2. Future Enhancements

For future enhancements of the Faculty Management System (FMS), several features can be integrated to address its current limitations. The following features can be added:

5.2.1. Notification System Integration

A key enhancement for the future is the integration of a real-time notification system. This would allow automatic alerts to be sent to students and teachers whenever there are changes to routines or attendance. Implementing push notifications or email alerts will improve communication and ensure users are immediately informed about updates, reducing the need for manual checks and increasing overall efficiency.

5.2.2. In-System Communication Feature

Introducing an in-system communication feature would enable teachers to directly communicate with students within the platform. This could include announcements, feedback on assignments, or reminders about upcoming classes. It would streamline communication, keeping all important academic information centralized within the system and reducing reliance on external messaging platforms.

5.2.3. Multi-Class and Multi-Department Support

Currently focused on a single class, the FMS could be expanded to support multiple classes and departments. This would make the system scalable and suitable for larger institutions. By allowing admins to manage different departments, teachers to be assigned across multiple classes, and students to access routines and attendance from different subjects, the system would cater to more complex educational environments.

5.2.4. Enhanced Reporting and Analytics

Future versions of the system could include enhanced reporting features, such as data analytics on student performance, attendance trends, and teacher efficiency. Admins could generate reports to monitor student progress, identify patterns in attendance, and assess teacher workloads. This would help in making informed decisions to improve both student outcomes and the overall functioning of the institution.

5.2.5. Mobile App Integration

To improve accessibility, a mobile application version of the FMS could be developed. This would allow students, teachers, and admins to access their data and stay updated on routines, attendance, and other information directly from their smartphones, enhancing usability and convenience for all users.

By implementing these future enhancements, the FMS could provide a more comprehensive, user-friendly, and scalable solution to manage faculty-related tasks efficiently.

5.3. Conclusion

We acquired important knowledge and abilities during the Faculty Management System project's development, which greatly enhanced our capacities. In order to make sure our application can manage expansion and rising user demand; we learned how to create a scalable system. We expanded our technical knowledge by investigating and utilizing industry-standard technologies like SQ Lite and Qt C++. We improved our knowledge of database joins and relationships, which helped us handle and retrieve data more effectively. Our ability to collaborate was enhanced by teamwork, emphasizing the value of coordination and communication. In order to improve user engagement and learning, we also investigated the gamification of computer systems. These experiences have given us a broad range of skills that will help us tackle future software development and teamwork challenges.

5.4. Recommendation

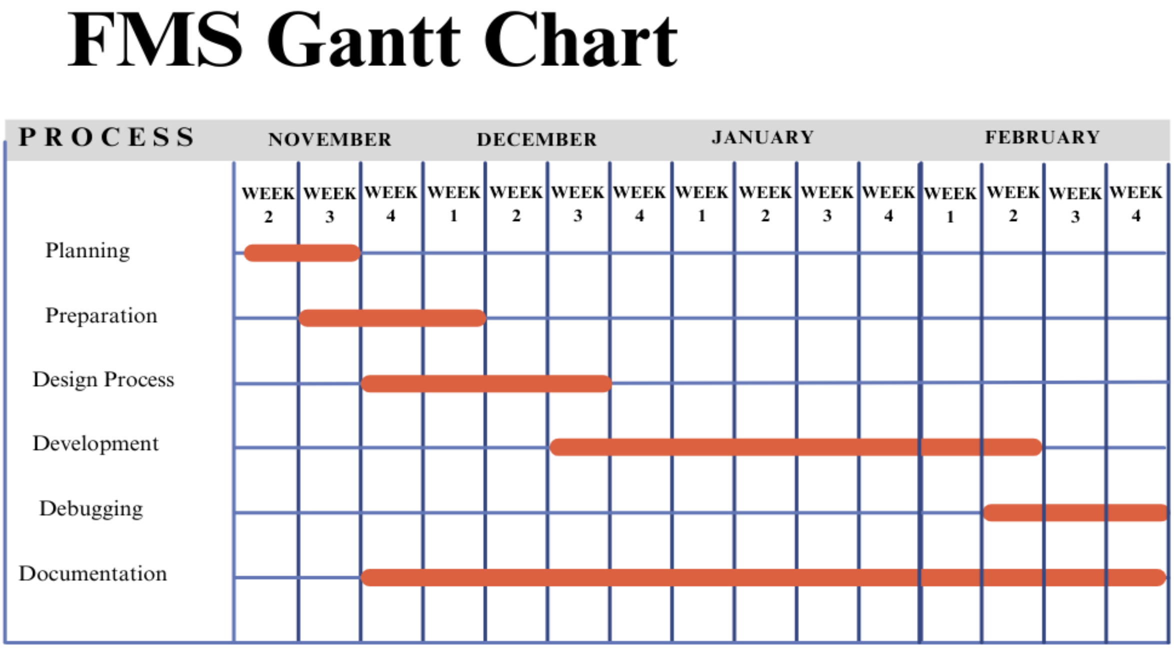
We strongly advise anyone interested in our project, the "Faculty Management System," to invest a substantial amount of time in learning about the Qt C++ and SQ Lite database. For concepts and features to be implemented successfully, a thorough understanding of these technologies is essential. All team members who have a firm understanding of these tools will encourage creative thinking and help the project succeed overall.

