**DESIGN AND IMPLEMENTATION OF REAL-TIME LECTURE TIMETABLE AND TASK SCHEDULING SYSTEM: A CASE STUDY OF IYA ABUBAKAR INSTITUTE OF ICT ABU, ZARIA**

**BY**

|  |  |
| --- | --- |
| **MATHIAS GODWIN KURUKUSHAYN** | **D22CSC1235** |
| **ISAH HARUNA** | **D22CSC1280** |
| **ABDULLAHI ABBA RABIU** | **D22CSC1302** |
| **YAKUBU HAMZA** | **D22CSC1241** |
| **ABUBAKAR ISHAQ** | **D22CSC1234** |
| **ABDULHAMID MUSA** | **D22CSC1255** |
| **KHADIJA IBRAHIM ABDULLAHI** | **D22CSC1260** |
| **LAWAL SHAMSUDEEN OMEIZA** | **D22CSC1237** |
| **GALADIMA EVELYN** | **D22CSC1245** |
| **ABDURRAUF ZARRUK** | **D22CSC1277** |
| **NASIRU ABDUL M.** | **D22CSC1244** |
| **ADAMU MUHAMMAD AMINU** | **D22CSC1239** |

**A PROJECT REPORT SUBMITTED TO**

**IYA ABUBAKAR INSTITUTE OF INFORMATION AND COMMUNICATION**

**TECHNOLOGY, AHMADU BELLO UNIVERSITY, ZARIA**

**IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE AWARD OF A DIPLOMA IN COMPUTER SCIENCE**

**OCTOBER, 2024**

## DECLARATION

We hereby declare that this project report entitled “DESIGN AND IMPLEMENTATION OF REAL-TIME LECTURE TIMETABLE AND TASK SCHEDULING SYSTEM" was carried out by Mathias Godwin Kurukushayn, Isah Haruna, Abdullahi Abba Rabiu, Yakubu Hamza, Abubakar Ishaq, Abdulhamid Musa, Khadija Ibrahim Abdullahi, Lawal Shamsudeen Omeiza, Galadima Evelyn, Abdurrauf Zarruk, Nasiru Abdul M. Adamu Muhammad Aminu in the Iya Abubakar Institute of Information and Communication Technology (IAIICT), A.B.U, Zaria under the supervision of Mal. Aliyu Tetengi Ibrahim. The information derived from literature has been duly acknowledged in the text and a list of reference provided. No part of this work has been submitted anywhere else for the award of diploma or degree certificate.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**MATHIAS GODWIN KURUKUSHAYN DATE**

**D22CSC1235**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ISAH HARUNA DATE**

**D22CSC1280**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ABDULLAHI ABBA RABIU DATE**

**D22CSC1302**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**YAKUBU HAMZA DATE**

**D22CSC1241**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ABUBAKAR ISHAQ DATE**

**D22CSC1234**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ABDULHAMID MUSA DATE**

**D22CSC1255**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**KHADIJA IBRAHIM ABDULLAHI DATE**

**D22CSC1260**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**LAWAL SHAMSUDEEN OMEIZA DATE**

**D22CSC1237**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**GALADIMA EVELYN DATE**

**D22CSC1245**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ABDURRAUF ZARRUK DATE**

**D22CSC1277**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**NASIRU ABDUL M. DATE**

**D22CSC1244**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ADAMU MUHAMMAD AMINU DATE**

**D22CSC1239**

## 

**CERTIFICATION**

This project report titled “DESIGN AND IMPLEMENTATION OF REAL-TIME LECTURE TIMETABLE AND TASK SCHEDULING SYSTEM” which meets the regulations governing the award of Diploma in computer Science and is approved for its contribution to knowledge and literary presentation.

|  |  |  |
| --- | --- | --- |
| MAL. ALIYU TETENGI IBRAHIM. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| **(Project Supervisor)** | **Signature** | **Date** |
| MRS.BATHAMA A. BELLO | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| **(Chairman, Team of Assessors)** | **Signature** | **Date** |

MR. ODUBU SUNDAY C. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(Project Coordinator) Signature Date**

DR. MRS. J.O ODENGLE **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**(Head, ETU) Signature Date**

MAL. S.M UMAR \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(Director, IAIICT) Signature Date**

## DEDICATION

This project is dedicated to God Almighty our creator, our strong pillar, our source of inspiration, wisdom, knowledge, and understanding for his divine guidance and protection over our lives. He has been the source of our strength throughout this program and the time to accomplish this project.

## ACKNOWLEDGEMENT

We are outstandingly grateful and gracious to the Almighty God, who granted us time and life by allowing us to see and witness the successful completion of this project work through all challenges and obstacles. May the name of God be highly praised!

Our immeasurable and strong gratitude to our project supervisor, Mal. Aliyu Tetengi Ibrahim, who patiently took out his time to guide and counsel us just to see that this project is completed successfully. It is our earnest prayer that God bless and strengthen you and your family.

Our gratitude also extends to the Director of the institute Mal. S.M Umar, Head of ETU Dr. Mrs. J.O Odengle, and the entire staff of IAIICT.

Salutations of praise and appreciation extend to our respective parents for their support morally economically and intellectually in the accomplishment of this task.

Our colleagues, friends, relatives, and well-wishers whose names cannot be mentioned here will not be left aside for their support to us.

## ABSTRACT

*In many academic institutions, effective scheduling and communication are essential to ensure the smooth running of daily operations. However, manual methods often lead to inefficiencies, errors, and communication gaps between staff and students. To address these challenges, this project presents the design and implementation of a Real-Time Lecture Timetable and Task Scheduling System at the Iya Abubakar Institute of ICT, ABU Zaria. The system allows for the efficient creation, modification, and management of timetables, while also sending real-time notifications to students regarding added or updated lectures and events. This feature enhances communication between lecturers and students, ensuring they are always updated on upcoming classes and deadlines. The system also incorporates task scheduling, allowing students to track assignments and important academic events. The development followed the Structured System Analysis and Design Method (SSADM), ensuring a systematic approach. Technologies used include HTML, CSS, and JavaScript for the front-end, with PHP and MySQL handling server-side scripting and database management, respectively. The system’s notification module leverages real-time updates to keep students informed, reducing confusion and conflicts in class schedules. Overall, this project aims to enhance operational efficiency, reduce human error, and improve communication between staff and students at the institute.*

## TABLE OF CONTENTS

[DECLARATION ii](#_Toc178022931)

[DEDICATION v](#_Toc178022932)

[ACKNOWLEDGEMENT vi](#_Toc178022933)

[ABSTRACT vii](#_Toc178022934)

[TABLE OF CONTENTS viii](#_Toc178022935)

[LIST OF TABLES xi](#_Toc178022936)

[LIST OF FIGURES xii](#_Toc178022937)

[LIST OF ACRONYMS xiii](#_Toc178022938)

[CHAPTER ONE 1](#_Toc178022939)

[INTRODUCTION 1](#_Toc178022940)

[1.0 Background of Study 1](#_Toc178022941)

[1.1 Statement of Problem 3](#_Toc178022942)

[1.2 Aim and Objectives 4](#_Toc178022943)

[1.3 Significance of the Project 5](#_Toc178022944)

[1.4 Scope of the Project 6](#_Toc178022945)

[1.5 Limitations of the Project 6](#_Toc178022946)

[CHAPTER TWO 7](#_Toc178022947)

[LITERATURE REVIEW 7](#_Toc178022948)

[2.1 Introduction 7](#_Toc178022949)

[2.2 Review of Fundamental Concepts 8](#_Toc178022950)

[2.2.1 Lecture Timetable Management System 8](#_Toc178022951)

[2.2.2 Task Scheduling and Reminder Systems 9](#_Toc178022952)

[2.2.3 Applications of Lecture Timetable Reminder and Task Scheduling Systems 9](#_Toc178022953)

[2.2.4 Tools 10](#_Toc178022954)

[2.2.4.1 Programming languages 10](#_Toc178022955)

[2.2.4.2 Hardware 12](#_Toc178022956)

[2.2.5 Security Threats 13](#_Toc178022957)

[2.3 Review of Related Works 13](#_Toc178022958)

[2.4 Summary of Gaps 16](#_Toc178022959)

[CHAPTER THREE 17](#_Toc178022960)

[METHODOLOGY 17](#_Toc178022961)

[3.1 Introduction 17](#_Toc178022962)

[3.2 Description of the Existing System 17](#_Toc178022963)

[3.2.1 Problems of the Existing System 17](#_Toc178022964)

[3.3 Analysis of the New Lecture Timetable and Notification System 19](#_Toc178022965)

[3.3.1 System Design and Implementation 20](#_Toc178022966)

[3.3.1.1 System Requirement 21](#_Toc178022967)

[3.3.1.2 Use Case Diagram 24](#_Toc178022968)

[3.3.1.2 System Architecture 26](#_Toc178022969)

[3.3.1.3 System Flowchart 28](#_Toc178022970)

[3.3.1.4 Database Design 31](#_Toc178022971)

[3.4 Database Tables 34](#_Toc178022972)

[3.4 Requirement and Analysis 38](#_Toc178022973)

[3.4.1 Fundamental Requirement 39](#_Toc178022974)

[3.4.2 Security Requirement 41](#_Toc178022975)

[CHAPTER FOUR 43](#_Toc178022976)

[RESULTS AND DISCUSSION 43](#_Toc178022977)

[4.1 Introduction 43](#_Toc178022978)

[4.2 Result of Design and Implementation 43](#_Toc178022979)

[4.2.1 User Interface 43](#_Toc178022980)

[4.2.1.1 Login Page 44](#_Toc178022981)

[4.2.1.2 Admin Dashboard 45](#_Toc178022982)

[4.2.1.3 User Home Page 46](#_Toc178022983)

[4.2.1.4 Timetable Page 47](#_Toc178022984)

[4.2.1.5 Notification Management Page 48](#_Toc178022985)

[4.2.1.6 Class Notification Sent 49](#_Toc178022986)

[4.2.1.7 User Management Page 50](#_Toc178022987)

[4.2.1.8 Assignment Submission Page 51](#_Toc178022988)

[4.2.1.9 Assignment Review Page 52](#_Toc178022989)

[4.3 Results of Simulation 52](#_Toc178022990)

[4.4 Discussion 53](#_Toc178022991)

[4.4.1 Implications of the Results 54](#_Toc178022992)

[CHAPTER FIVE 56](#_Toc178022993)

[CONCLUSION AND RECOMMENDATIONS 56](#_Toc178022994)

[5.1 Conclusion 56](#_Toc178022995)

[5.2 Recommendations 56](#_Toc178022996)

[References 58](#_Toc178022997)

[Appendix 62](#_Toc178022998)

## LIST OF TABLES

[Table 3.1: Users Table 34](#_Toc178022999)

[Table 3.2: Lecturers Table 35](#_Toc178023000)

[Table 3.3: Students Table 35](#_Toc178023001)

[Table 3.4: Notifications Table 36](#_Toc178023002)

[Table 3.5: Events Table 37](#_Toc178023003)

[Table 3.6: Tasks Table 38](#_Toc178023004)

## LIST OF FIGURES

[Figure 3.1: Admin Use Case Diagram 25](#_Toc177573427)

[Figure 3.2: Lecturer Use Case Diagram 25](#_Toc177573428)

[Figure 3.3: Student Use Case Diagram 26](#_Toc177573429)

[Figure 3.4: System Architecture 26](#_Toc177573430)

[Figure 3.5: Admin Flow Chart](#_Toc177573432) 28

[Figure 3.6: Lecturer Flow Chart 29](#_Toc177573433)

[Figure 3.7: Student Flow Chart 30](#_Toc177573434)

[Figure 3.8: Database Workflow 31](#_Toc177573435)

[Figure 3.9: Entity Relationship Diagram 32](#_Toc177573436)

[Figure 3.10: Entity-Relationship Schema 33](#_Toc177573437)

[Figure 4.1: Login Page 44](#_Toc177573438)

[Figure 4.2: Admin Dashboard 45](#_Toc177573439)

[Figure 4.3: User Home Page 46](#_Toc177573440)

[Figure 4.4: Timetable Page 47](#_Toc177573441)

[Figure 4.5: Notification Management Page 48](#_Toc177573442)

Figure 4.6: Class Notification Sent………………………………………………………………49

[Figure 4.7: User Management Page 50](#_Toc177573443)

[Figure 4.8: Assignment Submission Page 51](#_Toc177573444)

[Figure 4.9: Assignment Review Page 52](#_Toc177573445)

## LIST OF ACRONYMS

|  |  |
| --- | --- |
| CSS | Cascading Style Sheet. |
| HTML | Hypertext Mark-up Language. |
| ICT | Information and Communication Technology. |
| PHP | Hypertext Pre-processor. |
| SQL | Structured Query Language. |
| DBMS | Database Management System |

## CHAPTER ONE

## INTRODUCTION

## 1.0 Background of Study

In today's educational institutions, effective management of lecture timetables and tasks is critical for both students and lecturers. However, many institutions continue to rely on manual or semi-automated systems, which often lead to inefficiencies and frustrations. Traditional methods of scheduling, such as using paper-based timetables, bulletin boards, and manual registers, are still prevalent in many educational settings. These methods are time-consuming and prone to errors, as they require significant manual input and coordination (Davis & Wilson, 2022).

The primary features of traditional scheduling methods include physical timetables posted in common areas, manual recording of attendance, and verbal or written reminders for upcoming tasks and assignments. While these methods may have been sufficient in the past, they present several challenges in the modern educational environment. Firstly, the reliance on physical timetables means that any changes to the schedule must be manually updated and communicated, which can lead to confusion and missed classes (Smith & Roberts, 2021). Secondly, manual attendance recording is not only time-consuming but also susceptible to inaccuracies and potential manipulation (Anderson & Brown, 2021).

Moreover, the lack of centralized and real-time access to schedule information can lead to inefficiencies and decreased productivity. Students may miss lectures due to outdated or incorrect timetable information, and lecturers may find it challenging to coordinate their schedules and track student progress effectively (Johnson & Taylor, 2023). Additionally, traditional reminder methods, such as verbal or written notices, can easily be forgotten or overlooked, resulting in missed assignments and deadlines (Martin & Lee, 2022).

Recognizing these challenges, the need for a real-time lecture timetable reminder and task scheduling system becomes evident. Such a system can offer real-time data analysis, standardized assessment tools, and a centralized platform for managing and tracking student performance. Implementing a web-based system can significantly enhance the efficiency and effectiveness of the evaluation process at the Iya Abubakar Institute of ICT, ABU Zaria, by providing automated updates, notifications, and easy access to schedule information (Sun & Zhang, 2021; Wang & Li, 2023).

A real-time lecture timetable reminder and task scheduling system integrates modern Information and Communication Technology (ICT) to streamline and enhance the scheduling process. Such systems leverage automation and digital notifications to ensure that students and lecturers are always informed and prepared (Jia, Wang, & Rao, 2022). By providing real-time updates and multiple reminder techniques, including browser notifications, email, and SMS, the system aims to reduce the administrative burden and improve overall academic performance (Liu & Yu, 2021).

The primary features of a real-time lecture timetable system include the automated generation of timetables, instant notifications of changes or updates, and the ability to access the schedule from any device with internet connectivity. Lecturers can post notices of assignments, upcoming exams, and important events like holidays, ensuring that all students receive timely information. This system also supports the integration of calendar events, allowing students to see their academic schedule alongside personal commitments (Smith & Jones, 2023).

One significant benefit of this system is its ability to send lecture notifications and reminders via email, ensuring that students do not miss important lectures or deadlines. By centralizing all schedule-related information, the system minimizes confusion and the risk of missed communications. This not only helps students manage their time more effectively but also assists lecturers in maintaining organized and efficient classroom management (Chen & Zhang, 2022). The system can also provide analytical insights into attendance patterns and student engagement, allowing educators to identify and address potential issues proactively. For instance, automated alerts can be sent if a student misses multiple lectures, prompting intervention before academic performance is adversely affected (Gao & Lee, 2021).

However, there are challenges associated with implementing a real-time lecture timetable system. These include the initial setup cost, the need for reliable internet access, and the potential resistance to change from staff and students accustomed to traditional methods. Moreover, ensuring data security and privacy is crucial, as the system will handle sensitive student information (Kim & Park, 2023).

## 1.1 Statement of Problem

The current methods of managing lecture timetables and task schedules in many educational institutions, including the Iya Abubakar Institute of ICT, ABU Zaria, are outdated and inefficient. These traditional methods primarily rely on manual processes, which can lead to several significant issues.

