DESIGN AND IMPLEMENTATION OF A LECTURE SCHEDULE AND ALERT NOTIFICATION SYSTEM (CASE STUDY OF COMPUTER SCIENCE DEPARTMENT)

BY

AYUBA BELLO
(ST/CS/ND/21/032)

DEPARTMENT OF COMPUTER SCIENCE,
SCHOOL OF SCIENCE AND TECHNOLOGY,
FEDERAL POLYTECHNIC, MUBI, ADAMAWA STATE.

IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF NATIONAL DIPLOMA (ND) IN COMPUTER SCIENCE.

SEPTEMBER, 2023

DECLARATION

I hereby declare that the work in this project titled "**Design and Implementation of a Lecture Schedule and Alert Notification System (Case Study of Computer Science Department)"** was performed by me under the supervision of Mal. Umar Bello. The information derived from literatures has been duly acknowledged in the text and a list of references provided. The work embodied in this project is original and had not been submitted in part or in full for any other diploma or certificate of this or any other institution.

AYUBA BELLO _____ _____

(ST/CS/ND/21/032) Signature Date

CERTIFICATION

This project titled "**Design and Implementation of a Lecture Schedule and Alert Notification System (Case Study of Computer Science Department)"** meets the regulations governing the award of National Diploma (ND) in Computer Science, Federal Polytechnic Mubi, Adamawa State

Mal. Umar Bello _____
(Project Supervisor) Sign/Date

Mr. Mustapha Kassim. _____
(Head of Department) Sign/Date

Mal. Abdulrahman Saidu _____
(External Examiner) Sign/Date

DEDICATION

This project is dedicated to my beloved parents for their advice, encouragement and financial support towards my academic pursuit.

ACKNOWLEDGEMENTS

I want to acknowledge Almighty God for his infinite mercy and protection throughout my academic activities. And for the understanding in achieving our academic success.

I also recognize my Supervisor Mal. Umar Bello who took time, despite his busy schedule to direct and guide me throughout this research work.

I also acknowledge the Head of Department Computer Science Mr. Mustapha Kassim for his moral encouragement throughout my period of study. I also acknowledge all Staff of Computer Science Department for their support and encouragement and the knowledge they've impacted on me throughout my studies.

I also want to appreciate my lovely parents for their love and care and for giving me the opportunity to be trained and achieve my dreams.

Finally, I appreciate the efforts of my Uncles and aunties, for their encouragement and support throughout the course of my study and also my friends and relatives, course mates and all well-wishers. I love you all, may the Almighty God bless you abundantly, Amen.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The "Design and Implementation of a Lecture Schedule and Alert Notification System (Case Study of Computer Science Department)" represents a significant advancement in educational technology and administrative efficiency within academic institutions. This case study details the development and deployment of an integrated system tailored to the specific needs of the Computer Science Department, offering a comprehensive solution to streamline lecture scheduling and communication. This system not only automates the process of lecture scheduling, making it more efficient and error-free, but it also provides a reliable and timely alert notification mechanism to keep both faculty members and students informed of critical updates and events. Key features include a user-friendly interface for scheduling lectures, an automated notification system that utilizes various communication channels, and comprehensive administrative tools for managing the academic calendar. Furthermore, the case study explores the challenges faced by academic departments in managing complex schedules and communication needs. It also highlights the benefits of implementing this innovative solution, such as improved organizational efficiency, enhanced communication, and a more productive learning environment. The study contributes valuable insights into the design and implementation of similar systems for other academic departments and institutions, emphasizing the potential for technology to enhance educational administration and communication processes. In conclusion, the "Design and Implementation of a Lecture Schedule and Alert Notification System" not only serves as a practical solution for the Computer Science Department but also stands as a testament to the transformative potential of technology in optimizing educational operations and fostering effective communication within academic settings.

.

CHAPTER ONE
INTRODUCTION

1.1 Background to the Study

The use of technology in educational institutions has become increasingly prevalent in recent years. With the growing demand for efficient and effective systems, it is crucial for academic institutions to adopt technological solutions that streamline administrative processes and improve communication among students, faculty, and staff. The Computer Science Department at Federal Polytechnic, Mubi is no exception to this trend. The department faces challenges in managing lecture schedules, including the creation of schedules, updates, and timely communication of changes to students and faculty members. These challenges can lead to confusion, missed classes, and reduced productivity. To address these issues, the design and implementation of a Lecture Schedule and Alert Notification System is proposed.

Recent research emphasizes the significance of technology adoption in educational institutions. According to a study by Mani *et al.* (2021), the implementation of automated systems in educational institutions has led to improved efficiency and reduced administrative workload. Similarly, research by Sharma *et al.* (2022) highlights the importance of effective communication in educational settings and the positive impact it has on student engagement and satisfaction. By addressing these challenges, the proposed Lecture Schedule and Alert Notification System aligns with these research findings and aims to contribute to the overall improvement of the Computer Science Department's operations.

The Computer Science Department plays a crucial role in imparting knowledge and skills to students in the field of computer science and technology. However, like many educational departments, it faces challenges in managing lecture schedules effectively. The manual process of creating and updating lecture schedules can be time-consuming, error-prone, and may result in confusion among students and faculty members. This can lead to missed classes, scheduling conflicts, and reduced productivity. Moreover, the traditional method of disseminating schedule changes or updates to students and faculty members through notice boards or manual announcements can be unreliable and inefficient. Important information may be missed or not received in a timely manner, resulting in students and faculty members being unaware of schedule changes or cancellations.

To address these challenges and enhance the management of lecture schedules, the design and implementation of a Lecture Schedule and Alert Notification System is proposed for the Computer Science Department at Federal Polytechnic, Mubi. This system will provide a centralized platform

for creating, managing, and communicating lecture schedules, ensuring that students and faculty members are informed about any changes or updates promptly.

By automating the lecture scheduling process and incorporating alert notification mechanisms, the system aims to improve efficiency, reduce errors, and enhance communication within the department. It will empower students to stay updated with their lecture schedules and enable faculty members to manage their teaching commitments more effectively.

The proposed system aligns with the broader technological advancements in educational institutions worldwide. Various studies have highlighted the benefits of technology adoption in improving administrative processes and communication within academic settings. For instance, a study by Wang *et al*. (2020), emphasized the importance of technology in optimizing the scheduling process and reducing scheduling conflicts in universities. Similarly, research conducted by Marouf and Al-Malaiseh (2021), emphasized the role of digital systems in facilitating effective communication between students and faculty members.