References

Sourabh Sharma, S. M. (2022, 5 8). *Library Management System*. Retrieved from iJRASET: https://www.ijraset.com/research-paper/library-management-system

Staff, T. (2021, 7 1). *A Guide To Student Management Systen* . Retrieved from TUIOPAY: https://tuiopay.com/blog/what-is-a-student-management-system/

Appendix I: Gantt Chart



FMS Gantt Chart

PROCESS	NOVEMBER			DECEMBER				JANUARY				FEBRUARY			
	WEEK 2	WEEK 3	WEEK 4	WEEK 1	WEEK 2	WEEK 3	WEEK 4	WEEK 1	WEEK 2	WEEK 3	WEEK 4	WEEK 1	WEEK 2	WEEK 3	WEEK 4
Planning	■	■													
Preparation		■	■	■											
Design Process			■	■	■	■									
Development						■	■	■	■	■	■	■	■		
Debugging													■	■	■
Documentation			■	■	■	■	■	■	■	■	■	■	■	■	■

TABLE 1. 1 GANTT CHART

Appendix II: Screenshots

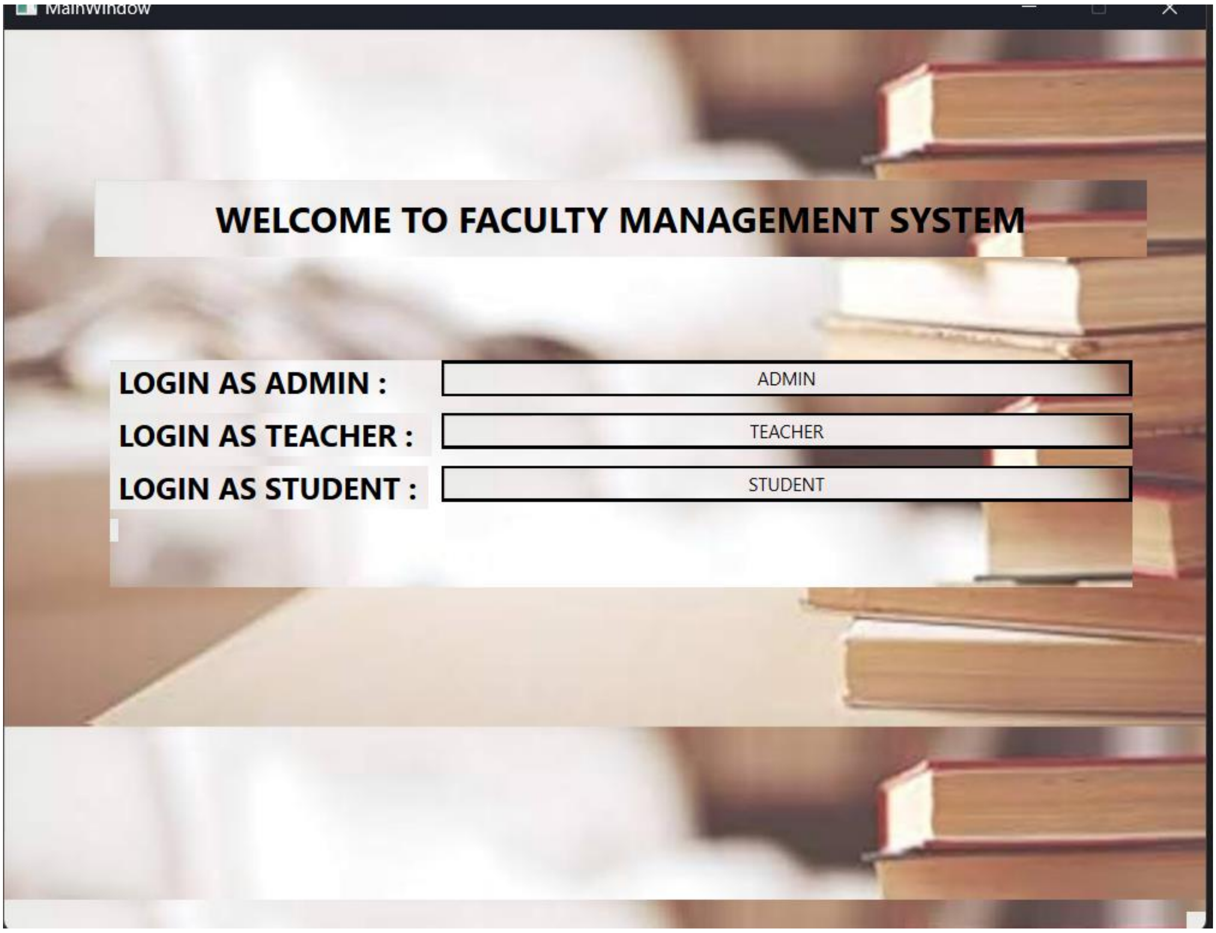


FIGURE 4. 1 MAIN PAGE

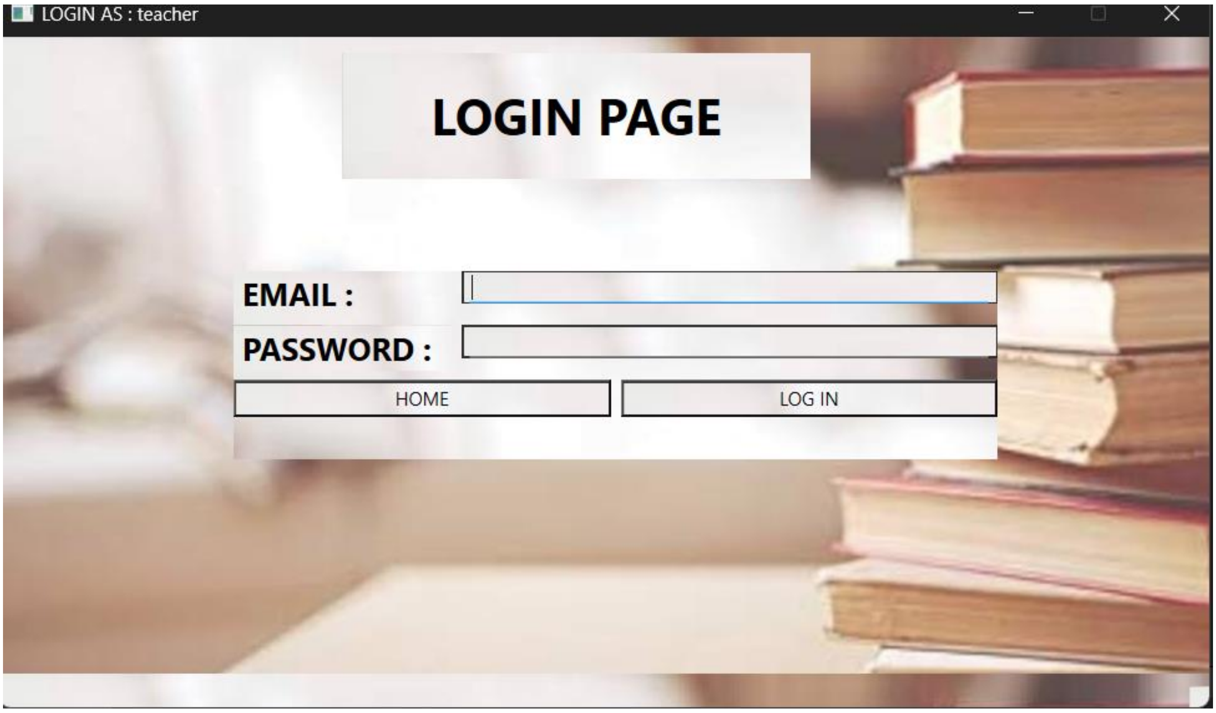


FIGURE 4. 2 LOGIN PAGE

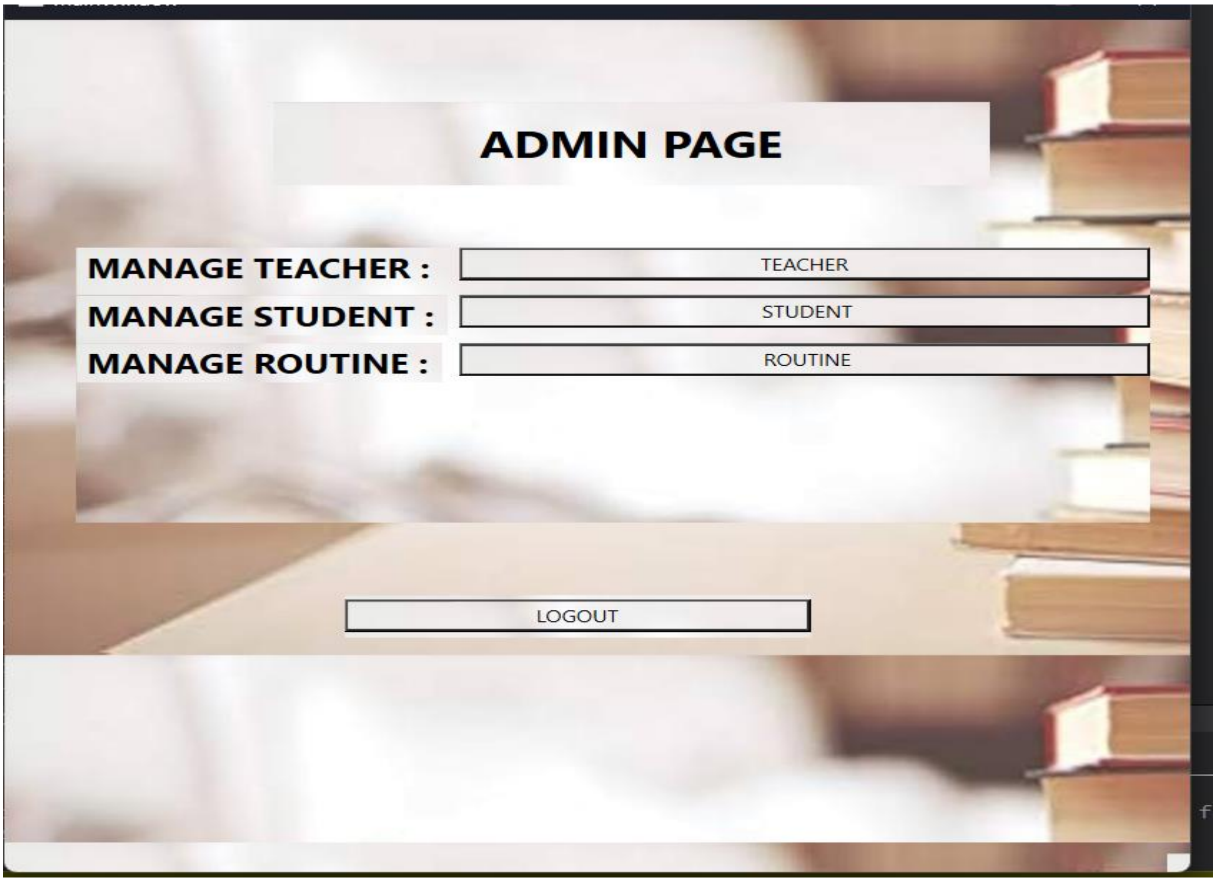


FIGURE 4. 3 ADMIN PAGE

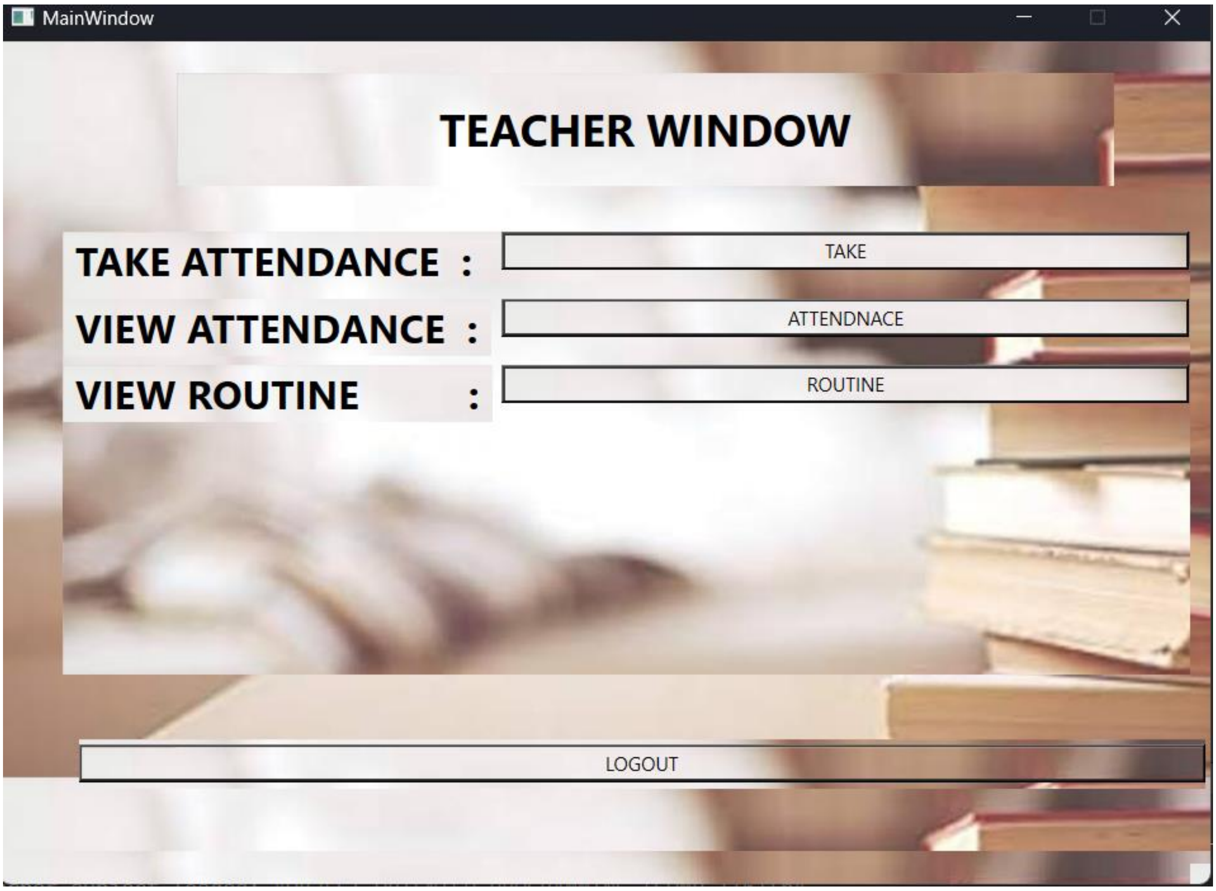


FIGURE 4. 4 TEACHER PAGE