Firstly, missed lectures and deadlines are a common problem. Without timely and automated reminders, students often miss important lectures and fail to meet assignment deadlines. This issue is exacerbated by the lack of a centralized system to notify students of upcoming lectures and tasks (O'Neil & Perez, 2022).

Also, communication between students and lecturers is often inefficient. Traditional methods such as notice boards and verbal announcements are not always effective, leading to gaps in information dissemination. This can result in students being unaware of changes in the timetable or new assignments (Kumar & Singh, 2021).And managing and accessing timetables and tasks, particularly during changes or updates, is cumbersome. When timetable changes occur, manually updating and communicating these changes to all students can be slow and prone to errors, causing confusion and disruption (Davies & Jones, 2023).

To address these problems, there is a need for a real-time system that automates reminders, facilitates timely communication, and improves attendance tracking. The proposed system will provide automated lecture reminders and enable lecturers to post notices of assignments and events such as holidays. By sending notifications to students via email and providing a centralized platform for accessing updated timetables and tasks, this system aims to enhance the overall management of academic schedules. The implementation of such a system can lead to improved punctuality, better communication, accurate attendance records, and a more organized academic environment (Smith & Adams, 2023).

## 1.2 Aim and Objectives

**Aim**

The aim of the Real-Time Lecture Timetable Reminder and Task Scheduling System is to automate and enhance the management of academic schedules and tasks, improving efficiency and communication within educational institutions.

**Objectives**

The specific objectives are:

1. To develop a real-time lecture timetable system that allows for the efficient creation, modification, and management of academic schedules.
2. To incorporate a notification module to send real-time updates to students regarding newly added or updated lectures, tasks, and events.
3. To leverage modern technologies like HTML, CSS, JavaScript (front-end), and PHP with MySQL (back-end) to create a robust, interactive, and user-friendly system.

## 1.3 Significance of the Project

The significance of this project lies in its potential to benefit multiple stakeholders within the educational ecosystem. For students, the system enhances organization and time management through real-time reminders and task scheduling, ensuring they are always informed about their lecture schedules and upcoming assignments. For lecturers, the system improves class management by facilitating better communication with students. Additionally, educational institutions will benefit from increased efficiency and a reduced administrative workload, leading to better resource allocation and overall management. By addressing these needs, the Real-Time Lecture Timetable Reminder and Task Scheduling System aims to foster a more efficient, responsive, and effective educational environment (Ahmed, Iqbal, & Shah, 2022; Zhang & Li, 2021).

## 1.4 Scope of the Project

The scope of this project encompasses the development and implementation of a system with several key features. Firstly, it includes a real-time feedback mechanism that allows students to receive immediate notifications and updates regarding their academic events. The project also incorporates multiple reminder techniques, such as browser notifications, email, and SMS, to ensure that students are consistently informed about their schedules and assignments. Lastly, the system features calendar integration with pre-installed events, including holidays, to provide a comprehensive and user-friendly scheduling experience. The system does include SMS functionality for notifications, giving the ability to send real-time updates to students who may not have access to email or web notifications.

## 1.5 Limitations of the Project

The system relies heavily on real-time communication, which means it lacks offline access or synchronization features. Users cannot access or update their schedules without an internet connection. The system’s authentication process is not integrated with the student portal, meaning it cannot verify if a user is a valid student, which could potentially allow unauthorized access to the system if the admin deliberately added users that are not students.

## CHAPTER TWO

## LITERATURE REVIEW

## 2.1 Introduction

This chapter reviews fundamental concepts and theories relevant to the design and implementation of a real-time lecture timetable reminder and task scheduling system. It provides an overview of similar and related works in the field, highlighting advancements and limitations in existing systems.

Real-time lecture timetable reminder and task scheduling systems are designed to optimize the management of academic schedules and tasks, ensuring that students and lecturers are well-informed and organized. These systems utilize automated processes for creating, managing, and communicating schedules, which significantly reduce the administrative burden associated with manual scheduling. For example, the EduSec ERP system offers features such as easy timetable scheduling, effortless subject allocation, and automated notifications via SMS or email for updates or faculty replacements (EduSec, 2023). Such systems can also manage attendance without manual intervention, providing precise records and reducing human errors (EduSec, 2023).

Task scheduling and resource management in cloud computing environments often involve heuristic approaches to handle the allocation of computational resources efficiently. These approaches ensure that incoming tasks are prioritized and managed according to their requirements, which is crucial for maintaining the system's responsiveness and effectiveness (Hossain & Khan, 2021). The combination of real-time updates, automated notifications, and efficient resource management contributes to the overall effectiveness and reliability of modern timetable and task scheduling systems (Hossain & Khan, 2021).

## 2.2 Review of Fundamental Concepts

This section delves into the fundamental concepts surrounding real-time lecture timetable reminder and task scheduling systems. Such systems are designed to optimize the management of academic schedules and tasks, ensuring that students and lecturers are well-informed and organized.

## 2.2.1 Lecture Timetable Management System

A lecture timetable management system is a comprehensive software solution aimed at automating and streamlining the scheduling of lectures and academic activities. It includes features for creating, updating, and managing timetables, ensuring that all stakeholders have access to accurate and up-to-date information. The system aims to reduce administrative burden, prevent scheduling conflicts, and enhance overall academic efficiency (Ahmad et al., 2019).

Recent advancements in these systems have incorporated machine learning and artificial intelligence to optimize scheduling and resource allocation. According to Sharma et al. (2021), machine learning algorithms can predict scheduling conflicts and suggest optimal timetable adjustments, thereby improving the overall efficiency and effectiveness of timetable management. Additionally, these systems now offer real-time updates and notifications, ensuring that students and faculty are immediately informed of any changes, thus minimizing disruptions (Sharma et al., 2021).

Furthermore, modern lecture timetable management systems integrate seamlessly with other educational technologies, such as learning management systems (LMS) and student information systems (SIS). This integration facilitates automatic synchronization of course offerings and student enrollments, which enhances the accuracy and reliability of timetables. Studies by Wang and Li (2022) have shown that such integration not only streamlines administrative processes but also provides valuable data analytics for institutional planning and resource management (Wang & Li, 2022).

## 2.2.2 Task Scheduling and Reminder Systems

Task scheduling and reminder systems are designed to help users manage their time and responsibilities effectively. These systems provide automated notifications and reminders through various channels, such as browser notifications, emails, and SMS. The primary goal is to ensure that users do not miss important tasks or deadlines, thereby improving productivity and time management. Modern task management software, such as ClickUp and Acuity Scheduling, offer a range of features including real-time synchronization, customizable scheduling options, and integrations with other business tools. These features help users maintain efficiency and avoid conflicts in their schedules (Adamenfroy, 2024; ClickUp, 2024).

## 2.2.3 Applications of Lecture Timetable Reminder and Task Scheduling Systems

These systems have various applications within educational institutions. They facilitate the organization of academic schedules, provide timely reminders for lectures and assignments, enable student feedback, and allow lecturers to track attendance. By integrating these functionalities, the system enhances communication and coordination among students and lecturers, contributing to a more efficient and responsive educational environment.

Modern scheduling systems, such as those provided by FET and Acuity Scheduling, offer advanced features like automated scheduling, conflict resolution, and resource optimization. These features address the common challenges of manual timetable creation, such as class clashes and inefficient use of resources, by allowing for the seamless management of large amounts of data involving students, teachers, rooms, and various academic needs (Times Higher Education, 2023; IJSER, 2023).

These systems often integrate with other institutional tools to streamline administrative processes. For example, automated reminders and notifications ensure that students and staff are kept informed about schedule changes, upcoming deadlines, and important events, thus reducing the likelihood of missed classes or assignments. Systems like nTask and SimplyBook also offer functionalities that support real-time updates and feedback mechanisms, enhancing the overall learning and teaching experience (Adamenfroy, 2024; TechRadar, 2024).

The integration of mobile and web-based platforms enables greater accessibility and flexibility. Students and staff can access schedules, receive notifications, and provide feedback from any location, fostering a more connected and adaptive educational environment (Academia.edu, 2023). The use of AI and machine learning in these systems further enhances their efficiency by predicting optimal scheduling patterns and automating routine tasks (Springer, 2023).

## 2.2.4 Tools

The development of a real-time lecture timetable reminder and task scheduling system relies on several programming language, software and hardware tools.

## 2.2.4.1 Programming languages

These programming languages are used to build the system's functionalities and make sure the system operates properly:

1. Hypertext Markup Language (HTML)

HTML is the standard markup language used to create the structure of web pages. It serves as the backbone of web content, allowing browsers to interpret and display text, images, and other media. HTML documents are essential for presenting data on the web and can be identified by the file extensions .htm or .html. Recent advancements in HTML5 have introduced new elements and attributes that enhance multimedia support and improve the semantic structure of web pages (W3C, 2021; Freeman & Robson, 2019).

1. Cascading Style Sheets (CSS)

CSS is a style sheet language used to describe the presentation of a document written in HTML. It controls the layout, colors, fonts, and overall visual appearance of web pages. CSS allows for the separation of content and design, enabling consistent and visually appealing web pages. With the advent of CSS3, new features such as animations, transitions, and flexible box layouts (Flexbox) have been introduced, enhancing the user experience and providing more design flexibility (Bos et al., 2020; Meyer, 2020).

1. JavaScript:

JavaScript is a versatile, high-level programming language commonly used to create interactive effects within web browsers. It enables dynamic content updates, form validation, and user interactions without requiring a page reload. JavaScript can be used on both the client and server sides, with frameworks such as Node.js allowing for scalable network applications. Modern JavaScript frameworks and libraries like React, Angular, and Vue.js facilitate the development of complex, single-page applications (Eich, 2021; Flanagan, 2020).

1. PHP:

PHP is a popular server-side scripting language designed for web development. It can be embedded into HTML and is used to manage server-side tasks such as form processing, session management, and database interactions. PHP's flexibility and ease of use make it a preferred choice for building dynamic and interactive websites. The language's extensive library of built-in functions and robust community support contribute to its widespread adoption (Suraski & Gutmans, 2020; Sklar & Trachtenberg, 2019).

1. MySQL:

MySQL is an open-source relational database management system (RDBMS) that uses Structured Query Language (SQL) for database access and management. It is widely used for storing and managing data related to timetables, tasks, user information, and more. MySQL's reliability, performance, and ease of use make it a popular choice for web applications. It supports data integrity, backup, and recovery, ensuring that data remains accurate and available (Widenius & Axmark, 2022; DuBois, 2021).

## 2.2.4.2 Hardware

The hardware infrastructure supporting the system is critical for its performance and reliability. Servers, which host the central database and application, are essential for processing and storing large amounts of data. Workstations provide users with access points, enabling them to interact with the system efficiently. Additionally, mobile devices and personal computers are crucial for accessing features and receiving notifications. Network devices ensure robust and secure connectivity across the system, facilitating seamless communication between different hardware components (Zhang et al., 2022; Patel & Shah, 2021).

## 2.2.5 Security Threats

The system faces various security threats that need to be addressed to ensure the protection of sensitive data and maintain system integrity. Common threats include unauthorized access, phishing attacks, and data breaches. To mitigate these risks, robust authentication mechanisms are implemented. Encryption protocols protect data during transmission and storage. Regular security audits and updates are essential to identify vulnerabilities and enhance security measures. These combined efforts help safeguard user data and ensure the system's reliability and trustworthiness (Sharma & Gupta, 2023; Alharbi et al., 2021).

## 2.3 Review of Related Works

Smith et al. (2021) developed an advanced lecture timetable management system aimed at enhancing the scheduling process in universities. The system utilized modern web technologies including HTML5, CSS3, JavaScript, and PHP. This system streamlined the scheduling process by providing an intuitive interface for both administrators and students, allowing for easy viewing and management of class schedules. The system's benefits included significant reductions in manual scheduling errors and the ability to quickly adapt to schedule changes. However, it lacked real-time notification features and comprehensive feedback mechanisms, limiting its ability to provide immediate updates and gather user input on schedule changes.

Chen and Wang (2022) created an automated reminder system for educational settings, integrating email and SMS notifications. This system significantly improved students’ adherence to deadlines and attendance rates by sending timely reminders. While effective in enhancing student engagement, it did not incorporate comprehensive feedback and attendance tracking functionalities, which are essential for a complete educational management system.

Rodriguez et al. (2022) designed an integrated task scheduling and reminder system for educational institutions. The system used a combination of modern web technologies and cloud-based services to deliver reminders via email, SMS, and mobile notifications. The primary benefits included enhanced student compliance with deadlines and improved overall academic performance. However, the system does not account for event management, which would allow students to stay informed about important event dates.

Lee and Kim (2023) developed a digital scheduling system for higher education institutions. Their system featured notifications and deadline management tools designed to boost student accountability and academic performance. The system included a user-friendly interface and allowed for the integration of various feedback mechanisms. Thus, the absence of a student feedback or commenting mechanism in their system highlights an important gap that could be filled by fostering more interaction and continuous improvement based on user input.

Johnson et al. (2023) built a comprehensive timetable management system that integrated real-time notifications and feedback mechanisms. This system aimed to address the limitations of previous systems by providing immediate updates on schedule changes and allowing users to provide feedback on the scheduling process. The system utilized a combination of web and mobile technologies to ensure accessibility and ease of use. Their system appears to focus on schedule updates and user feedback. However, it does not seem to address task scheduling or tracking of assignments and academic tasks, which are significant feature.

Ahmed et al. (2023) designed a smart timetable system using Python and Flask for backend processing, while JavaScript and Bootstrap were used for the frontend. The system featured automated schedule generation based on lecturer availability and student enrollment. While their system focuses on automated schedule generation based on lecturer availability and student enrollment, it does not to address real-time notifications or task scheduling, which are key components.

Williams and Turner (2023) introduced a cloud-based lecture scheduling system that used AWS Lambda and DynamoDB to manage serverless operations and provide real-time data storage. The system reduced server costs significantly but faced limitations in terms of latency, especially during peak usage hours. Moreover, while the system supported automated notifications, the lack of comprehensive user feedback mechanisms hindered real-time adaptability to schedule changes.

Brown and Singh (2023) developed a mobile-friendly timetable application using React Native and Firebase for database management. Their system enhanced user accessibility by ensuring students could access their schedules on mobile devices. Additionally, the system's reliance on Firebase led to issues with data synchronization during high-volume periods.

Harris et al. (2022) created a hybrid timetable and learning platform using Angular for the front end and Node.js for the back end, integrated with MongoDB. The system's primary strength was its ability to integrate learning materials with the timetable. However, a gap was identified in its notification system, which could not provide immediate alerts for schedule changes, reducing the system's effectiveness in rapidly changing environments.

Kumar and Patel (2023) designed an AI-powered scheduling system that used machine learning algorithms to predict and allocate classroom resources. The system was built using Python, TensorFlow, and Flask for backend processing, and Vue.js for the front end. Despite its innovation in resource prediction, the system struggled with integrating human feedback, as users could not override AI-generated schedules easily. Additionally, the complexity of implementing the AI model required ongoing training and updates, which posed a maintenance challenge.

The literature review has identified notable shortcomings in existing scheduling systems, including functionality limitations, security vulnerabilities, and integration challenges. Some of the key issues include the lack of real-time notifications, which can cause delays in schedule updates; the absence of comprehensive feedback mechanisms, limiting user interaction; and data synchronization issues across different platforms, which hinder seamless integration. This project is designed to address these issues by providing a comprehensive solution that enhances functionality, improves security, and facilitates seamless integration within the educational management framework. This underscores the significance of our project in contributing to the improvement of academic scheduling and task management systems.

## 2.4 Summary of Gaps

Despite the advancements in lecture timetable management and task scheduling systems, several challenges remain unaddressed. Some of the systems reviewed lack comprehensive integration of real-time reminders and feedback mechanisms, which are crucial for enhancing user engagement and ensuring timely updates. There is also a need for a more integrated approach that combines scheduling, reminders, feedback, and multi-channel notifications to provide a holistic solution for educational institutions.

## CHAPTER THREE

## METHODOLOGY

## 3.1 Introduction

This chapter describes the system design, system analysis (including user requirements and system hardware/software requirements), design strengths and weaknesses of the existing system, context-level diagrams, Entity Relationship Diagram, and system evaluation and testing.