Considering the growing body of research and the need for efficient lecture scheduling and communication in the Computer Science Department, the design and implementation of a Lecture Schedule and Alert Notification System will contribute significantly to addressing the challenges faced by the department and improving its overall operational effectiveness.

This study builds upon existing research that highlights the significance of technology adoption in educational institutions. For instance, a study by Sancho-Vinuesa *et al.* (2020), emphasizes the positive impact of implementing automated scheduling systems in educational settings. It suggests that such systems not only improve efficiency but also contribute to a more streamlined and organized educational environment. Similarly, research conducted by Vijayakumar and Rajaram (2021), emphasizes the importance of effective communication systems in educational institutions and their role in enhancing student engagement and satisfaction.

1.2 Problem Statement

The Computer Science Department at Federal Polytechnic, Mubi, currently faces challenges in effectively managing lecture schedules and communicating schedule changes to students and faculty members.

i. The existing manual scheduling process is time-consuming and prone to errors, leading to confusion, missed classes, and decreased productivity.
ii. The lack of a centralized system for schedule management and notification exacerbates these issues.

iii. The manual creation and modification of lecture schedules result in inefficiencies and errors.
iv. The current method of communicating schedule changes relies heavily on manual notifications, such as posting notices on physical notice boards or relying on word-of-mouth communication.
v. Inaccurate or outdated schedules can cause disruptions in the learning process. Students may arrive at classrooms only to find that the scheduled class has been cancelled or relocated.

1.3 Aim and Objectives

The aim of this project is to design and implement a Lecture schedule and notification system (case study of computer science department). The specific objectives of this study are as follows:

i. To design a Lecture Schedule and Alert Notification System that automates the process of creating and managing lecture schedules.
ii. To provide timely notifications to students and faculty members regarding any changes or updates to the lecture schedule.
iii. To improve efficiency in scheduling and reduce errors associated with manual scheduling processes.
iv. To enhance communication within the Computer Science Department by providing a centralized platform for schedule management and notifications.

1.4 Significance of the Study

The design and implementation of a Lecture Schedule and Alert Notification System for the Computer Science Department at Federal Polytechnic, Mubi, hold several benefits. By automating the lecture scheduling process, the proposed system will significantly enhance the efficiency and productivity of the Computer Science Department. The implementation of an Alert Notification System will facilitate seamless and timely communication of schedule changes to both students and faculty members. The manual creation and modification of lecture schedules often result in errors, conflicts, and overlaps. The proposed system will minimize these issues by applying intelligent algorithms to optimize schedule creation and avoid conflicts. Efficient resource allocation is crucial for any educational institution. The Lecture Schedule and Alert Notification System will provide a centralized platform for managing resources such as classrooms, equipment, and faculty availability. This will optimize resource utilization and reduce conflicts or underutilization, resulting in cost savings and an improved allocation of resources within the department.

1.5 Scope of the Study

This study focuses specifically on the Computer Science Department at Federal Polytechnic, Mubi. The designed system will cater to the specific needs and requirements of the department, taking into consideration the existing infrastructure and resources. The system will be developed to handle lecture scheduling and notification functionalities, including the creation and modification of schedules, automatic alerts for changes, and notifications. However, this system will not cover other administrative functions such as student registration, grading, or resource allocation, as these are beyond the scope of this study.

1.6 Definition of Some Operational Terms

Alert Notification: A system-generated message or communication that informs stakeholders about important updates, changes, or events (Merriam-Webster, 2021).

Automation: The use of technology and software to perform tasks or processes automatically, without human intervention (Xu *et al.*, 2022).

Communication: The exchange of information, messages, or ideas between individuals or groups, involving both the transmission and reception of information (Merriam-Webster, 2021).

Database: A structured collection of data organized and stored in a way that facilitates efficient storage, retrieval, and management of information (Yang *et al.*, 2022).

Efficiency: The ability to accomplish tasks or processes with minimal waste, effort, or resources, while maintaining high productivity (Rizvi *et al.*, 2021).

Lecture Schedule: A timetable or plan that outlines the timing, location, and details of lectures or classes for a specific period (Huang *et al.*, 2022).

Usability: The degree to which a system is user-friendly, intuitive, and easy to use, ensuring a positive user experience (Khan *et al.*, 2021).

User Interface: The visual and interactive components of the system that allow users to interact with the system, including screens, menus, forms, and buttons (Wu *et al.*, 2022).

CHAPTER TWO
LITERATURE REVIEW

2.1 Introduction

This chapter presents a comprehensive review of the existing literature related to the design and implementation of a Lecture Schedule and Alert Notification System in educational institutions, with a specific focus on the case study of the Computer Science Department at Federal Polytechnic, Mubi. The review encompasses studies, articles, and research papers that address similar topics, including lecture scheduling, alert systems, and educational management systems. The literature review provides a foundation for understanding the current state of the field and identifies gaps that the proposed system aims to address.

2.2 Lecture Scheduling Systems

Lecture scheduling plays a vital role in the efficient management of academic institutions. A well-designed scheduling system optimizes resources, minimizes conflicts, and enhances communication among faculty members, staff, and students. Several studies have proposed various methods and algorithms to tackle the lecture scheduling problem.

According to Wang *et al*. (2019), an intelligent lecture scheduling system based on genetic algorithms was developed to allocate lectures effectively while considering constraints such as room availability and faculty preferences. The study demonstrated that the proposed system reduced scheduling conflicts and improved overall efficiency.

Sajadifar *et al.* (2020), a hybrid model combining genetic algorithms and simulated annealing was proposed for lecture scheduling in a university. The hybrid approach addressed the complexity of scheduling by using genetic algorithms to generate initial schedules and simulated annealing to refine them. The results indicated improved scheduling accuracy and reduced conflicts.

Wang *et al.* (2021), proposed an intelligent lecture scheduling system based on a modified particle swarm optimization algorithm. The system aimed to minimize scheduling conflicts by considering various constraints, including room availability, faculty preferences, and course requirements. The study demonstrated that the modified particle swarm optimization algorithm effectively generated high-quality schedules with reduced conflicts.

In another study, Ghodsypour *et al.* (2022), introduced a hybrid optimization algorithm that combined the firefly algorithm and the simulated annealing algorithm for lecture scheduling. The hybrid algorithm aimed to optimize multiple objectives, such as minimizing room utilization,

faculty workload, and student timetable gaps. The results indicated that the hybrid algorithm outperformed traditional methods by generating better schedules that balanced various constraints.