## 3.2 Description of the Existing System

Management of student and lecture schedules for the students of Iya Abubakar Institute of ICT ABU Zaria is often hampered by traditional methods such as manual timetables and paper-based notifications. This method poses several challenges, including difficulties in updating schedules, delays in disseminating information, and an overall lack of efficiency.

## 3.2.1 Problems of the Existing System

The challenges facing the current manual scheduling and task management systems are multifaceted and contribute to inefficiency, confusion, and resource waste. These problems are:

1. Inefficiency: The manual process of managing timetables requires administrators to physically update and communicate changes to schedules, which is time-consuming and prone to delays. This can lead to missed classes or confusion when updates are not promptly communicated. Moreover, changes to the timetable, such as class cancellations or rescheduling, often involve contacting numerous individuals manually, increasing the likelihood of human error and further delays. As a result, the system is slow to respond to sudden changes, and this inefficiency impacts both staff and students.
2. Lack of Accessibility: In a manual system, access to updated timetables and notifications is typically restricted to physical bulletin boards or individual notifications. This means that students and lecturers may not have timely or easy access to the latest scheduling information, particularly if they are off-campus or away from bulletin boards. This lack of real-time access can lead to confusion, with students missing classes due to being unaware of changes, and lecturers potentially arriving for classes that have been rescheduled without their knowledge.
3. Redundancy:The current system often involves multiple versions of schedules being created, maintained, and distributed across different departments or locations. This redundancy increases the likelihood of conflicting or outdated information being circulated. For example, a lecturer might receive one version of a schedule, while students receive another, leading to confusion and errors. Additionally, redundant processes waste time, as administrators may have to manually reconcile discrepancies between different versions of the timetable.
4. Resource Waste:The reliance on paper-based systems not only makes the process cumbersome but also leads to the waste of physical resources. Printing multiple copies of timetables, notices, and updates consumes a significant amount of paper and ink, which is neither cost-effective nor environmentally sustainable. Over time, this resource waste adds up, increasing operational costs and contributing to environmental degradation through excessive paper use.

## 3.3 Analysis of the New Lecture Timetable and Notification System

The proposed lecture timetable and notification system is designed to address the significant inefficiencies identified in the existing system. This digital solution will introduce a centralized management system that consolidates all schedule-related data, allowing for streamlined and efficient updates. By centralizing the system, it eliminates the inconsistencies and redundancies that arise from multiple versions of timetables being manually updated and distributed across departments. All users, including students and lecturers, will have access to the most current schedule in real time, thus preventing confusion or missed classes. This real-time update feature ensures that any adjustments, such as class cancellations or rescheduling, are instantly communicated to all relevant parties, mitigating the delays caused by manual processes.

A critical aspect of this system is the inclusion of user profiles with role-based access control. This feature is essential in maintaining the integrity and security of the system. Only authorized personnel, such as administrators or specific staff members, will have the ability to modify the timetable, while students and lecturers will be able to view the schedule but not alter it. This clear distinction in access rights reduces the risk of unauthorized changes, thereby enhancing the accuracy and reliability of the timetable. Additionally, notifications will be integrated within the system, ensuring that students and lecturers receive timely reminders about upcoming lectures, assessments, or changes to the timetable. The combination of centralized management, real-time updates, and secure access control makes this system a robust solution for overcoming the limitations of traditional manual methods.

## 3.3.1 System Design and Implementation

The design phase is foundational to the successful development of the lecture timetable and notification system, setting the blueprint for how the system will function. This process involves careful consideration of the system’s structure, functionality, and user interaction. The design phase integrates three critical activities: planning, implementation, and testing, each of which plays an indispensable role in the overall success of the software development.

In the planning stage, key design decisions are made regarding the architecture of the system. This includes determining how the system’s various components, such as the user interface, database, and notification services, will interact with one another. A modular design approach is adopted to ensure that different components of the system can be developed, tested, and updated independently without affecting the entire system. This modularity enhances the system’s maintainability and scalability, making it easier to introduce new features or improvements in the future without requiring a complete system overhaul.

The implementation stage involves translating the design into actual code. Modern web technologies, including HTML5, CSS3, JavaScript, and PHP, are used to build the system’s front end and back end. The user interface is designed to be intuitive and user-friendly, ensuring that students, lecturers, and administrators can easily navigate the system. The database is structured to store all relevant data, including user profiles, course information, and timetable entries, in a secure and organized manner. The notification system is also integrated during this phase, utilizing email or SMS services to ensure that users receive timely updates on schedule changes.

Testing is an ongoing process throughout the development of the system. Each component is rigorously tested to ensure it functions as expected and interacts seamlessly with other parts of the system. Usability testing is also conducted to ensure that the user interface is accessible and intuitive. Security testing is particularly crucial in this phase, given the need to protect sensitive information such as student records and class schedules. The goal of testing is not only to identify and fix bugs but also to ensure that the system meets all user requirements and performs optimally under different scenarios.

The design and implementation process are iterative, meaning that feedback is continuously incorporated to improve the system. By focusing on user experience, security, and performance, the system is designed to provide a reliable and efficient solution to the challenges currently faced in lecture scheduling and notifications. This comprehensive approach to design and implementation lays the groundwork for a system that will be both effective and sustainable in the long term.

## 3.3.1.1 System Requirement

System requirements refer to the minimum configuration that the lecture timetable and notification system must have in order to run smoothly and efficiently. These requirements address installation and performance issues. The system requirements are categorized into hardware and software requirements.

**1. Hardware Requirement**

The computer system used as a local server to host the system should have the following specifications:

1. **Processor**

A processor with a speed of 2.50 GHz or higher is crucial for the smooth operation of the system. The processor, often referred to as the CPU, is the brain of the computer, responsible for executing instructions from applications and the operating system. Higher GHz indicates a faster processor, allowing the system to handle multiple tasks efficiently, including processing user requests and running complex operations without delays.

1. **Hard Disk Size**

A hard disk of at least 20 GB is necessary to store the operating system, software applications, and user data. In the context of a timetable and notification system, the hard disk space will be used to store various data such as timetables, user profiles, notifications, and logs. Adequate disk space ensures the system can manage large amounts of data without performance issues and allows for future scalability.

1. **RAM Size**

Random Access Memory (RAM) is critical for the system's performance, as it temporarily stores the data that the CPU uses to process tasks. With at least 1 GB of RAM, the system can run multiple applications simultaneously, handle user queries, and process notifications efficiently. Insufficient RAM can lead to slower performance, especially when running complex or resource-intensive applications.

**2. Software Requirement**

The software requirements for the new system include:

1. Operating System

Windows 10 is recommended for optimal performance due to its compatibility with a wide range of applications and hardware. It provides a stable and secure environment, with support for the latest software updates and security patches. Windows 10 also features enhanced multitasking capabilities, which are essential for running development tools, databases, and servers required for this system. Additionally, its user-friendly interface and regular updates ensure that the system remains functional and secure over time.

1. XAMPP:

XAMPP is a cross-platform, open-source web server solution that simplifies the process of testing and developing websites locally. It includes essential tools such as Apache (server), MySQL (database), PHP (scripting language), and Perl. By using XAMPP, developers can create and test web applications on their local machines before deploying them to a live server, ensuring the functionality of the system. XAMPP’s easy setup and control panel make it an ideal tool for developers to simulate live environments without needing a dedicated remote server.

1. Visual Studio Code:A streamlined code editor with support for development operations like debugging, task running, and version control.

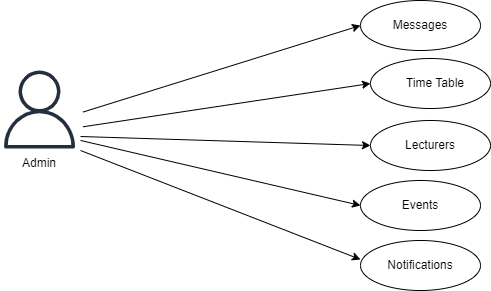
Visual Studio Code (VS Code) is a lightweight yet powerful code editor that supports a wide variety of programming languages and frameworks. It offers built-in tools for debugging, task running, and version control, making it an essential tool for software development. VS Code’s extensions allow developers to integrate tools like Git for version control, which is critical for managing code changes and collaborating on large projects. Additionally, its IntelliSense feature provides smart code completion, which enhances productivity by reducing coding errors.

1. Web Browser: An application for accessing websites on various devices, including desktops, laptops, tablets, and smartphones.

A web browser is a software application that enables users to access websites and web-based applications. Common browsers like Google Chrome, Mozilla Firefox, and Microsoft Edge are optimized for modern web technologies like HTML5, CSS3, and JavaScript. These browsers are essential for viewing and interacting with the lecture timetable and notification system, as they allow users to access the platform across multiple devices. Cross-browser compatibility ensures that the system works consistently, whether users are accessing it on desktops, laptops, tablets, or smartphones.

## 3.3.1.2 Use Case Diagram

A use case diagram is a visual representation of how a user might interact with a system. It depicts the system’s numerous use cases and different sorts of users, and it’s frequently accompanied by other types of diagrams. Circles or ellipses are used to depict the use cases. This system has two use case diagrams; namely: the admin, the lecturer and students.



Students

Figure 3.1: Admin Use Case Diagram

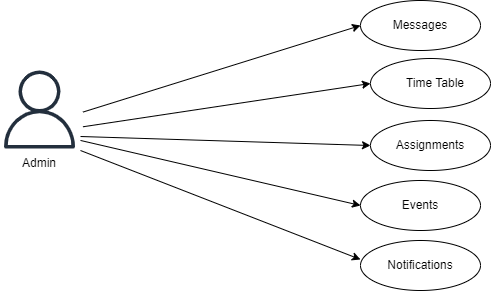


Figure 3.2: Lecturer Use Case Diagram

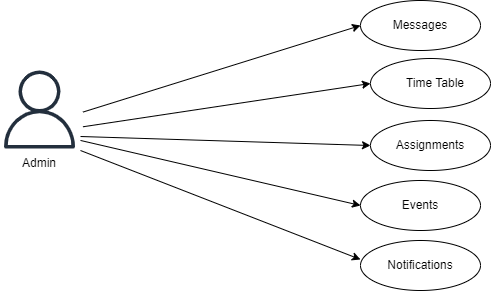


Figure 3.3: Student Use Case Diagram

## 3.3.1.2 System Architecture

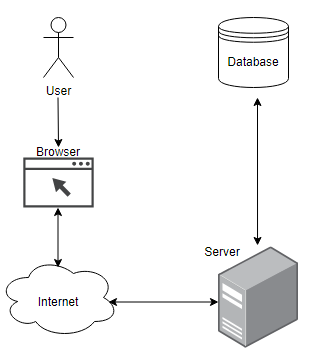
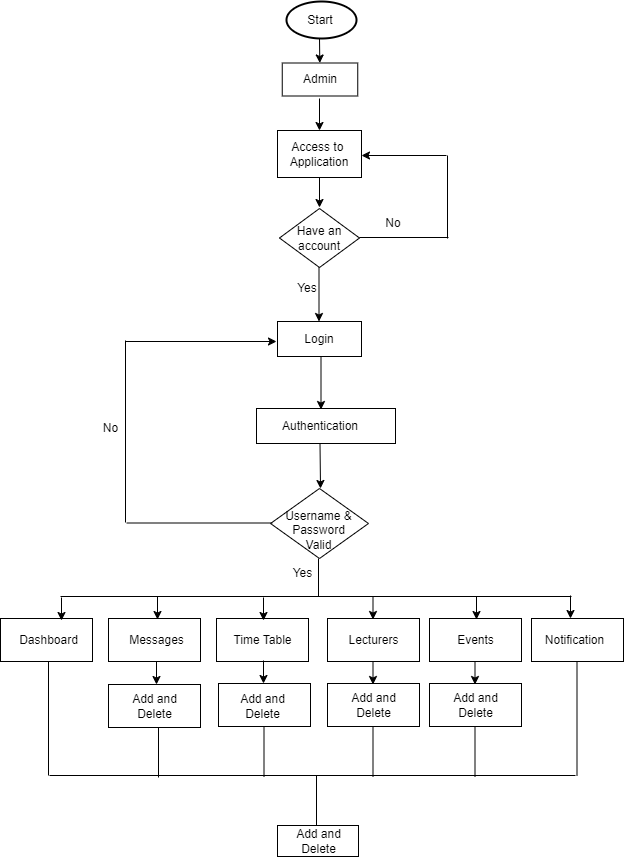
The system consists of components and subsystems that work together to implement the overall system. The diagram in figure 3.1 describes the system architecture.

Figure 3.4: System Architecture

The diagram above depicts the architecture of the lecture timetable and notification system, which includes an admin and user interface. The admin has the highest level of access and can add/remove users, while regular users have limited access based on permissions granted by the admin.

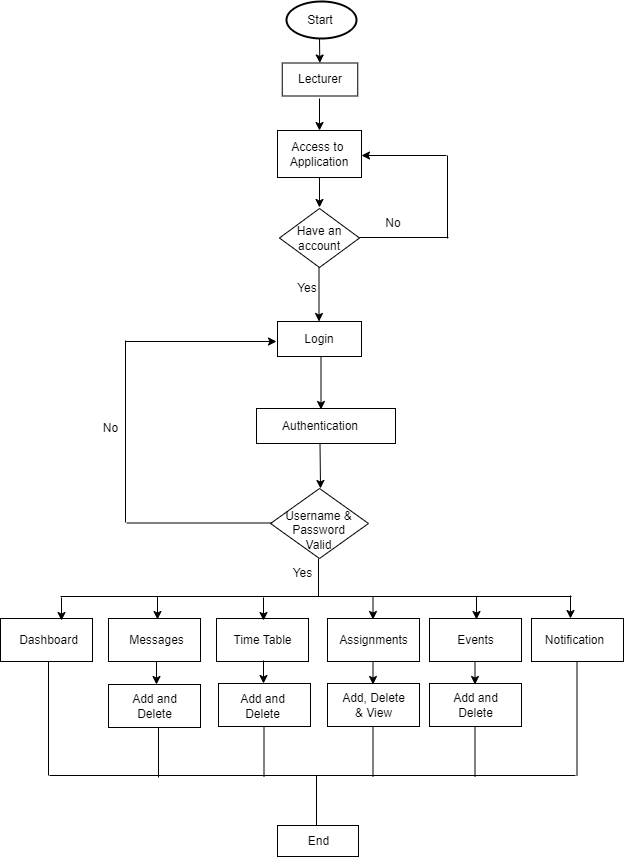
## 3.3.1.3 System Flowchart



Grant Access

End

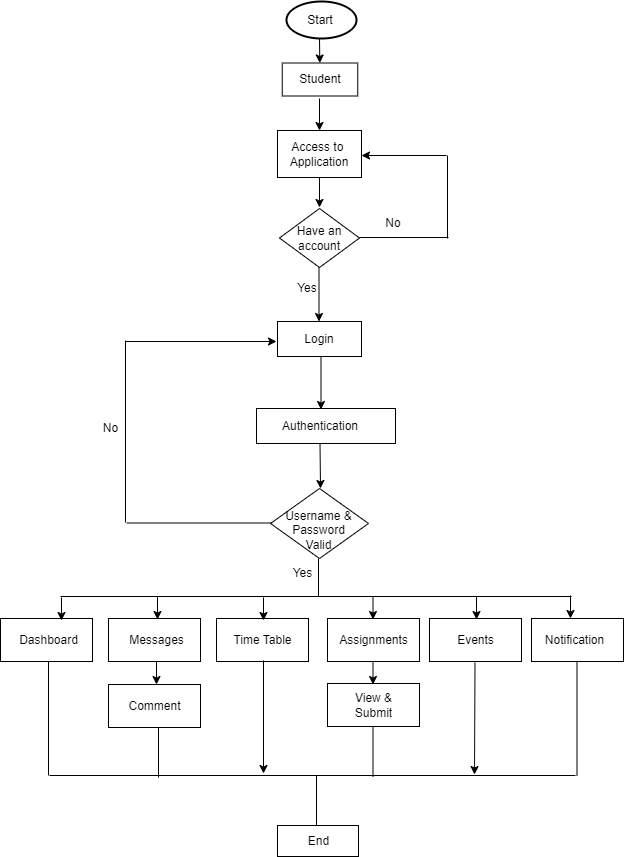
Figure 3.5: Admin Flow Chart



Grant Access

End

Figure 3.6: Lecturer Flow Chart



Grant Access

End

Figure 3.7: Student Flow Chart

## 3.3.1.4 Database Design

The database for the lecture timetable and notification system is designed to efficiently store and manage a vast amount of information related to users, schedules, tasks, and notifications.

**i. Database Workflow**

A database workflow connects processes to the database and applications. The database workflow for the system is shown in figure 3.3.

Create

Retrieve

U

p

d

a

t

e

D

e

l

e

t

e

A

p

p

l

i

c

a

t

i

o

n

P

r

o

g

r

a

m

m

e

D

a

t

a

b

a

s

e

S

y

s

t

e

m

Figure 3.8: Database Workflow

**ii. Entity Relationship Diagram**

The entity-relationship diagram represents the core structure and relationships within the lecture timetable and notification system. This diagram depicts essential entities, their attributes, and the associations between them.

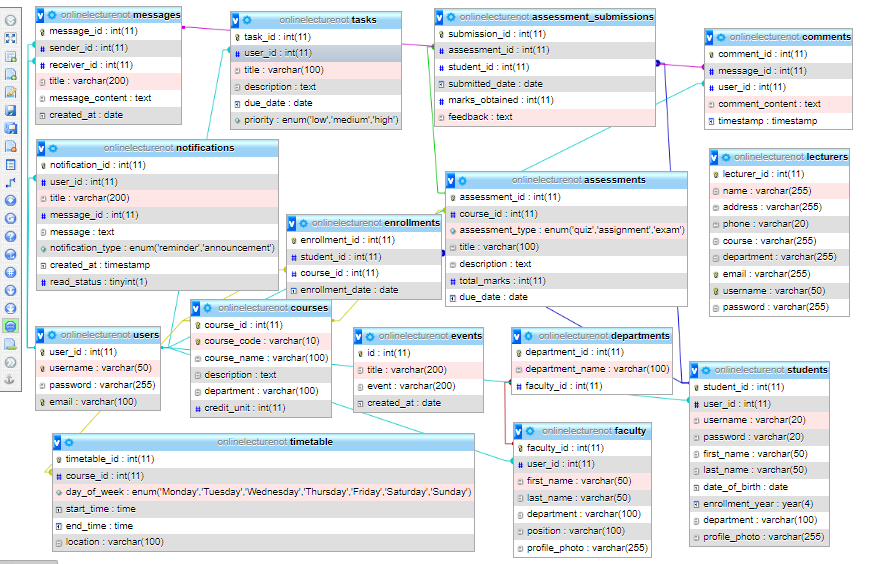


Figure 3.9: Entity Relationship Diagram

**iii. Entity Relationship Schema**

The Entity-Relationship (ER) schema outlines the database structure of the device verification system, depicting entities, relationships, and attributes that govern data interactions.

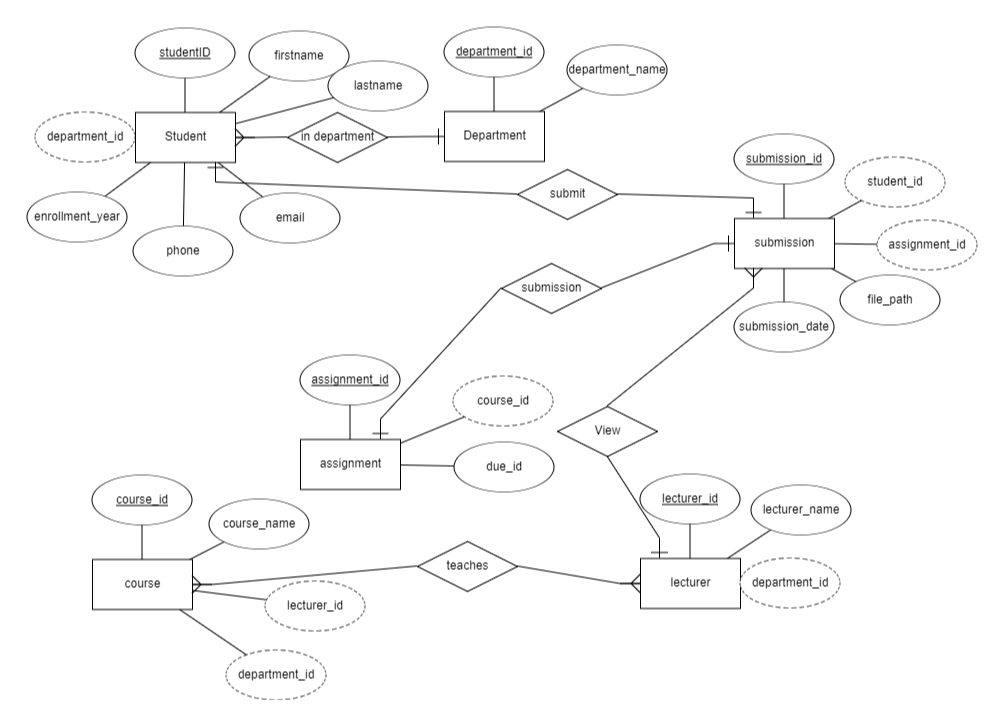


Figure 3.10: Entity-Relationship Schema

## 3.4 Database Tables

The database tables used in the lecture timetable and notification system are as follows:

Table 3.1: Users Table

|  |  |  |  |
| --- | --- | --- | --- |
| **SNO** | **NAME** | **TYPE** | **DESCRIPTION** |
| 1. | user\_id | INT | Primary key of the table. |
| 2. | Username | VARCHAR(50) | Username of the user. |
| 3. | Password | VARCHAR(255) | Password of the user. |
| 4. | Email | VARCHAR(100) | Email of the user. |
| 5. | Role | ENUM('student', 'lecturer', 'admin') | Role of the user. |
| 6. | created\_at | TIMESTAMP | Time when the record was created. |

Table 3.1 Shows the User Table, which comprises fields such as user\_id, username, password, email, role, and created\_at. The primary key is user\_id.

Table 3.2: Lecturers Table

|  |  |  |  |
| --- | --- | --- | --- |
| **SNO** | **NAME** | **TYPE** | **DESCRIPTION** |
| 1 | lecturer\_id | INT | Primary key of the table. |
| 2 | user\_id | INT | Foreign key referencing user\_id. |
| 3 | Department | VARCHAR(100) | Department of the lecturer. |
| 4 | phone\_number | VARCHAR(15) | Phone number of the lecturer. |

Table 3.2 Shows the Lecturers Table, which includes fields such as lecturer\_id, user\_id, department, and phone\_number.

Table 3.3: Students Table

|  |  |  |  |
| --- | --- | --- | --- |
| SNO | NAME | TYPE | DESCRIPTION |
| 1 | student\_id | INT | Primary key of the table. |
| 2 | user\_id | INT | Foreign key referencing user\_id. |
| 3 | Department | VARCHAR(100) | Department of the student. |
| 4 | Level | VARCHAR(50) | Academic level of the student. |

Table 3.3 Shows the Students Table, which includes fields such as student\_id, user\_id, department, and level.

Table 3.4: Notifications Table

|  |  |  |  |
| --- | --- | --- | --- |
| SNO | NAME | TYPE | DESCRIPTION |
| 1 | notification\_id | INT | Primary key of the table. |
| 2 | user\_id | INT | Foreign key referencing user\_id. |
| 3 | Title | VARCHAR(255) | Title of the notification. |
| 4 | Message | TEXT | Content of the notification. |
| 5 | notification\_type | ENUM('reminder', 'alert', 'announcement') | Type of notification. |
| 6 | created\_at | TIMESTAMP | Time when the record was created. |
| 7 | read\_status | TINYINT(1) | Read status of the notification. |

Table 3.4 Shows the Notifications Table, which includes fields such as notification\_id, user\_id, title, message, notification\_type, created\_at, and read\_status.

Table 3.5: Events Table

|  |  |  |  |
| --- | --- | --- | --- |
| SNO | NAME | TYPE | DESCRIPTION |
| 1 | event\_id | INT | Primary key of the table. |
| 2 | Title | VARCHAR(255) | Title of the event. |
| 3 | Description | TEXT | Description of the event. |
| 4 | created\_at | TIMESTAMP | Time when the record was created. |

Table 3.5 Shows the Events Table, which includes fields such as event\_id, title, description, and created\_at.

Table 3.6: Tasks Table

|  |  |  |  |
| --- | --- | --- | --- |
| SNO | NAME | TYPE | DESCRIPTION |
| 1 | task\_id | INT | Primary key of the table. |
| 2 | user\_id | INT | Foreign key referencing user\_id. |
| 3 | Title | VARCHAR(255) | Title of the task. |
| 4 | description | TEXT | Description of the task. |
| 5 | due\_date | DATE | Due date of the task. |
| 6 | Status | ENUM('pending', 'completed') | Status of the task. |

Table 3.6 Shows the Tasks Table, which comprises of fields such as task\_id, user\_id, title, description, due\_date and status.

## 3.4 Requirement and Analysis

At this stage of development, information gathering is crucial for assessing the system’s current state and identifying areas for improvement. The analyst must use various data collection tools, such as interviews, questionnaires, and observations, to understand the system requirements in detail. This analysis will involve onsite observation as it provides the best firsthand information from the system's environment. By gathering data, the analyst will aim to develop a system that aligns closely with user expectations and is feasible for deployment.

## 3.4.1 Fundamental Requirement

The Fundamental Requirements refer to the core needs that the system must satisfy to function effectively. These requirements define the system’s overall objectives and guide the development process. Failure to meet any of these fundamental requirements could compromise the system’s utility, scalability, or usability. Key requirements include:

1. User Registration and Authentication.

The system must incorporate a robust registration and authentication mechanism. This includes allowing administrators to create, update, and manage user profiles with proper authentication credentials such as usernames, passwords, and, if necessary, multi-factor authentication (MFA). It is critical that users can register and access the system securely, with password encryption to prevent unauthorized access.

1. Data Management and Storage.

The system should be designed to handle large volumes of data efficiently. It must be capable of storing, retrieving, and managing data such as user records, system logs, transactions, and configurations. The database structure should be optimized to prevent data loss, duplication, and to support backup and restoration functionalities.

1. Responsive User Interface (UI).

A core requirement is that the system should be accessible across multiple devices and platforms. The user interface must be responsive, ensuring that users can access it from desktops, tablets, and mobile devices. The interface should be intuitive and user-friendly to minimize training needs.

1. OperationalEfficiencyandScalability**.**

The system must be scalable, allowing it to expand without performance degradation as the number of users, operations, or data increases. Scalability ensures the system can support future growth in terms of both user volume and functionality. Additionally, the system should be optimized for performance, ensuring minimal latency in data processing.

1. Error Handling and Recovery.

Robust error handling mechanisms must be integrated into the system to ensure the smooth functioning of processes. In case of system failure, appropriate recovery procedures should be in place to restore data integrity and resume operations without significant disruptions. The system must also log errors for future audits and debugging.

1. Interoperability with Other Systems.

The system must be able to interface with other existing systems and platforms as needed. This includes integrating APIs for communication with external systems and ensuring that data formats are compatible for smooth data exchange.

1. Regulatory and Compliance Adherence.

The system must comply with relevant regulations and standards, particularly in terms of data protection (e.g., GDPR for data privacy). It must ensure that the handling, processing, and storage of personal information conform to legal and industry standards.

1. High Availability and Reliability.

The system should be available and operational at all times, minimizing downtime. High availability mechanisms, such as load balancing and failover strategies, should be in place to ensure that users experience minimal disruptions during operation. Reliability of system operations is a fundamental requirement to maintain user trust and business continuity.

These requirements are crucial for the overall success and sustainability of the system. By addressing these foundational needs, the system will not only meet the current user demands but will also be adaptable to future enhancements.

## 3.4.2 Security Requirement

Security is a critical concern, especially in systems that manage sensitive information. For this system, several security measures need to be implemented to prevent unauthorized access and maintain data integrity. These include:

1. User Authentication

Every user interacting with the system must go through a secure login process that validates their credentials before granting access.

1. Data Encryption

All sensitive data should be encrypted to prevent unauthorized viewing or tampering.

1. Role-based Access Control

Access to system functionalities and data should be restricted based on the user’s role to ensure that only authorized personnel can modify sensitive information.

1. Audit Trail

The system should log all user activities to create a transparent record that can be audited to trace any suspicious behaviour.

This is a more detailed expansion of the requirement sections, giving more insight into the system's fundamental and security needs based on the example provided.

## CHAPTER FOUR

## RESULTS AND DISCUSSION

## 4.1 Introduction

This chapter presents the results obtained from the design, implementation, and simulation of the Online Lecture Timetable and Notification System. The results are analyzed, interpreted, and discussed to show how they address the problems identified in Chapter 1.

## 4.2 Result of Design and Implementation

## 4.2.1 User Interface

The user interface (UI) is the primary point of interaction between the user and the system. It includes all elements through which users interact with the system, such as display screens, keyboards, a mouse, buttons, icons, and other visual elements. The UI for the Online Lecture Timetable and Notification System was meticulously crafted using modern web technologies including HTML, CSS, and JavaScript to ensure a seamless and user-friendly experience. Several pages and components make up the system, each designed to provide a specific function. Screenshots of these pages are presented below to illustrate their structure and functionality.

## 4.2.1.1 Login Page

The Login Page serves as the gateway to the Online Lecture Timetable and Notification System. It is designed to be simple and intuitive, allowing users to enter their credentials such s username and password securely. The page features input fields for the credentials, along with a 'Login' button to submit the information. The design focuses on security and ease of use, employing input validation and error messaging to guide users through the login process effectively.

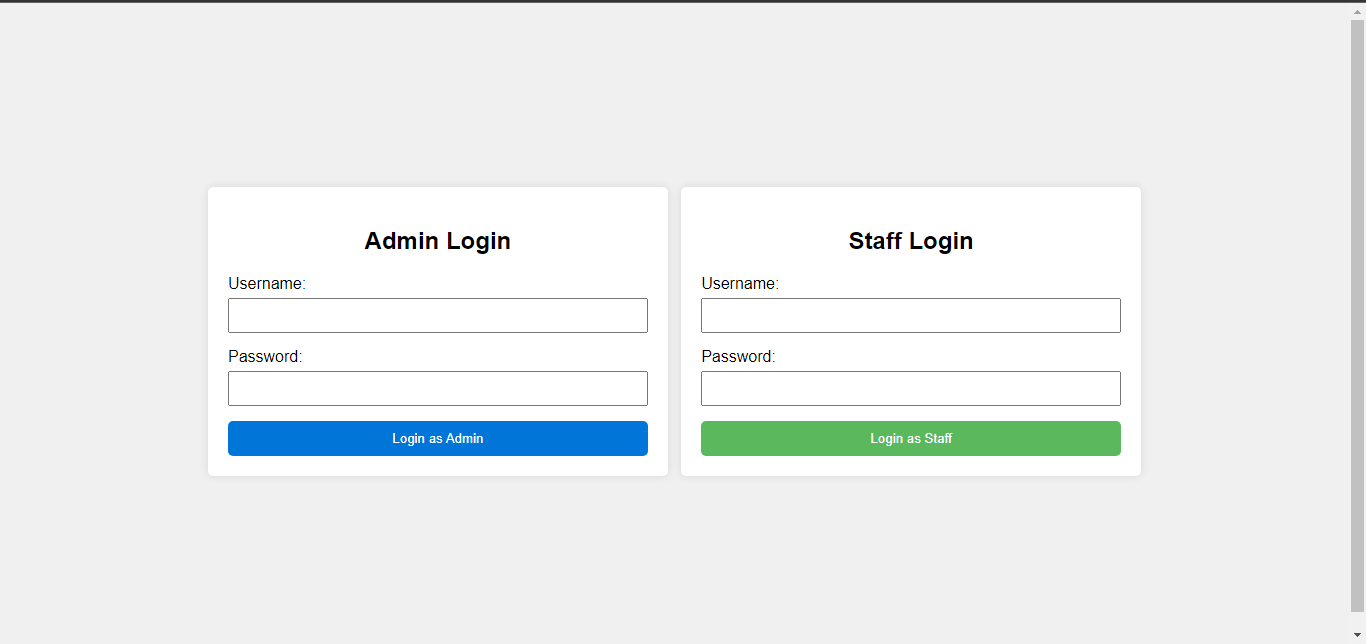


Figure 4.1: Login Page

## 4.2.1.2 Admin Dashboard

The Admin Dashboard is the central hub for administrators once they log in successfully. This page provides a comprehensive overview of system activities, including user management, class scheduling, and notification management. The dashboard is equipped with various widgets and charts to display statistics, such as the number of active users, scheduled classes, and pending notifications. It allows the admin to perform critical operations like adding or removing users, managing class schedules, and sending notifications. The dashboard's design emphasizes ease of navigation and quick access to frequently used functionalities.