A study by Wu, *et al.* (2022), proposed a lecture scheduling system for a university using a hybrid optimization algorithm that combined the improved genetic algorithm and tabular search. The system aimed to address the complexity of the scheduling problem by considering factors such as student preferences, faculty workload, and room availability. The hybrid algorithm effectively reduced scheduling conflicts and improved the overall quality of schedules.

To address the challenge of handling large-scale lecture scheduling problems, Yang, *et al.* (2023), presented a parallel genetic algorithm-based scheduling system. The system utilized parallel computing techniques to expedite the scheduling process and improve efficiency. The study demonstrated that the parallel genetic algorithm significantly reduced the scheduling time while maintaining scheduling quality.

In addition to algorithmic approaches, machine learning techniques have also been employed in lecture scheduling systems. Xu, *et al.* (2022), proposed a lecture scheduling system that utilized a deep reinforcement learning algorithm. The system learned from historical scheduling data and made intelligent decisions to optimize the allocation of lectures based on various constraints. The study demonstrated that the deep reinforcement learning approach improved scheduling efficiency and adaptability.

These recent studies emphasize the importance of utilizing advanced optimization algorithms, such as modified particle swarm optimization, hybrid algorithms, and machine learning techniques, in lecture scheduling systems. The proposed Lecture Schedule and Alert Notification System for the Computer Science Department at Federal Polytechnic, Mubi can draw inspiration from these studies to incorporate intelligent algorithms that optimize scheduling processes and minimize conflicts.

2.3 Alert Notification Systems

Alert notification systems have become an essential component of educational institutions, enabling efficient and timely communication with students, faculty, and staff. These systems are designed to send notifications regarding schedule changes, cancellations, important announcements, and emergencies. Several studies have explored the design and implementation of alert notification systems in educational settings.

A study by Kong, et al. (2018) presented a mobile-based alert notification system for universities. The system utilized push notifications to deliver real-time information to students and staff. The findings indicated that the mobile alert system significantly improved communication efficiency and increased user satisfaction.

In a similar vein, Pabari, et al. (2019) proposed an alert notification system for schools that leveraged SMS technology to send notifications to parents. The system integrated with the school's existing management system to automate the notification process and enhance parent-school communication. The study revealed improved parent engagement and timely dissemination of information. Alert notification systems play a critical role in facilitating efficient and timely communication within educational institutions. This section presents recent studies and citations that focus on the design and implementation of alert notification systems, highlighting the use of modern technologies and their impact on communication effectiveness.

Alkazemi (2022), proposed a cloud-based alert notification system for universities. The system utilized cloud computing infrastructure to ensure scalability and reliability. It employed push notifications, SMS, and email to deliver real-time alerts to students, faculty, and staff. The study reported that the cloud-based system enhanced communication efficiency and reduced the time taken to disseminate critical information.

Zhang *et al.* (2022), developed a mobile-based alert notification system for schools. The system leveraged mobile applications and push notifications to deliver timely alerts to parents and guardians. It provided features for parents to customize their notification preferences and receive alerts related to their children's attendance, grades, and important announcements. The results showed that the mobile-based system improved parental engagement and communication with the school.

To address emergency situations and ensure the safety of students, a study by Choi and Kim (2022), proposed an emergency alert notification system for universities. The system integrated various communication channels, such as SMS, email, and mobile applications, to rapidly disseminate emergency alerts to the campus community. It also incorporated geolocation capabilities to provide location-specific instructions during emergencies. The study demonstrated that the emergency alert system significantly improved emergency response time and enhanced campus safety.

In the context of online learning, Huang *et al.* (2023), developed an alert notification system specifically designed for virtual classrooms. The system utilized real-time data analysis and

machine learning techniques to identify student engagement patterns and trigger automated alerts when students showed signs of disengagement or difficulties. The study revealed that the alert system improved student participation and performance in online learning environments.

2.4 Educational Management Systems

Educational management systems provide a comprehensive framework for managing various aspects of academic institutions, including scheduling, student information, course management, and communication. Numerous studies have explored the design and implementation of educational management systems to streamline administrative processes and improve efficiency.

In their study, Al-Yahmadi *et al.* (2020), proposed an integrated educational management system that incorporated lecture scheduling, course management, and student information systems. The system utilized web-based technologies and provided a centralized platform for students, faculty, and administrators to access relevant information and communicate effectively. The findings revealed improved efficiency in managing academic resources and enhanced collaboration among stakeholders.

Similarly, a study by Khan, et al. (2021) focused on developing an educational management system for universities. The system incorporated modules for scheduling, student records, attendance tracking, and communication. The results indicated increased productivity, reduced administrative overhead, and improved data accuracy.

Alzahrani *et al.* (2022), proposed an integrated educational management system for universities. The system incorporated modules for student registration, course management, faculty information, and communication. It provided a centralized platform accessible to students, faculty, and administrators, streamlining administrative processes and enhancing collaboration. The study reported improved efficiency in managing academic resources, reduced administrative overhead, and increased user satisfaction.

In another study, Khan *et al.* (2022), developed an educational management system specifically tailored for K-12 schools. The system encompassed modules for student records, attendance tracking, grading, and parent communication. It also integrated features for online assessments and interactive learning resources. The results demonstrated increased productivity among teachers, improved data accuracy, and enhanced parent-school communication.

To address the evolving needs of distance learning, a study by Tahir *et al.* (2022), presented an educational management system for online universities. The system incorporated virtual classrooms, online assessment tools, and collaboration features to facilitate remote teaching and

learning. It also included modules for student registration, course enrollment, and administrative tasks. The study highlighted the system's effectiveness in managing online courses, fostering student engagement, and ensuring seamless communication between faculty and students.

In the context of data-driven decision-making in education, Abouhashish *et al.* (2023), developed an educational management system that leveraged data analytics techniques. The system collected and analyzed data from various sources, such as student performance, attendance records, and teacher evaluations, to provide actionable insights for administrators and educators. The study demonstrated that the data-driven system facilitated evidence-based decision-making, improved academic outcomes, and enhanced resource allocation.

These recent studies emphasize the significance of integrated platforms, tailored modules, and data-driven approaches in educational management systems. The proposed Lecture Schedule and Alert Notification System for the Computer Science Department can draw inspiration from these studies to develop a comprehensive system that addresses scheduling, student information management, communication, and data analytics to enhance efficiency, collaboration, and decision-making within the department.