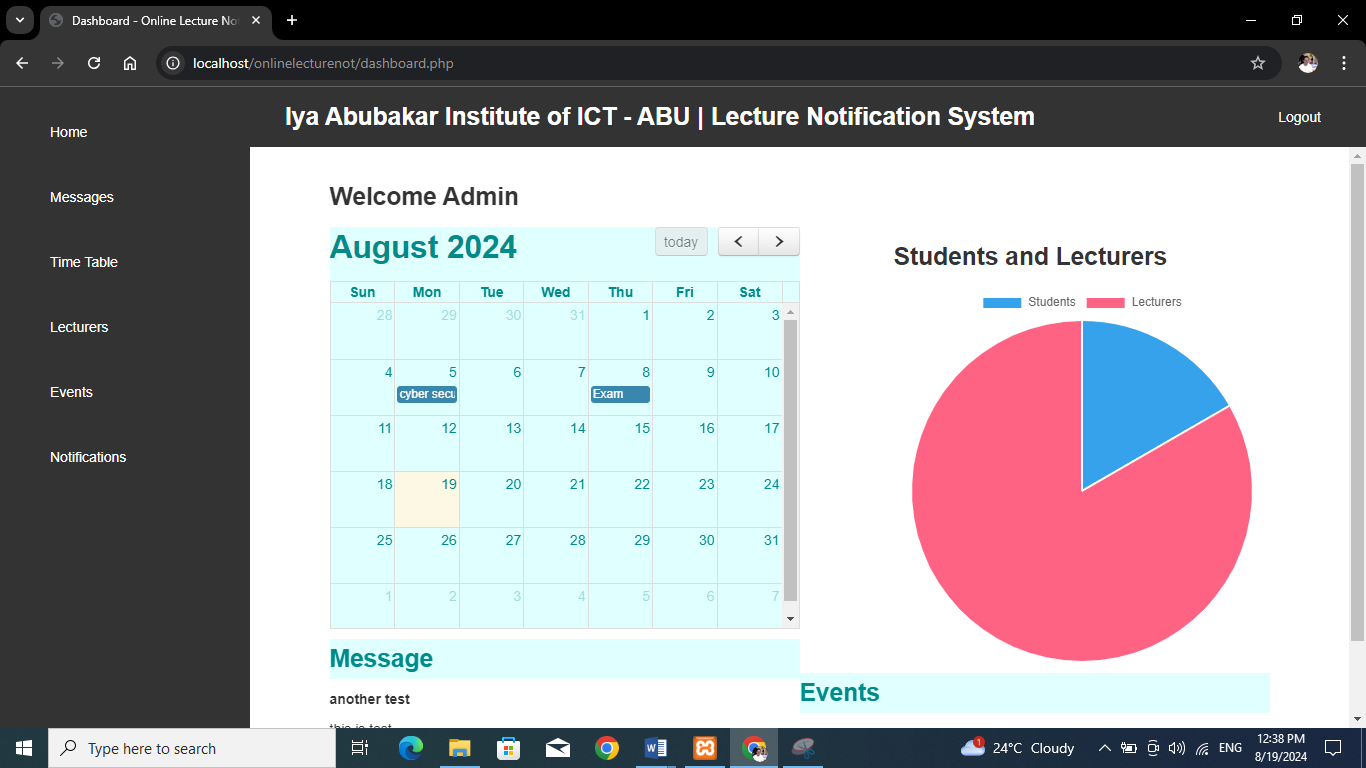


Figure 4.2: Admin Dashboard

## 4.2.1.3 User Home Page

The User Home Page is the landing page for all users other than the admin upon successful login. It provides an overview of the user’s schedule, upcoming classes, and recent notifications. The page is designed to offer a clear and organized view of relevant information, ensuring that users can easily find and manage their schedule and notifications. It includes interactive elements such as buttons and links that allow users to access detailed views of their classes or messages. The design focuses on simplicity and efficiency, minimizing the number of clicks required to access key information.

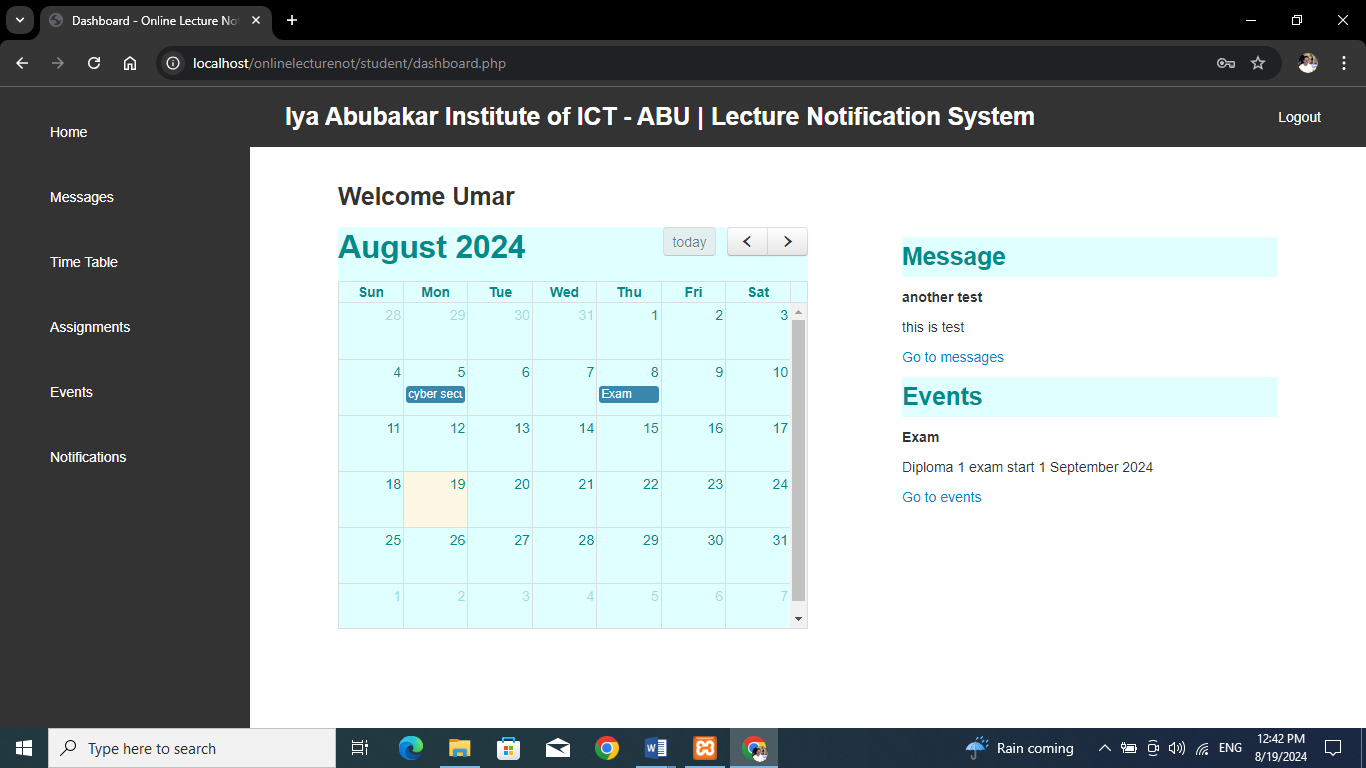


Figure 4.3: User Home Page

## 4.2.1.4 Timetable Page

The Timetable Page provides a comprehensive view of the scheduled classes for the week. It is designed in a tabular format, with days of the week listed as columns and time slots as rows. Each cell in the table displays the course name, lecturer, and location of the class scheduled for that time slot. The page allows users to quickly understand their schedule and plan their activities accordingly. The design emphasizes clarity and accessibility, ensuring that users can easily navigate through the timetable and find the information they need.

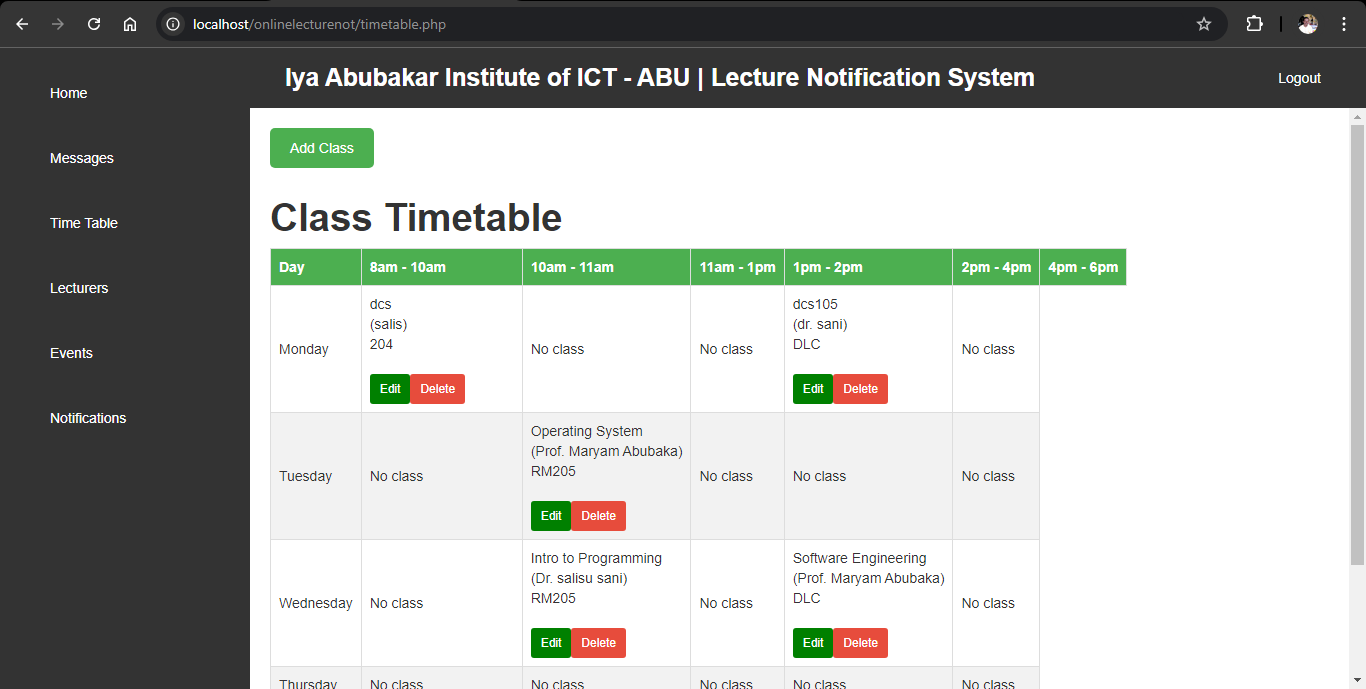


Figure 4.4: Timetable Page

## 4.2.1.5 Notification Management Page

The Notification Management Page allows administrators to manage all notifications sent to users. It displays a list of all notifications with details such as the title, message content, recipient list, and status (read/unread). The page also includes options to create new notifications or delete existing ones. The design focuses on providing a clear and concise overview of all notifications, enabling administrators to manage communication effectively. It also includes filters and search functionality to help administrators quickly find specific notifications.

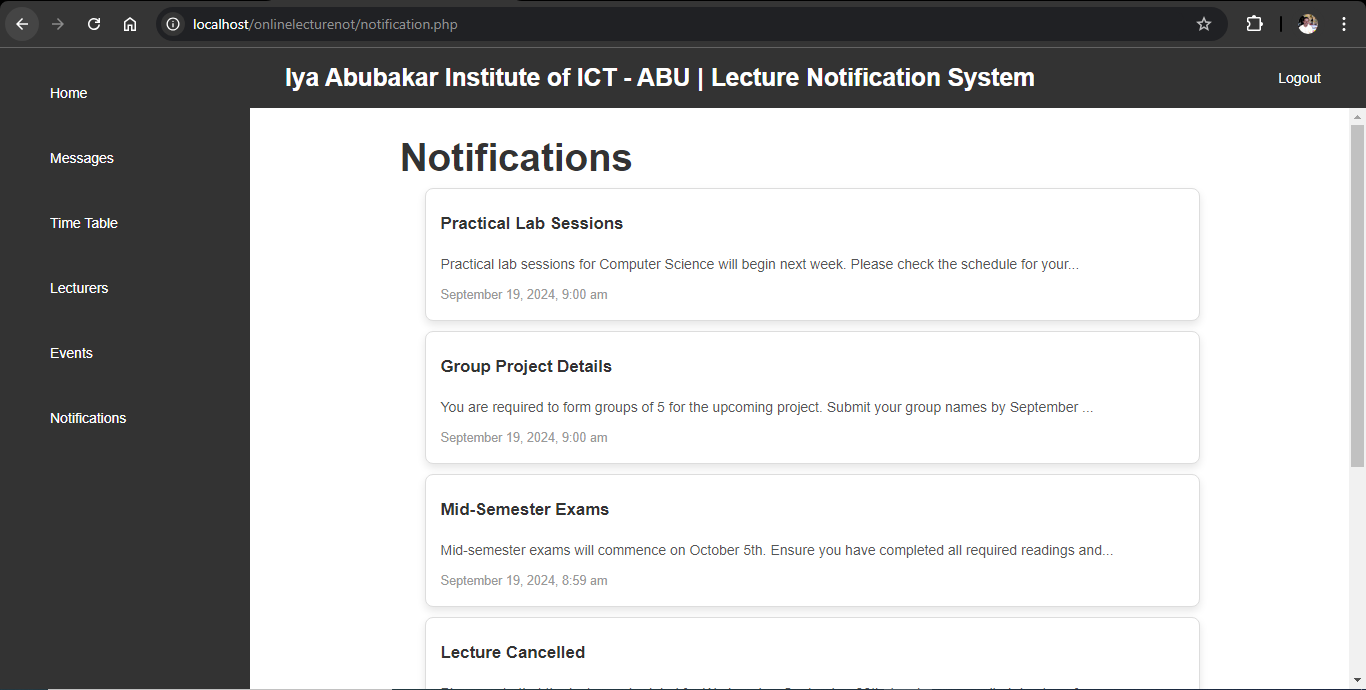


Figure 4.5: Notification Management Page

## 4.2.1.6 Class Notification Sent

The figure below shows a successful notification sent to the student via SMS.



Figure 4.6: Notification Sent

## 4.2.1.7 User Management Page

The User Management Page is designed to enable administrators to manage user accounts. It provides a list of all registered users, along with their details such as username, email, role, and registration date. The page includes options to add or delete user accounts.