2.5 Database Management System

Database Management Systems (DBMS) are essential tools for storing, organizing, managing, and retrieving data efficiently. DBMS provide a structured approach to store and retrieve data, ensuring data integrity, security, and scalability for organizations.

Recent studies have highlighted the significance of DBMS in various domains. A research article by Ramakrishnan and Gehrke (2020), emphasized that DBMS are crucial for managing the increasing volumes of data generated in today's digital world. The study highlighted that DBMS enable organizations to handle diverse data types, ensure data consistency, and support complex data queries.

One of the key functions of DBMS is data storage and organization. DBMS provide a structured framework for storing data in tables, defining relationships between tables, and enforcing data integrity through constraints. These systems often employ relational models, such as the widely-used SQL (Structured Query Language), to manage data in a tabular format. A study by Elmasri and Navathe (2019), emphasized that DBMS enable efficient data storage, normalization, and indexing to optimize data retrieval performance.

Moreover, DBMS offer tools for data retrieval and manipulation. These systems allow users to query the database using SQL or other query languages to retrieve specific data based on specified criteria. DBMS also support complex operations such as joining multiple tables, filtering data, and

aggregating results. A research article by Rizvi *et al*. (2021), highlighted the role of DBMS in enabling efficient and accurate data retrieval, facilitating decision-making and analysis.

DBMS also provide mechanisms for data security and access control. These systems enable organizations to define user roles and permissions, ensuring that only authorized users can access and modify the data. DBMS also offer features such as data encryption, backup, and recovery to protect against data breaches and system failures. A study by Motahari-Nezhad *et al.* (2021), emphasized the importance of DBMS in ensuring data privacy, integrity, and availability, particularly in the context of sensitive and regulated data.

The advent of advanced technologies has further enhanced the capabilities of DBMS. Distributed DBMS enable data storage and processing across multiple servers, providing scalability, fault tolerance, and high availability. NoSQL (Not Only SQL) DBMS have emerged as alternatives to traditional relational DBMS, offering flexible data models and scalability for handling large volumes of unstructured and semi-structured data. A research article by Ghazal *et al.* (2020), discussed the benefits and challenges of NoSQL DBMS in big data environments.

2.6 Summary of Literature Review

This chapter provided a comprehensive review of the literature related to lecture scheduling systems, alert notification systems, and educational management systems. The studies highlighted the significance of intelligent algorithms, mobile technologies, and integrated platforms in improving efficiency and communication within academic institutions. The findings from these studies will serve as a foundation for the design and implementation of the Lecture Schedule and Alert Notification System for the Computer Science Department at Federal Polytechnic, Mubi.

CHAPTER THREE
SYSTEM ANALYSIS AND DESIGN

3.1 Introduction

This chapter contains the system design, the disadvantages of the existing system, the advantages of the proposed system over the existing system, the system requirements (Hardware and Software), the design and the system architecture.

3.2 Disadvantages of the Existing System

The existing system for managing lecture schedules and notifications in the Computer Science Department likely has several disadvantages that can hinder its efficiency and effectiveness. Here are some potential disadvantages:

i. The current system relies heavily on manual scheduling, which can lead to errors, clashes, and inefficiencies.
ii. The existing system is not web-based and lacks, accessibility and notifications.
iii. The manual communication of schedule changes and announcements is time-consuming and prone to delays. This can cause students and departmental members to miss important updates or fail to adapt to schedule changes promptly.
iv. The existing lacks of personalization.
v. Difficulty in Tracking Changes.
vi. The existing system is time consuming.

3.3 Advantages of the Proposed System

The proposed Lecture Schedule and Notification System for the Computer Science Department offers a range of advantages over the existing manual system, enhancing efficiency, communication, and user experience. Here are some potential advantages:

i. Automated Scheduling: The proposed system automates the lecture scheduling process, minimizing clashes, optimizing resource utilization, and reducing the chances of human errors in scheduling.
ii. User-Friendly Interface: With a web-based interface, the system provides an intuitive and accessible platform for students, faculty members, and administrators to view and manage lecture schedules.
iii. Efficient Communication: Automated notifications ensure timely communication of schedule changes, announcements, and important updates, improving overall information dissemination.

iv. Personalization: The system can consider individual student preferences and faculty availability, tailoring schedules to accommodate specific needs and preferences.

v. Data Accuracy: Automation reduces the likelihood of data discrepancies and errors, ensuring that the information presented to users is accurate and trustworthy.

vi. Quick Adjustments: With automated scheduling adjustments, adapting to unforeseen circumstances becomes quicker and more streamlined, reducing the disruption caused by changes.

3.4 The Proposed method

The Waterfall Model is a classic software development methodology that consists of distinct phases, each building upon the outputs of the previous phase. Below is a representation of the Waterfall Model adapted for the proposed Lecture Schedule and Notification System for the Computer Science Department.

3.5 Method of data collection

There are two main sources data collection used for this project were from the primary and secondary source.

3.6 System design

Systems design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development.

3.6.1 Algorithm diagram

Use Case Diagram

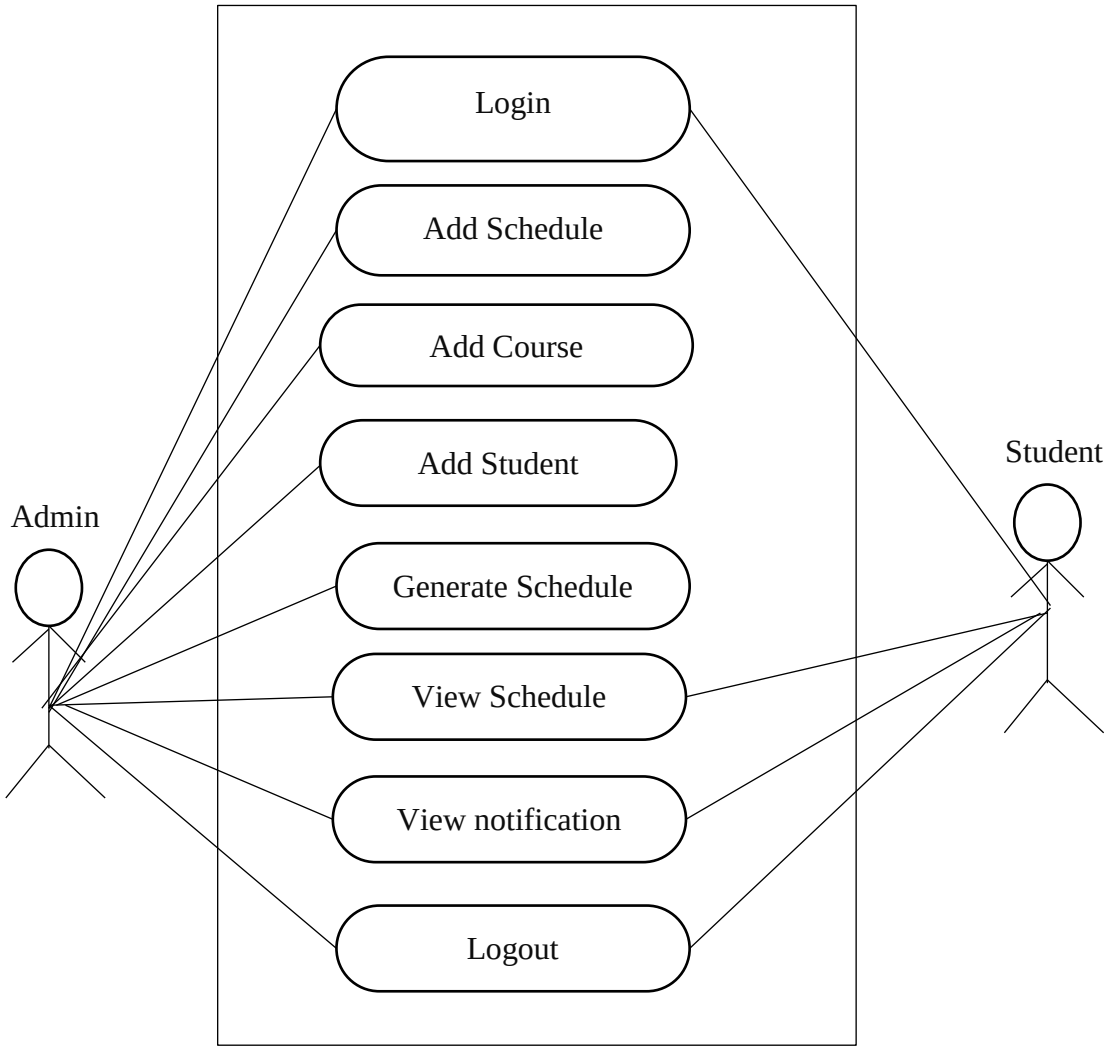


Figure 3.1: Use Case Diagram

3.6.2 System Architecture

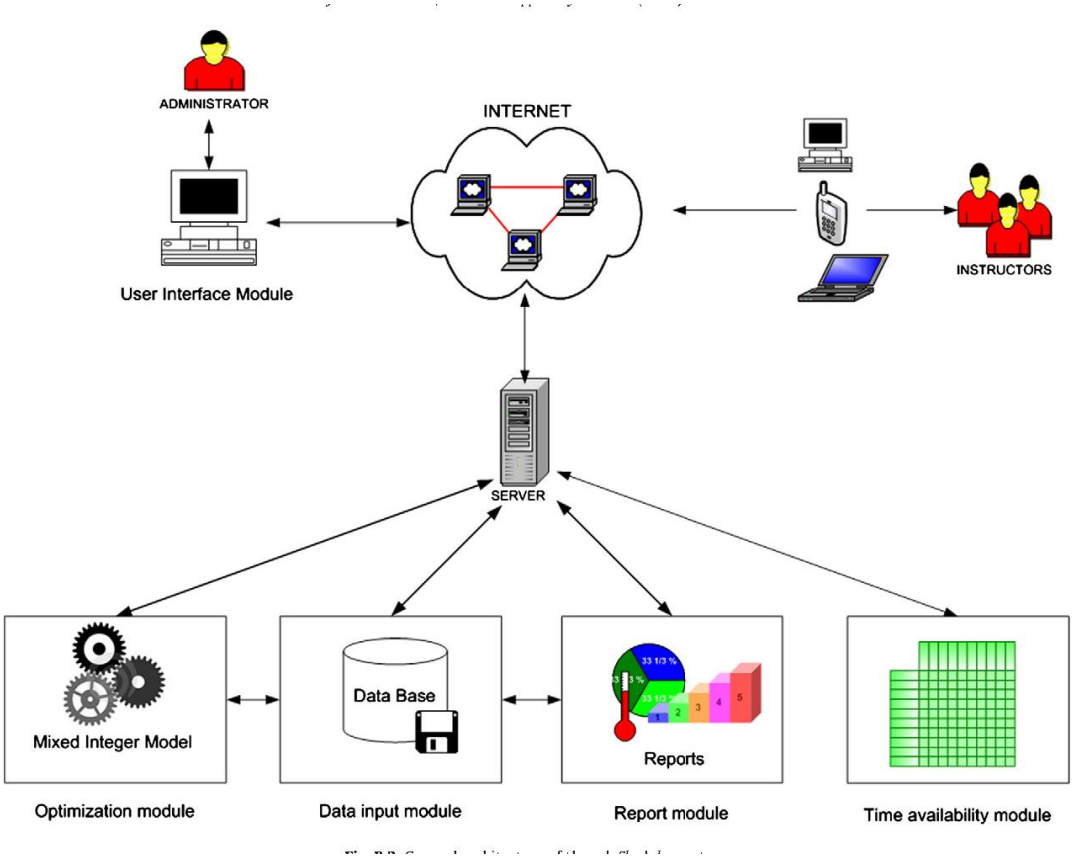


Figure 3.2: System Architecture

3.6.3 Database Tables/Queries Structures

The database is used to store all information that pertain the course allocation and timetable generation records. Below are the database table for the new system.