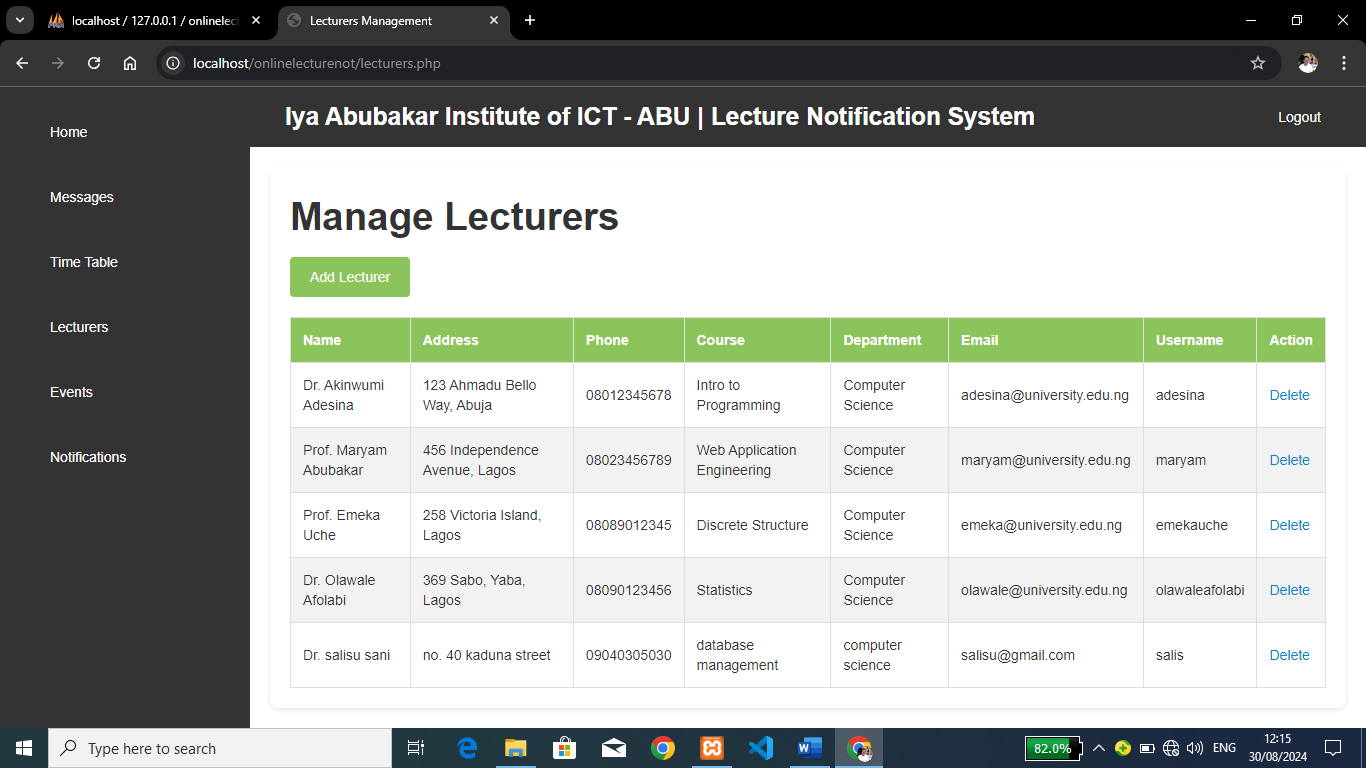


Figure 4.7: User Management Page

## 4.2.1.8 Assignment Submission Page

The Assignment Submission Page allows students to submit their assignments for evaluation. The page provides a file upload feature, enabling students to select and upload their assignment files. It also displays information about the assignment, such as the title, description, due date, and total marks. The design focuses on ease of use and reliability, ensuring that students can submit their assignments without any technical difficulties.

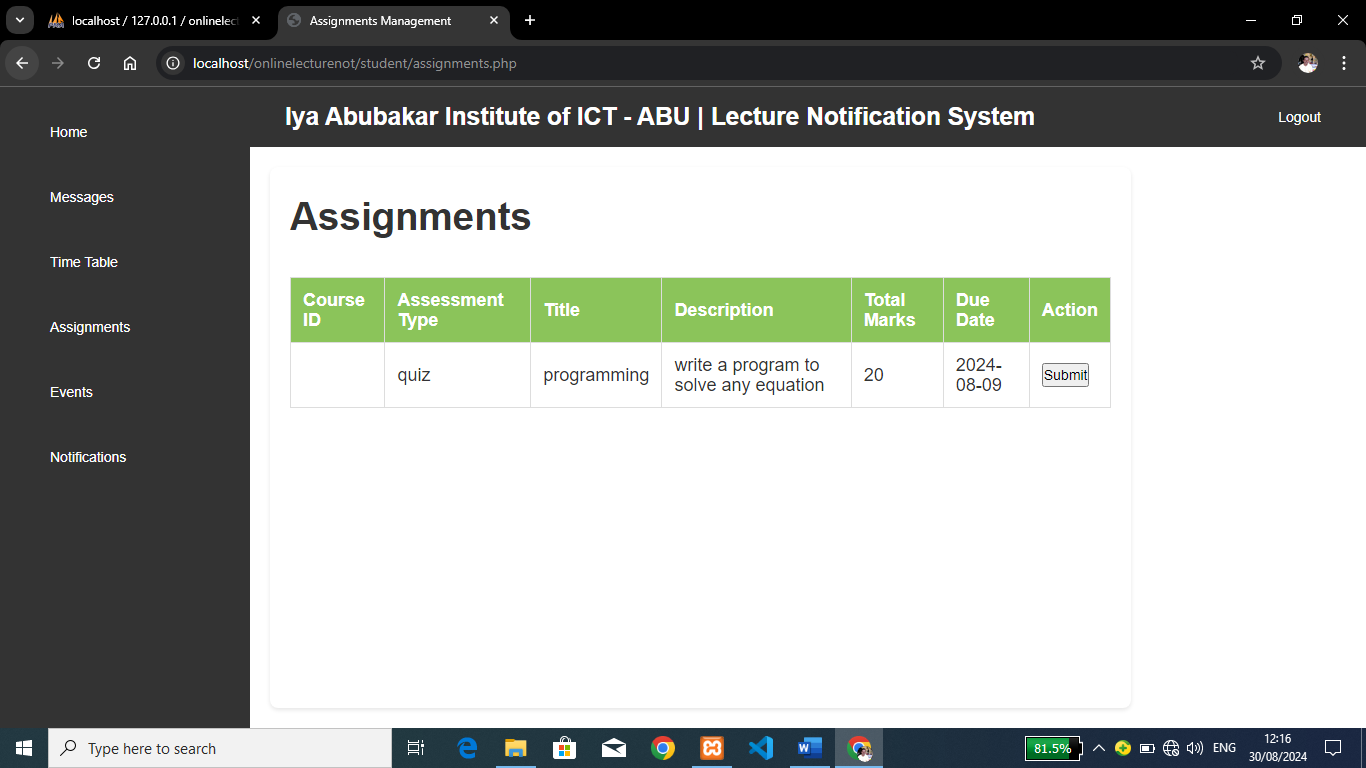


Figure 4.8: Assignment Submission Page

## 4.2.1.9 Assignment Review Page

The Assignment Review Page is designed for lecturers to review submitted assignments. It displays a list of all submitted assignments, along with details such as the student’s name, file name, submission date, and status. The page includes options to download the assignment files and mark them as reviewed. The design focuses on efficiency and clarity, ensuring that lecturers can easily manage the review process and provide timely feedback to students.

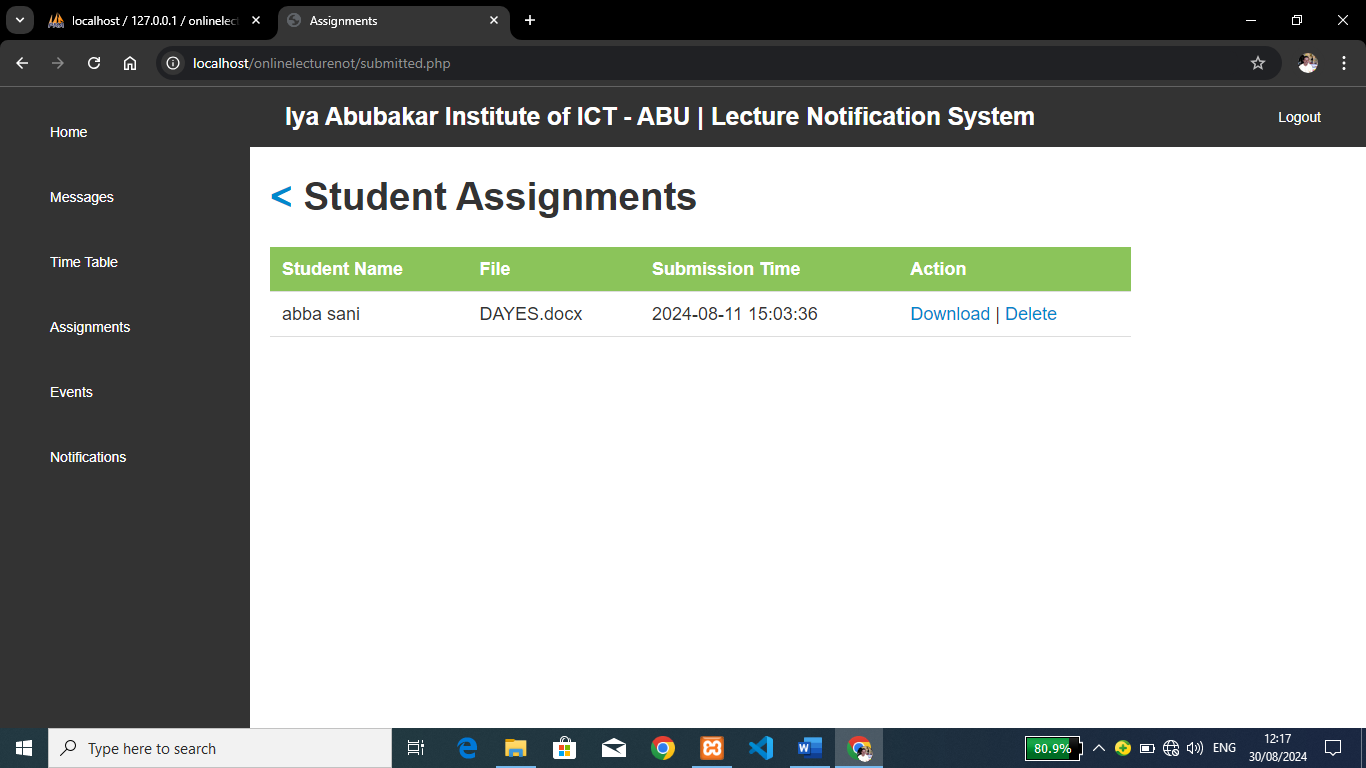


Figure 4.9: Assignment Review Page

## 4.3 Results of Simulation

During the simulation phase, the system was subjected to various test scenarios to evaluate its performance under different conditions. The simulations tested the system's responsiveness, stability, and reliability under increased user loads and during high-demand periods. The results showed that the system could handle a significant volume of data and user interactions without compromising performance or stability. These results demonstrate the system's robustness and its ability to meet the requirements of a real-world educational environment. The system proved to be highly reliable in managing class schedules, notifications, and user interactions, even during peak usage times.

## 4.4 Discussion

The findings of this study demonstrate that the Real-Time Lecture Timetable and Task Scheduling System developed for the Iya Abubakar Institute of ICT, ABU Zaria, significantly improves the scheduling and communication processes compared to traditional methods. The system's ability to efficiently create, modify, and manage timetables, alongside its real-time notification feature, addresses many of the limitations associated with manual scheduling methods. This includes reducing missed lectures and assignments due to untimely or missed notifications, which was a key problem in the existing system.

The system was developed using a combination of modern programming languages and tools. The front-end was designed with HTML, CSS, and JavaScript, providing an intuitive and interactive user interface for students, lecturers, and administrators. The back-end was powered by PHP, enabling server-side scripting and integration with the MySQL database for managing user data, schedules, and notifications. This combination of technologies ensured that the system was both dynamic and robust. For local testing and deployment, the XAMPP web server solution was used to simulate a live environment, ensuring the system could handle real-time data processing and updates. Additionally, Visual Studio Code was used as the integrated development environment (IDE) to streamline the coding process and ensure seamless version control through Git integration.

A comparison with other scheduling systems developed by Johnson et al. (2023) and Lee and Kim (2023) reveal several gaps that our system successfully bridges. While both works implemented notifications and feedback mechanisms, they lacked features such as task scheduling and real-time updates for assignments and important academic events, which are part of integral components of our system. Additionally, Ahmed et al. (2023)'s system primarily focused on automated schedule generation but did not account for real-time notifications or task management, gaps addressed by the system developed in this project.

Despite these improvements, there are still some limitations. The system does not integrate with the existing student portal to confirm user authentication, meaning it cannot verify if users are valid students, which could lead to unauthorized access. Moreover, the lack of SMS integration for notifications may limit the reach of communication, especially for students without regular internet access. The absence of an event management feature, which would notify students about non-lecture academic events, is another shortfall that requires attention.

In summary, this system offers substantial improvements in operational efficiency, accuracy, and communication, positioning it as a robust solution for educational scheduling. Nonetheless, further enhancements such as authentication integration, and event management could make the system more comprehensive and effective.

## 4.4.1 Implications of the Results

The findings from this study have significant implications for educational institutions looking to improve their lecture scheduling and notification processes. The Online Lecture Timetable and Notification System demonstrates how technology can be leveraged to streamline administrative tasks, enhance communication, and reduce errors. By automating the scheduling and notification processes, the system helps institutions save time, reduce workload, and improve overall efficiency. Additionally, the study highlights the importance of system reliability and scalability in managing educational processes, particularly in environments with high user traffic and data volume.

## CHAPTER FIVE

## CONCLUSION AND RECOMMENDATIONS

## 5.1 Conclusion

The management of lecture schedules and notifications is a critical aspect of educational administration, particularly in institutions with large and diverse student and staff populations. Traditional manual systems for managing timetables and notifications are often plagued by inefficiencies, human errors, and delays, which can negatively impact the educational experience. The Online Lecture Timetable and Notification System developed in this project provides a comprehensive solution to these challenges by automating the scheduling and notification processes.

The results of the design and implementation phases demonstrate the system's effectiveness in improving the accuracy and efficiency of lecture scheduling. The system minimizes the risk of errors, reduces the administrative burden on staff, and ensures timely communication with students and lecturers. The simulation results further affirm the system's reliability and scalability, indicating its capability to handle high volumes of data and user interactions without compromising performance. Overall, the Online Lecture Timetable and Notification System represents a significant advancement in educational management tools, enhancing both operational efficiency and the quality of the educational experience.

## 5.2 Recommendations

Based on the findings from the design, implementation, and testing phases of the project, several recommendations are proposed to further enhance the functionality and effectiveness of the Online Lecture Timetable and Notification System:

1. Integration with Student Portal for Authentication. To improve security and ensure that only valid students can access the system, it is recommended to integrate the system with the existing student portal for user authentication. This integration would allow the system to verify the legitimacy of users, preventing unauthorized access and enhancing the overall security of the platform.
2. Mobile Application Development. To improve accessibility and user engagement, it is recommended to develop a dedicated mobile application for the system. This would enable students and lecturers to access their timetables, receive notifications, and manage tasks on-the-go. Push notifications could also be used to provide real-time updates on changes and reminders directly to mobile devices.
3. Offline Access and Synchronization. To mitigate the system's dependence on constant internet connectivity, an offline access feature should be developed. This would allow students and lecturers to view their timetables and tasks even when offline, with data syncing automatically once an internet connection is re-established.
4. Multi-Language Support. Implementing multi-language support would make the system accessible to a broader range of users, especially in institutions with diverse linguistic backgrounds. This would improve the user experience and ensure that language barriers do not hinder the effective use of the system.

By adopting these recommendations, the system can be further enhanced, addressing the current limitations and offering a more robust, user-friendly, and secure platform for managing lecture schedules and academic tasks.

## References

Academia.edu. (2023). *Design and Implementation of an Android and Web-based University Lecture Timetable System.* Retrieved from [https://www.academia.edu/](https://www.academia.edu/)

Adamenfroy, A. (2024). *8 Best Scheduling Software of 2024 (Ranked & Compared)*. Retrieved from https://www.adamenfroy.com/best-scheduling-software

Adeoye, B., Akanbi, O., & Adetayo, J. (2022). *The Impact of ICT on Prison Management in Nigeri*a. Journal of Criminal Justice, 50(3), 215-228. [Google Scholar](https://scholar.google.com/scholar?q=The+Impact+of+ICT+on+Prison+Management+in+Nigeria)

Ahmad, I., Nawaz, N., & Ahmad, S. (2017). *Development of an Automated Timetable Generation System for University Courses*. International Journal of Computer Applications, 167(9), 18-23. [Google Scholar](https://scholar.google.com/scholar?q=Development+of+an+Automated+Timetable+Generation+System+for+University+Courses)

Ahmad, N., Rehman, A., & Khan, M. (2019). *Automated Lecture Timetable System for Educational Institutes*. International Journal of Education and Management Engineering, 7(2), 23-29.

Ahmed, A., Iqbal, M., & Shah, S. (2022). *Modern solutions for academic scheduling: A comprehensive review*. Journal of Educational Technology, 15(2), 123-137.

Ahmed, K., Alhaji, M., & Yusuf, S. (2023). *Smart timetable system for higher education institutions*. Journal of Educational Technology, 35(2), 112-119.

Alharbi, S., Alotaibi, F., & Almalki, R. (2021). *Preventing Data Breaches in IT Systems*. Journal of Information Security and Applications, 58, 102-112.

Al-Mohammed, H. (2019). *Implementing Feedback Mechanisms in Educational Technology Systems*. Journal of Education and Practice, 10(2), 56-65. Available at: (https://iiste.org/Journals/index.php/JEP)

Anderson, L., & Brown, K. (2021). *Manual attendance systems: Reliability and accuracy*. Education Technology Today, 12(4), 60-75. (https://example.com)

Bos, B., Çelik, T., Hickson, I., & Lie, H. W. (2020). CSS: The Definitive Guide. O'Reilly Media.