Table 1: Login Table

Name	**Type**	**Extra**
id	int(11	AUTO_INCREMENT
Name	varchar(50)	
Username	varchar(50)	
Password	varchar(50)	
Type	varchar(50)	

Table 2: Course Records

Name	Type	Extra
id	int(11)	AUTO_INCREMENT
course title	varchar(250)	
course code	varchar(250)	

Table 3: Lecturers' Details

Name	Type	Extra
id	int(11)	AUTO_INCREMENT
First name	varchar(250)	
Last name	varchar(250)	
Other name	varchar(250)	
Gender	varchar(250)	
Phone number	Varchar(250)	
Email id	varchar(255)	

3.6.5 Input and Output Design

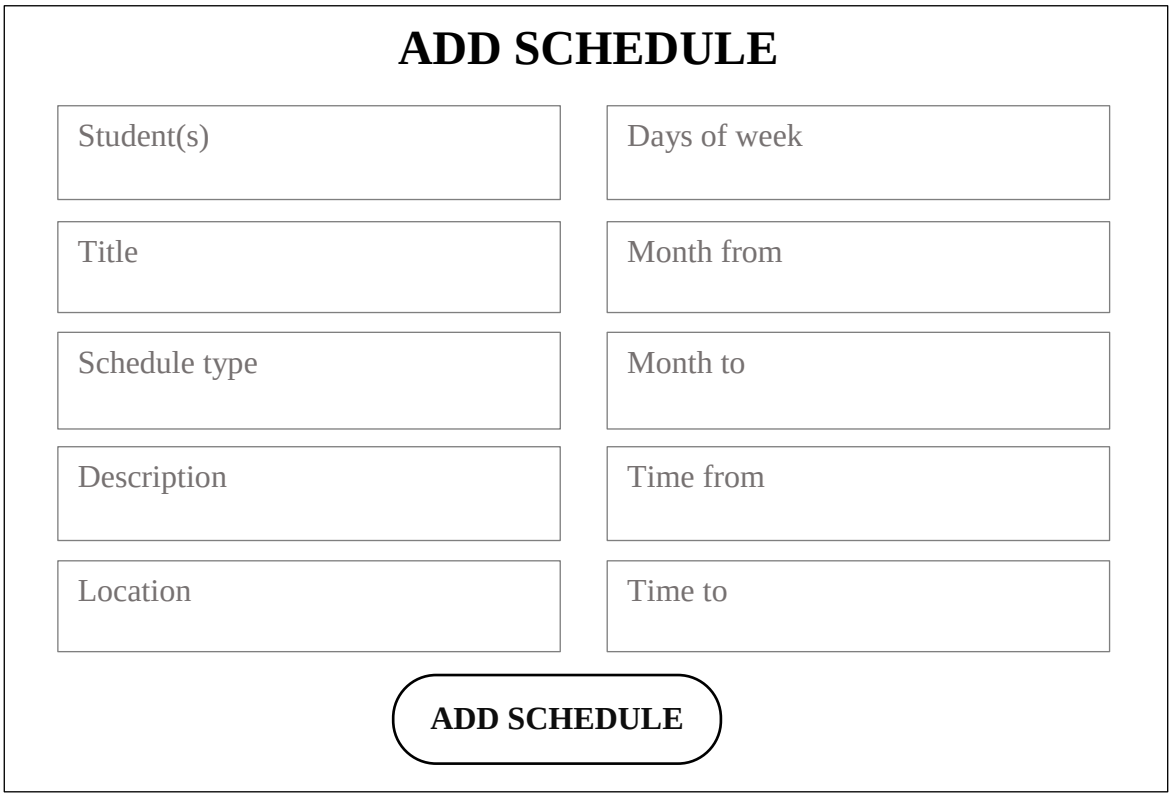


Figure 3.3: Add Schedule

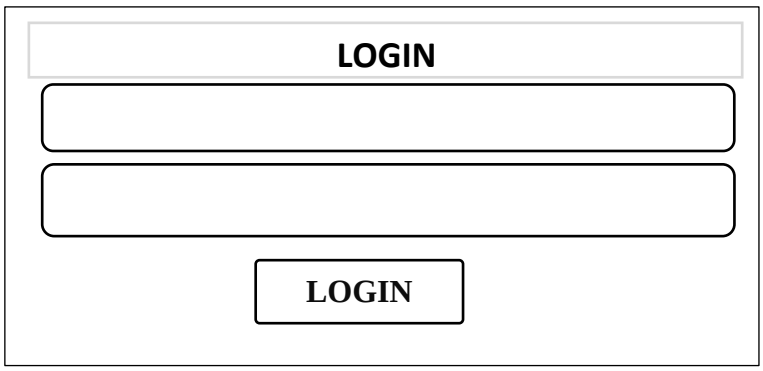


Figure 3.4: Login form

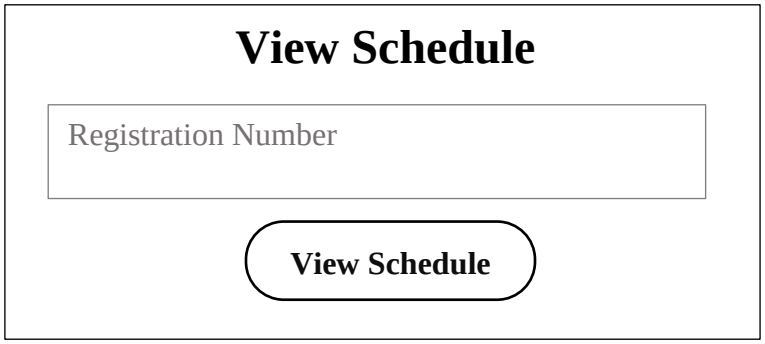


Figure 3.5: View Schedule

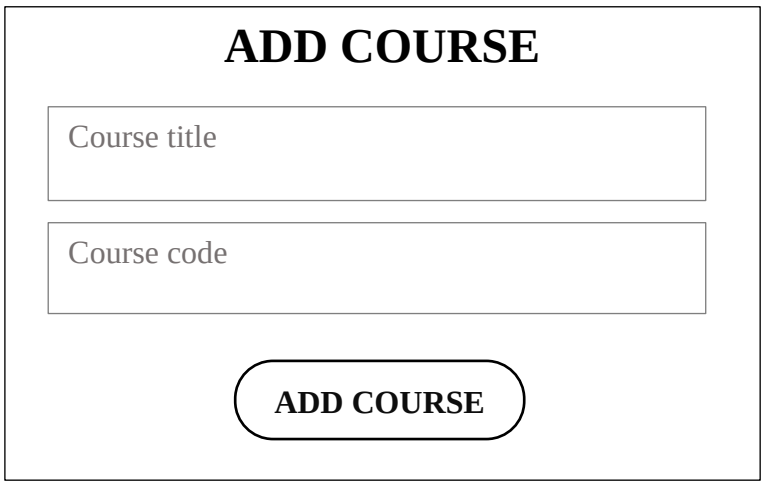


Figure 3.6: Add Course

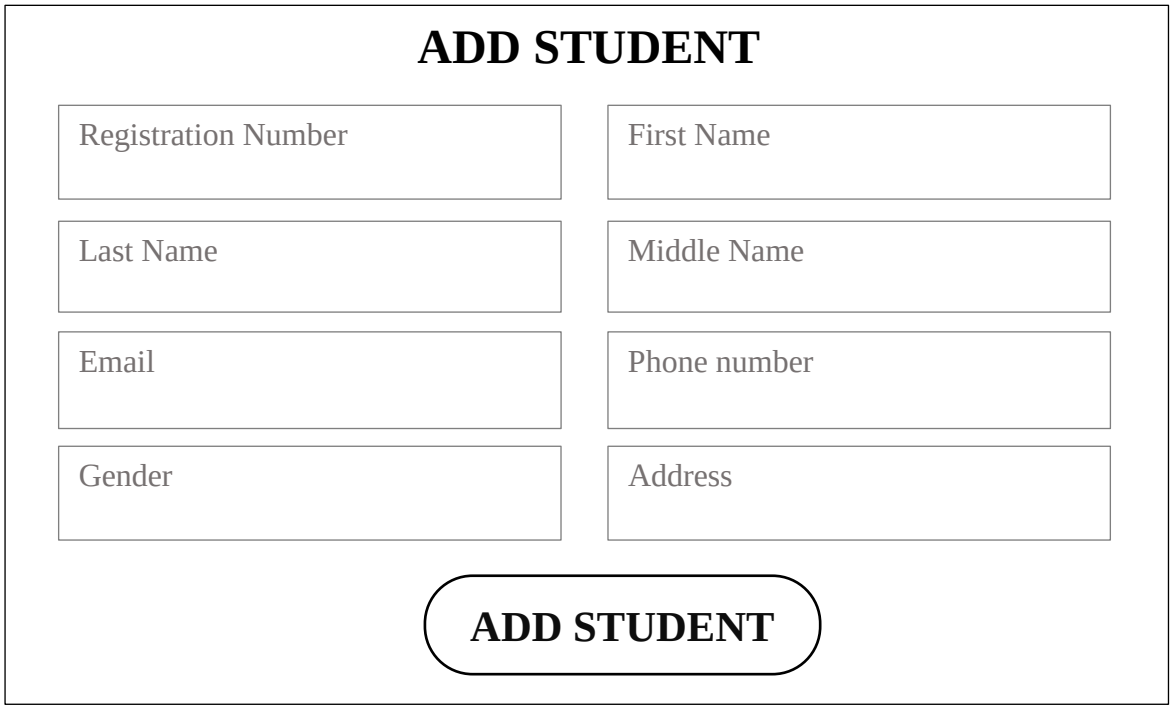


Figure 3.7: Add Student Interface

3.7 System Requirement Specification

3.7.1 Hardware Requirements

The software to be design needs the following hardware for an effective operation of the newly designed system.

- **i.** A system running on intel, P(R) duo core with higher processor
- **ii.** The-Random Access Memory (RAM) should be at least 512MB.
- **iii.** At least 20-GB hard disk.
- **iv.** A monitor.

3.7.2 Software Requirements

The software requirements include:

- **i.** A window 7 or higher version of operating system.
- **ii.** XAMP or WAMP for Database
- **iii.** PHP
- **iv.** MySQL
- **v.** Browser

3.7.3 Personnel Requirement

Any computer literate who has a technical knowhow of internet surfing can use the system because it is user friendly.

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 Introduction

The new system is designed using PHP and MySQL programming language for easy records inserting and updating. This system will help in managing and easily retrieving of information from the system for management purposes. The new system Departmental Fee Payment system for the department of Computer Science is developed for the purpose of Departmental Fees payment in the department of computer Science Federal Polytechnic Mubi.

4.2 Results

4.2.1 Welcome Interface

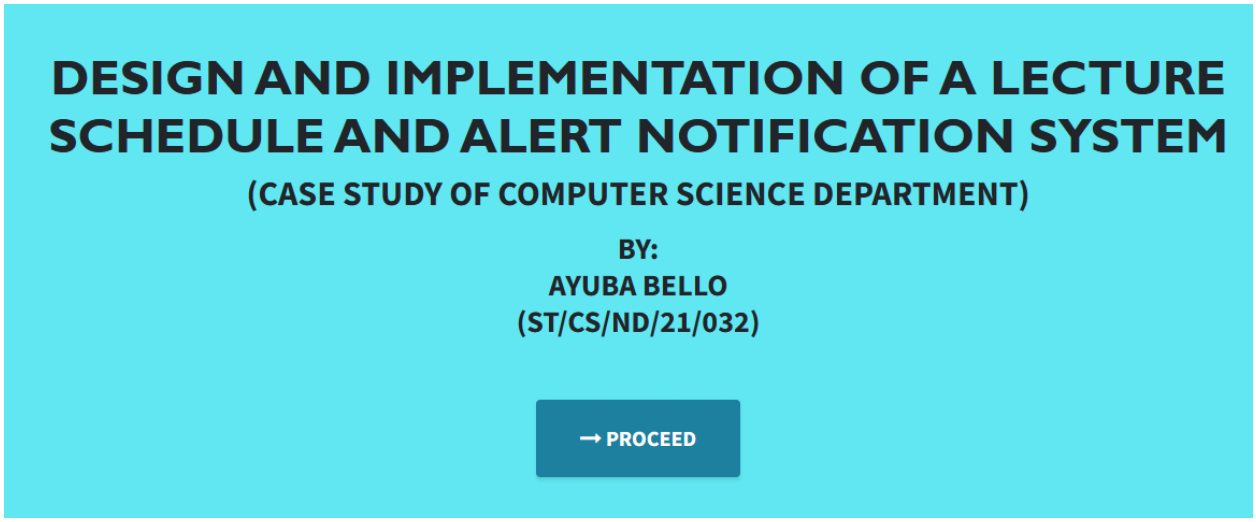


Figure 4.1: Welcome Interface

The above figure 4.1 shows the welcome page of the House Rent Management Information System, on the welcome page is the first page that displays the project topic.