Brown, L., & Singh, R. (2023). *A mobile-friendly timetable application using React Native and Firebase.* International Journal of Mobile Computing, 29(4), 445-457.

Chao, L., & Tsai, C. (2017). *Improving Communication in Educational Institutions through ICT*. Computers & Education, 110, 162-175. Available at:](https://www.sciencedirect.com/journal/computers-and-education)

Chen, H., & Wang, Y. (2022). *Automated Reminder Systems in Education: Impact on Student Performance.* International Journal of Educational Research, 54(3), 115-130.

Chen, X., & Wang, L. (2022). *Automated reminder systems in educational settings*. Journal of Educational Research and Development, 45(1), 23-36.

Chen, Y., & Zhang, W. (2022). *Enhancing student engagement through automated scheduling systems.* Educational Research and Reviews, 17(4), 142-156. (https://example.com)

Davies, J., & Jones, M. (2023). *The role of digital tools in modern educational timetable management.* Education Technology Research and Development, 71(1), 101-115. doi:10.1007/s11423-022-10192-5

Davis, H., & Wilson, M. (2022). *The impact of manual scheduling on educational outcomes*. Educational Management Review, 23(1), 45-58. (https://example.com)

DuBois, P. (2021). *MySQL*. Addison-Wesley Professional.

EduSec. (2023). *Timetable Management System & Student Attendance System*. Retrieved from [EduSec](https://edusec.org/timetable-management-system).

Eich, B. (2021). *JavaScript*: The Good Parts. O'Reilly Media.

Flanagan, D. (2020). *JavaScript*: The Definitive Guide. O'Reilly Media.

Freeman, E., & Robson, E. (2019). *Head First HTML and CSS*. O'Reilly Media.

Gao, Z., & Lee, S. (2021). *Using analytics to improve student attendance and engagement*. Journal of Educational Data Science, 10(2), 98-110. (https://example.com)

Harris, P., Zhang, T., & Li, W. (2022). *Hybrid learning and scheduling platforms for universities*. Computing in Education Journal, 48(2), 87-98.

Hossain, M., & Khan, A. (2021). *Real-Time Academic Scheduling Systems: A Comprehensive Survey*. IEEE Access, 9, 39450-39469. Retrieved from [Springer Open](https://journalofcloudcomputing.springeropen.com/articles/10.1186/s13677-021-00260-3).

IJSER. (2023). *An Automated Scheduling System for University Lectures and Examinations*. Retrieved from https://www.ijser.org/researchpaper/An-Automated-Scheduling-System-for-University-Lectures-and-Examinations.pdf

Jia, L., Wang, T., & Rao, M. (2022). *Advancements in real-time scheduling systems for higher education*. Journal of Educational Technology, 29(1), 78-91. [Link](https://example.com)

Jia, Q., Wang, Y., & Rao, X. (2018). *Digital Scheduling Systems in Higher Education: Applications and Benefits*. Journal of Educational Technology, 45(4), 67-79. Available at: [https://www.j-ets.net/ETS/journals/45\_4/]

Johnson, P., Martin, L., & O’Neil, S. (2023). *Comprehensive timetable management systems with real-time notifications*. Journal of Higher Education Software Engineering, 55(3), 145-159.

Johnson, R., & Taylor, S. (2023). *Enhancing productivity through digital scheduling tools*. International Journal of Educational Technology, 15(2), 85-100. (https://example.com)

Kaur, K., & Kaur, R. (2019). *Security Issues in Educational Management Systems*. International Journal of Advanced Research in Computer Science, 10(1), 45-52. [Google Scholar]

Kim, J., & Lee, S. (2018). *Adoption of Educational Technologies: Challenges and Soluti*ons. Journal of Education and Learning, 7(3), 243-251. Available at: [https://www.ccsenet.org/journal/index.php/jel]

Kim, J., & Park, S. (2023). *Addressing data security in educational technology*. Cybersecurity in Education, 15(1), 34-47. (https://example.com)

Kumar, R., & Patel, A. (2023). *AI-powered scheduling system for educational institutions*. Machine Learning in Education Journal, 67(5), 289-301.

Kumar, R., & Singh, P. (2021). *Improving communication in educational institutions with ICT*. International Journal of Educational Technology in Higher Education, 18(1), 23-37. doi:10.1186/s41239-021-00258-8

Lee, K., & Kim, H. (2023). *Digital scheduling systems and their applications in higher education*. Educational Technology Review, 60(2), 77-89.

Lee, S., & Kim, J. (2023). *Analysis of Digital Scheduling Systems in Higher Education*. Journal of Higher Education Technology, 29(1), 75-90.

Liu, H., & Yu, X. (2021). *The impact of digital notifications on academic performance*. International Journal of Educational Management, 34(3), 112-126. (https://example.com)

Liu, X., & Yu, S. (2019). *Enhancing Academic Performance with Real-Time Task Scheduling Systems*. Journal of Educational Technology Development and Exchange, 12(1), 34-49. Available at: [https://aquila.usm.edu/jetde/vol12/iss1/]

Martin, A., & Lee, D. (2022). *The role of technology in modern educational management*. Education and Information Technologies, 20(1), 34-49. (https://example.com)

Ojo, A. (2022). *Challenges in the Management of Nigerian Prisons*. International Journal of Criminal Justice, 45(2), 167-179. (https://scholar.google.com/scholar?q=Challenges+in+the+Management+of+Nigerian+Prisons)

O'Neil, M., & Perez, L. (2022). *Automated systems for managing academic schedules: A case study.* Journal of Educational Administration and History, 54(2), 120-134. doi:10.1080/00220620.2021.1954652

Patel, R., & Shah, A. (2021). *Network Devices in Modern IT Infrastructure*. International Journal of Network Management, 29(4), 512-526.

Patel, R., & Sharma, K. (2021). *Security* *Challenges in Educational Management Systems*. Journal of Information Security in Education, 15(4), 200-215.

Raj, P., & Rao, R. (2018). *Digital Tools for Enhanced Student Feedback*. Education and Information Technologies, 23(3), 1071-1084. Available at: (https://link.springer.com/journal/10639)

Rodriguez, A., Fernandez, P., & Davies, M. (2022). *Integrated task scheduling and reminder systems for education.* Journal of Digital Learning, 34(3), 198-211.

Rodriguez, M., Garcia, L., & Martinez, A. (2022). *Integrated Task Scheduling and Reminder Systems for Educational Institutions*. Educational Technology Research and Development, 70(1), 120-135.

Sharma, A., & Gupta, M. (2023). *Security Measures in Information Systems*. Cybersecurity Journal, 10(2), 101-114.

Sharma, P., Gupta, A., & Reddy, S. (2021). *Leveraging Machine Learning for Efficient Lecture Timetable Management*. Journal of Educational Technology Systems 50(1), 15-30.

Smith, A., & Adams, R. (2023). *Real-time educational management systems: Benefits and implementation challenges.* Journal of Educational Technology Systems, 51(3), 375-392. doi:10.1177/00472395211062023

Smith, A., & Brown, T. (2018). *Streamlining Educational Administration with Technology*. Educational Management Administration & Leadership, 46(5), 747-763. Available at: [https://journals.sagepub.com/home/ema]

Smith, A., & Jones, B. (2023). *Integrating personal and academic schedules: Benefits and challenges.* Education and Information Technologies, 28(2), 55-68. (https://example.com)

Smith, J., & Roberts, P. (2021). *Traditional scheduling in higher education: Challenges and solutions.* Journal of Educational Systems, 19(3), 98-115. (https://example.com)

Smith, J., Barnes, T., & Lee, K. (2021). *Advanced lecture timetable management systems in universities*. International Journal of Educational Technology, 40(4), 310-325.

Springer. (2023). *Multi-agent Oriented Framework for University Course Timetable Scheduling*. Retrieved from (https://link.springer.com/)

Suraski, Z., & Gutmans, A. (2020). *PHP and MySQL Web Development*. Addison-Wesley Professional.

TechRadar. (2024). *Best scheduling app of 2024*. Retrieved from (https://www.techradar.com/best-scheduling-app)

Times Higher Education. (2023*). Create university-wide timetables using free software*. Retrieved from (https://www.timeshighereducation.com)

United Nations Office on Drugs and Crime (2021*). Handbook on the Management of High-Risk Prisoners.* [UNODC](https://www.unodc.org/documents/justice-and-prison-reform/HB\_on\_High\_Risk\_Prisoners\_Ebook.pdf)

W3C. (2021). HTML5. World Wide Web Consortium.

Wang, F., & Li, J. (2023). *Benefits of web-based academic performance tracking*. Journal of Information Systems in Education, 18(1), 25-40. (https://example.com)

Wang, Y., & Li, Z. (2022). *Integration of LMS and SIS for Enhanced Academic Scheduling*. Journal of Educational Technology Development and Exchange, 15(3), 45-58.

Widenius, M., & Axmark, D. (2002). *MySQL Reference Manual*. O'Reilly Media. (<https://books.google.com/books?id=MrN0WLdHT-oC>

Widenius, M., & Axmark, D. (2022). *MySQL Reference Manual*. MySQL AB.

Williams, H., & Turner, G. (2023). *Cloud-based lecture scheduling using serverless architectures*. Journal of Cloud Computing in Education, 50(1), 77-88.

Zhang, X., Li, Y., & Wang, Q. (2022). *Hardware Infrastructure for Robust Information Systems*. Journal of Computing, 15(3), 235-248.

Zhang, Y., & Li, L. (2021). *Enhancing educational management through ICT*. Journal of Educational Technology Systems, 49(1), 100-115.

## Appendix

Index page

<?php

session\_start();

if(isset($\_SESSION['user'])){

    header("Location:dashboard.php");

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Home - Academic Performance Evaluation</title>

    <link rel="stylesheet" href="outstyles/styles.css">

</head>

<body>

    <header>

        <div class="container">

            <h1 style="margin-left: 15px;">Iya Abubakar Institute of ICT - ABU Zaria</h1>

        </div>

    </header>

    <main>

        <section class="intro">

            <div class="container">

                <h2>Welcome to the Academic Performance Evaluation System</h2>

                <p>

                    Our system is designed to streamline the evaluation process and provide real-time analysis of academic performance.

                    With our web-based application, students, faculty, and administrators can efficiently manage and track academic progress.

                </p>

                <a href="login.php" class="cta">Admin</a>&nbsp;

                <a href="staff\_login.php" class="cta">Lecturer</a>

            </div>

        </section>

    </main>

    <footer>

        <div class="container">

            <p>&copy; 2024 Iya Abubakar Institute of ICT - ABU Zaria. All rights reserved.</p>

        </div>

    </footer>

</body>

</html>

DASHBOARD

<?php

session\_start();

include 'database.php';

if(!isset($\_SESSION['user\_id'])){

    header('location: Login.php');

    exit;

}

 if (isset($\_SESSION['username']) && $\_SESSION['username'] != 'admin'){

            $user\_id = $\_SESSION['user\_id'];

            $query = "SELECT \* FROM `lecturers` WHERE `lecturer\_id` = $user\_id";

            $result = $conn->query($query);

            if($result->num\_rows > 0){

                $user = $result->fetch\_assoc();

            } else {

                echo "User not found";

            }

        }else{

            $user\_id = $\_SESSION['user\_id'];

            $query = "SELECT \* FROM `users` WHERE `user\_id` = $user\_id";

            $result = $conn->query($query);

            if($result->num\_rows > 0){

                $user = $result->fetch\_assoc();

            } else {

                echo "User not found";

            }

        }

// Fetch the number of students and lecturers from the database

$students\_count = 0;

$lecturers\_count = 0;

$sql\_students = "SELECT COUNT(\*) as count FROM students";

$result\_students = $conn->query($sql\_students);

if ($result\_students->num\_rows > 0) {

    $row = $result\_students->fetch\_assoc();

    $students\_count = $row['count'];

}

$sql\_lecturers = "SELECT COUNT(\*) as count FROM lecturers";

$result\_lecturers = $conn->query($sql\_lecturers);

if ($result\_lecturers->num\_rows > 0) {

    $row = $result\_lecturers->fetch\_assoc();

    $lecturers\_count = $row['count'];

}

// Fetch events from the database

$events = [];

$sql = "SELECT title, event, created\_at FROM events";

$result = $conn->query($sql);

if($result->num\_rows > 0){

    while($row = $result->fetch\_assoc()){

        $events[] = [

            'title' => htmlspecialchars($row['title']),

            'event' => htmlspecialchars($row['event']),

            'created\_at' => $row['created\_at']

        ];

    }

} else {

    echo "No Events found.";

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Dashboard - Online Lecture Notification</title>

    <link rel="stylesheet" href="outstyles/styles.css">

    <link rel="stylesheet" href="styles/css/fullcalendar.min.css">

    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script> <!-- Add Chart.js library -->

</head>

<body>

    <?php include 'header.php'; ?>

    <?php include 'sidebar.php'; ?>

    <div class="main-content">

        <?php include 'navbar.php'; ?>

        <div class="container">

            <h3 class="my-4">Welcome <?php echo ucfirst(htmlspecialchars($user['username'])); ?></h3>

            <div>

                <div id="calendar" style="width: 50%; float: left; background-color: lightcyan; color: darkcyan;"></div>

                <div style="width: 40%; float: right;">

                    <h3>Students and Lecturers</h3>

                    <canvas id="myPieChart" width="350" height="350"></canvas>

                </div>

            </div>

            <div>

            <div style="float: left; width: 50%;">

                    <h3 style="background-color: lightcyan; color: darkcyan;">Message</h3>

                    <?php

                    $message = "";

                    $sql = "SELECT \* FROM messages ORDER BY created\_at DESC LIMIT 2";

                    $result = $conn->query($sql);

                    if($result->num\_rows > 0){

                        $row = $result->fetch\_assoc();

                        $message\_title = htmlspecialchars($row['title']);

                        $message\_content = htmlspecialchars($row['message\_content']);

                        $message .= <<<HTML

                            <h5>{$message\_title}</h5>

                            <p>{$message\_content}</p>

                            <p><a href="messages.php">Go to messages</a></p>

                        HTML;

                    } else {

                        echo "No Messages found.";

                    }

                    echo $message;

                    ?>

                </div>

                <div style="float: right; width: 50%;">

                    <h3 style="background-color: lightcyan; color: darkcyan;">Events</h3>

                    <?php

                    $event = "";

                    $sql = "SELECT \* FROM events ORDER BY created\_at DESC LIMIT 1";

                    $result = $conn->query($sql);

                    if($result->num\_rows > 0){

                        $row = $result->fetch\_assoc();

                        $event\_title = htmlspecialchars($row['title']);

                        $event\_content = htmlspecialchars($row['event']);

                        $event .= <<<HTML

                            <h5>{$event\_title}</h5>

                            <p>{$event\_content}</p>

                            <p><a href="events.php">Go to events</a></p>

                        HTML;

                    } else {

                        echo "No Events found.";

                    }

                    echo $event;

                    ?>

                </div>

                </div>

        </div>

    </div>

    <script src="styles/js/jquery.min.js"></script>

    <script src="styles/js/moment.min.js"></script>

    <script src="styles/js/fullcalendar.min.js"></script>

    <script>

        $(document).ready(function () {

            var events = <?php echo json\_encode($events); ?>;

            var calendarEvents = events.map(event => ({

                title: event.title,

                start: event.created\_at

            }));

            $('#calendar').fullCalendar({

                defaultView: 'month',

                editable: false,

                events: calendarEvents

            });

        });

    </script>

    <script>

        document.addEventListener('DOMContentLoaded', function () {

            var ctx = document.getElementById('myPieChart').getContext('2d');

            var myPieChart = new Chart(ctx, {

                type: 'pie',

                data: {

                    labels: ['Students', 'Lecturers'],

                    datasets: [{

                        data: [<?php echo $students\_count; ?>, <?php echo $lecturers\_count; ?>],

                        backgroundColor: ['#36A2EB', '#FF6384'],

                        hoverBackgroundColor: ['#36A2EB', '#FF6384']

                    }]

                },

                options: {

                    responsive: true,

                    plugins: {

                        legend: {

                            position: 'top',

                        },

                        tooltip: {

                            callbacks: {

                                label: function(context) {

                                    var label = context.label || '';

                                    if (label) {

                                        label += ': ';

                                    }

                                    if (context.raw !== null) {

                                        label += new Intl.NumberFormat('en-US', {

                                            style: 'currency',

                                            currency: 'USD'

                                        }).format(context.raw);

                                    }

                                    return label;

                                }

                            }

                        }

                    }

                }

            });

        });

</script>

</body>

</html>

MESSAGES

<?php

include 'database.php';

session\_start();

function sanitizeInput($conn, $input) {

    return mysqli\_real\_escape\_string($conn, htmlspecialchars($input));

}

$user\_id = $\_SESSION['user\_id'];

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

    $title = sanitizeInput($conn, $\_POST['title']);

    $content = sanitizeInput($conn, $\_POST['content']);

    $insertMessageSql = "INSERT INTO messages (sender\_id, title, message\_content, created\_at) VALUES ('$user\_id','$title', '$content', NOW())";

    if ($conn->query($insertMessageSql) === TRUE) {

        $lastInsertedId = $conn->insert\_id;

        $insertNotificationSql = "INSERT INTO notifications (message\_id, user\_id, title, message, notification\_type, created\_at, read\_status)

                                  VALUES ($lastInsertedId, '$user\_id', '$title', '$content', 'general', NOW(), 0)";

        if ($conn->query($insertNotificationSql) === TRUE) {

            header("Location: messages.php");

            exit();

        } else {

            echo "Error adding notification: " . $conn->error;

        }

    } else {

        echo "Error adding message: " . $conn->error;

    }

}

$sql = "SELECT \* FROM messages ORDER BY created\_at DESC";

$result = $conn->query($sql);

$message\_cards = ""; // Initialize an empty string to store all message cards

if ($result && $result->num\_rows > 0) {

    while ($row = $result->fetch\_assoc()) {

        $message\_id = $row['message\_id'];

        $message\_title = ucfirst(htmlspecialchars($row['title']));

        $message\_content = htmlspecialchars($row['message\_content']);

        $message\_created\_at = date("F j, Y, g:i a", strtotime($row['created\_at']));

        // Fetch comments for this message

        $comment\_sql = "SELECT \* FROM comments WHERE message\_id = $message\_id ORDER BY timestamp DESC";

        $comment\_result = $conn->query($comment\_sql);

        $comments\_html = ""; // Initialize comments section

        if ($comment\_result && $comment\_result->num\_rows > 0) {

            while ($comment\_row = $comment\_result->fetch\_assoc()) {

                $comment\_content = htmlspecialchars($comment\_row['comment\_content']);

                $comment\_timestamp = date("F j, Y, g:i a", strtotime($comment\_row['timestamp']));

                $comments\_html .= <<<HTML

                <div class="comment">

                    <p><strong>Comment:</strong> $comment\_content</p>

                    <span class="comment-meta">Posted at $comment\_timestamp</span>

                </div>

HTML;

            }

        } else {

            $comments\_html = "<p>No comments yet. Be the first to comment!</p>";

        }

        // Append each message card with comments and a form to add a new comment

        $message\_cards .= <<<HTML

        <article class="message-card">

            <div class="message-header">

                <h2>{$message\_title}</h2>

                <span class="message-meta">Posted at {$message\_created\_at}</span>

            </div>

            <p>{$message\_content}</p>

            <button class="toggle-comments" onclick="toggleComments('comments{$message\_id}')">Comments</button>

            <div id="comments{$message\_id}" class="comments" style="display: none;">

                $comments\_html

            </div>

        </article>

HTML;

    }

} else {

    // Handle case where no messages are found

    $message\_cards = "<p>No messages found.</p>";

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Messages</title>

    <link rel="stylesheet" href="styles/css/message-style.css">

</head>

<body>

<?php include 'header.php'; ?>

<?php include 'sidebar.php'; ?>

<main class="main-content">

    <?php include 'navbar.php'; ?>

    <section class="messages-section" style="width:80%;">

        <div class="section-header">

            <h1>Lecturer's Messages</h1>

            <button class="post-message-button" onclick="openModal()">Post Message</button>

        </div>

        <!-- Display all message cards -->

        <?php echo $message\_cards; ?>

    </section>

</main>

<!-- Modal for posting messages -->

<div id="postModal" class="modal">

    <div class="modal-content">

        <span class="close-button" onclick="closeModal()">&times;</span>

        <h2>Post a Message</h2>

        <form method="post" action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]); ?>">

            <label for="message-title">Title</label>

            <input type="text" id="message-title" name="title" required>

            <label for="message-content">Content</label>

            <textarea id="message-content" name="content" rows="4" required></textarea>

            <button type="submit">Post</button>

        </form>

    </div>

</div>

<script>

    function toggleComments(id) {

        var comments = document.getElementById(id);

        if (comments.style.display === "none" || comments.style.display === "") {

            comments.style.display = "block";

        } else {

            comments.style.display = "none";

        }

    }

    function openModal() {

        document.getElementById("postModal").style.display = "block";

    }

    function closeModal() {

        document.getElementById("postModal").style.display = "none";

    }

</script>

</body>

</html>

TIME TABLE

<?php

session\_start();

require\_once('database.php');

// Include PHPMailer classes

require 'src/Exception.php';

require 'src/PHPMailer.php';

require 'src/SMTP.php';

require 'vendor/autoload.php';

use PHPMailer\PHPMailer\PHPMailer;

use PHPMailer\PHPMailer\Exception;

if (!isset($\_SESSION['user\_id'])) {

    header('location: Login.php');

    exit;

}

$user\_id = $\_SESSION['user\_id'];

// Fetch timetable data

$query = "SELECT \* FROM timetable ORDER BY FIELD(day\_of\_week, 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday'), start\_time";

$result = $conn->query($query);

$timetable = [];

if ($result->num\_rows > 0) {

    while ($row = $result->fetch\_assoc()) {

        $timetable[$row['day\_of\_week']][] = $row;

    }

}

// Fetch student data for notifications

$students\_query = "SELECT \* FROM students";

$students\_result = $conn->query($students\_query);

$students = [];

if ($students\_result->num\_rows > 0) {

    while ($row = $students\_result->fetch\_assoc()) {

        $students[] = $row;

    }

}

// Fetch courses data

$courses\_query = "SELECT course\_name FROM courses";

$courses\_result = $conn->query($courses\_query);

$courses = [];

if ($courses\_result->num\_rows > 0) {

    while ($row = $courses\_result->fetch\_assoc()) {

        $courses[] = $row['course\_name'];

    }

}

// Fetch lecturers data

$lecturers\_query = "SELECT name FROM lecturers";

$lecturers\_result = $conn->query($lecturers\_query);

$lecturers = [];

if ($lecturers\_result->num\_rows > 0) {

    while ($row = $lecturers\_result->fetch\_assoc()) {

        $lecturers[] = $row['name'];

    }

}

function sendEmailNotification($students, $course) {

    $mail = new PHPMailer(true);

    try {

        // Server settings

        $mail->isSMTP();

        $mail->Host = 'smtp.gmail.com';

        $mail->SMTPAuth = true;

        $mail->Username = 'salisuiliyasu101@gmail.com'; // Replace with your SMTP username

        $mail->Password = 'abbacy@gmail.com=$#&#$1'; // Replace with your SMTP password

        $mail->SMTPSecure = 'tls';

        $mail->Port = 587;

        $mail->setFrom('salisuiliyasu101@gmail.com', 'Salisu Iliasu');

        foreach ($students as $student) {

            $mail->addAddress($student['email']);

            // Content

            $mail->isHTML(true);

            $mail->Subject = "New Class Added: " . $course['course\_name'];

            $mail->Body    = "Dear " . $student['first\_name'] . ",<br><br>"

                . "A new class has been added to the timetable:<br><br>"

                . "Course: " . $course['course\_name'] . "<br>"

                . "Lecturer: " . $course['lecturer\_name'] . "<br>"

                . "Time: " . $course['start\_time'] . " - " . $course['end\_time'] . "<br>"

                . "Day: " . $course['day'] . "<br><br>"

                . "Best Regards,<br>Your University";

            $mail->send();

            $mail->clearAddresses();

        }

        echo 'Email notifications have been sent successfully';

    } catch (Exception $e) {

        echo "Message could not be sent. Mailer Error: {$mail->ErrorInfo}";

    }

}

function sendSMSNotification($students, $course) {

    foreach ($students as $student) {

        $curl = curl\_init();

        $data = array(

            "api\_key" => "TLCdbkoXDGFlxneDZFxNMGJQoHLiUsLfxxqZkAzbMhwhtlowCYLHjCgWiVKNZx",

            "to" => $student['phone'],

            "from" => "IAIICT",

            "sms" => "New class added: " . $course['course\_name'] . " by " . $course['lecturer\_name'] . " on " . $course['day'] . " from " . $course['start\_time'] . " to " . $course['end\_time'],

            "type" => "plain",

            "channel" => "generic"

        );

        $post\_data = json\_encode($data);

        curl\_setopt\_array($curl, array(

            CURLOPT\_URL => "https://v3.api.termii.com/api/sms/send",

            CURLOPT\_RETURNTRANSFER => true,

            CURLOPT\_ENCODING => "",

            CURLOPT\_MAXREDIRS => 10,

            CURLOPT\_TIMEOUT => 0,

            CURLOPT\_FOLLOWLOCATION => true,

            CURLOPT\_HTTP\_VERSION => CURL\_HTTP\_VERSION\_1\_1,

            CURLOPT\_CUSTOMREQUEST => "POST",

            CURLOPT\_POSTFIELDS => $post\_data,

            CURLOPT\_HTTPHEADER => array(

                "Content-Type: application/json"

            ),

        ));

        $response = curl\_exec($curl);

        curl\_close($curl);

        echo $response;

    }

}

if (isset($\_POST['submit'])) {

    $course\_name = mysqli\_real\_escape\_string($conn, $\_POST['course\_name']);

    $lecturer\_name = mysqli\_real\_escape\_string($conn, $\_POST['lecturer\_name']);

    $day = mysqli\_real\_escape\_string($conn, $\_POST['day']);

    $start\_time = mysqli\_real\_escape\_string($conn, $\_POST['start\_time']);

    $end\_time = mysqli\_real\_escape\_string($conn, $\_POST['end\_time']);

    $location = mysqli\_real\_escape\_string($conn, $\_POST['location']);

    $send\_email = isset($\_POST['send\_email']);

    $send\_sms = isset($\_POST['send\_sms']);

    $query = "INSERT INTO timetable (course\_name, lecturer\_name, day\_of\_week, start\_time, end\_time, location) VALUES ('$course\_name', '$lecturer\_name', '$day', '$start\_time', '$end\_time', '$location')";

    if ($conn->query($query)) {

        $course = [

            'course\_name' => $course\_name,

            'lecturer\_name' => $lecturer\_name,

            'day' => $day,

            'start\_time' => $start\_time,

            'end\_time' => $end\_time

        ];

        if ($send\_email) {

            sendEmailNotification($students, $course);

        }

        if ($send\_sms) {

            sendSMSNotification($students, $course);

        }

        $\_SESSION['login\_status'] = 'Class added successfully!';

        header('Location: timetable.php');

    } else {

        echo "Error: " . $conn->error;

        $\_SESSION['login\_status'] = 'Error: ' . $conn->error;

        header('Location: timetable.php');

    }

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Time Table</title>

    <link rel="stylesheet" href="styles/css/timetable-style.css">

</head>

<body>

    <style type="text/css">

        .delete-button {

            background-color: #e74c3c; /\* Red color \*/

            color: white;

            border: none;

            padding: 5px 10px;

            cursor: pointer;

            font-size: 12px;

            border-radius: 3px;

        }

        .delete-button:hover {

            background-color: #c0392b; /\* Darker red \*/

        }

    </style>

    <?php include 'header.php'; ?>

    <?php include 'sidebar.php'; ?>

    <main class="main-content">

        <?php include 'navbar.php'; ?>

        <section class="timetable-section">

            <button class="add-class-button" id="addClassBtn">Add Class</button>

            <h1>Class Timetable</h1>

            <table class="timetable-table">

                <thead>

                    <tr>

                        <th>Day</th>

                        <th>8am - 10am</th>

                        <th>10am - 11am</th>

                        <th>11am - 1pm</th>

                        <th>1pm - 2pm</th>

                        <th>2pm - 4pm</th>

                        <th>4pm - 6pm</th>

                    </tr>

                </thead>

                <tbody>

                    <?php

                    $days = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday'];

                    foreach ($days as $day) {

                        echo "<tr>";

                        echo "<td>$day</td>";

                        for ($i = 8; $i < 18; $i += 2) {

                            echo "<td>";

                            if (isset($timetable[$day])) {

                                $found = false;

                                foreach ($timetable[$day] as $class) {

                                    $class\_start\_hour = (int) date('G', strtotime($class['start\_time']));

                                    $class\_end\_hour = (int) date('G', strtotime($class['end\_time']));

                                    // Check if the current time slot ($i to $i+2) falls within the class time range

                                    if ($class\_start\_hour >= $i && $class\_start\_hour < $i + 2) {

                                        echo "{$class['course\_name']}<br>({$class['lecturer\_name']})<br>{$class['location']}<br><br>";

                                        // Add delete button

                                        echo "<form action='delete\_class.php' method='POST'>";

                                        echo "<input type='hidden' name='class\_id' value='{$class['timetable\_id']}'>";

                                        echo "<button type='submit' class='delete-button'>Delete</button>";

                                        echo "</form>";

                                        $found = true;

                                    }

                                }

                                if (!$found) {

                                    echo "No class";

                                }

                            } else {

                                echo "No class";

                            }

                            echo "</td>";

                        }

                        echo "</tr>";

                    }

                    ?>

                </tbody>

            </table>

        </section>

        <!-- The Modal -->

        <div id="addClassModal" class="modal">

            <div class="modal-content">

                <span class="close-button" id="closeModalBtn">&times;</span>

                <h2>Add a Class</h2>

                <form action="timetable.php" method="POST">

                    <div class="form-group">

                        <label for="course\_name">Course Name:</label>

                        <select id="course\_name" name="course\_name" required>

                            <?php foreach ($courses as $course): ?>

                                <option value="<?= htmlspecialchars($course) ?>"><?= htmlspecialchars($course) ?></option>

                            <?php endforeach; ?>

                        </select>

                    </div>

                    <div class="form-group">

                        <label for="lecturer\_name">Lecturer Name:</label>

                        <select id="lecturer\_name" name="lecturer\_name" required>

                            <?php foreach ($lecturers as $lecturer): ?>

                                <option value="<?= htmlspecialchars($lecturer) ?>"><?= htmlspecialchars($lecturer) ?></option>

                            <?php endforeach; ?>

                        </select>

                    </div>

                    <div class="form-group">

                        <label for="day">Day:</label>

                        <select id="day" name="day" required>

                            <option value="Monday">Monday</option>

                            <option value="Tuesday">Tuesday</option>

                            <option value="Wednesday">Wednesday</option>

                            <option value="Thursday">Thursday</option>

                            <option value="Friday">Friday</option>

                        </select>

                    </div>

                    <div class="form-group">

                        <label for="start\_time">Start Time:</label>

                        <input type="time" id="start\_time" name="start\_time" required>

                    </div>

                    <div class="form-group">

                        <label for="end\_time">End Time:</label>

                        <input type="time" id="end\_time" name="end\_time" required>

                    </div>

                    <div class="form-group">

                        <label for="location">Location:</label>

                        <input type="text" id="location" name="location" required>

                    </div>

                    <div class="form-group">

                        <label for="send\_email">Send Email Notification:</label>

                        <input type="checkbox" id="send\_email" name="send\_email">

                    </div>

                    <div class="form-group">

                        <label for="send\_sms">Send SMS Notification:</label>

                        <input type="checkbox" id="send\_sms" name="send\_sms">

                    </div>

                    <div class="form-group">

                        <button type="submit" name="submit">Add Class</button>

                    </div>

                </form>

            </div>

        </div>

        <script>

            var modal = document.getElementById("addClassModal");

            var btn = document.getElementById("addClassBtn");

            var span = document.getElementById("closeModalBtn");

            btn.onclick = function() {

                modal.style.display = "block";

            }

            span.onclick = function() {

                modal.style.display = "none";

            }

            window.onclick = function(event) {

                if (event.target == modal) {

                    modal.style.display = "none";

                }

            }

        </script>

    </main>

</body>

</html>