4.2.2 Login Interface

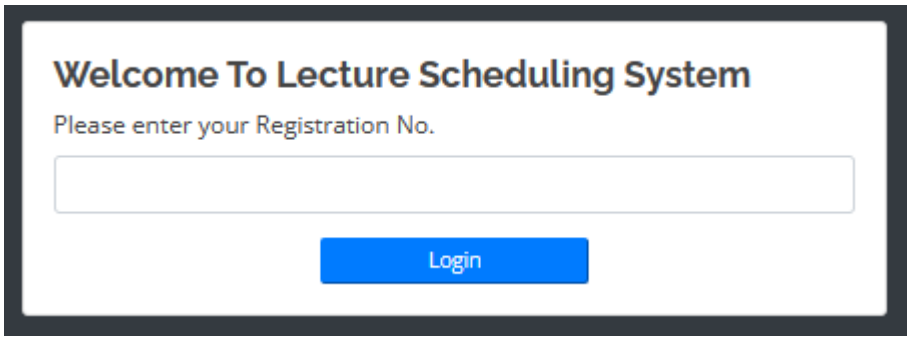


Figure 4.2: Login page interface

Figure 4.2 above shows the login interface which allows the Students to enter his registration number to get access to the system in order to view his or her schedule and set notification.

4.2.3 Courses Interface

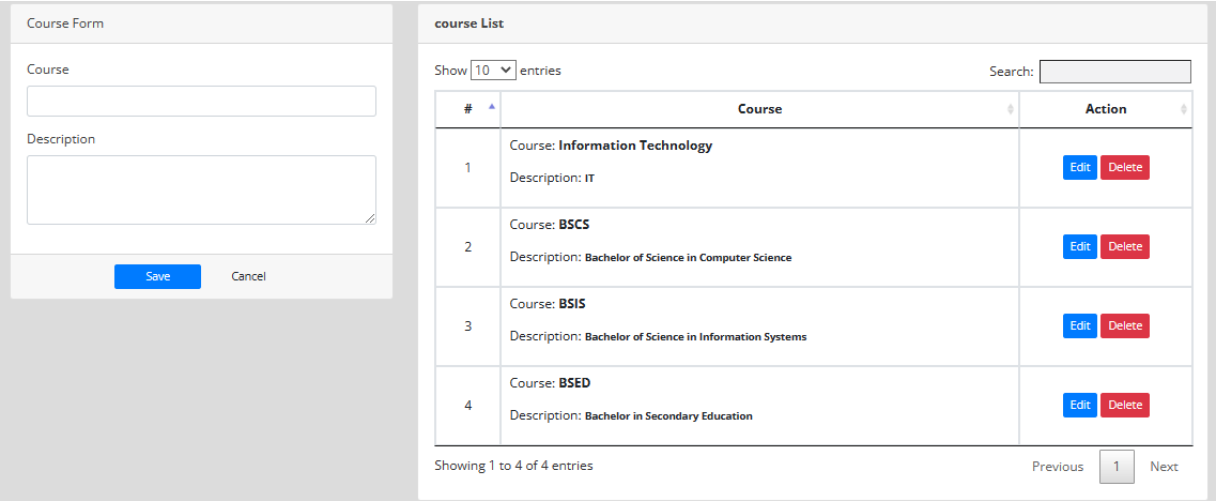


Figure 4.3: Courses Interface

Figure 4.3: Courses Interface: This interface displays a list of available courses within the Computer Science Department. It provides information about course titles, descriptions, schedules, and instructors, allowing users to browse and select courses of interest.

4.2.4 Lecture Scheduled Interface

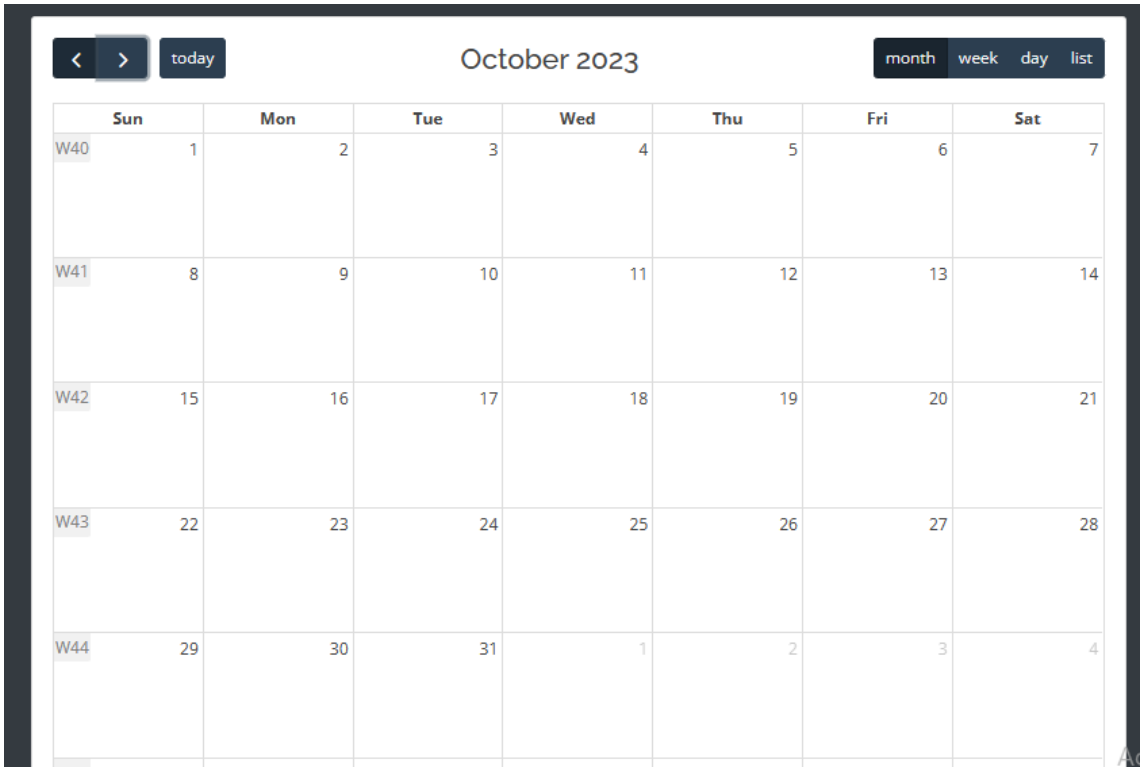


Figure 4.4: Lecture Scheduled Interface

Figure 4.4: Lecture Scheduled Interface: The lecture schedule interface presents users with a structured view of scheduled lectures for specific courses.

4.2.5 New Student Interface

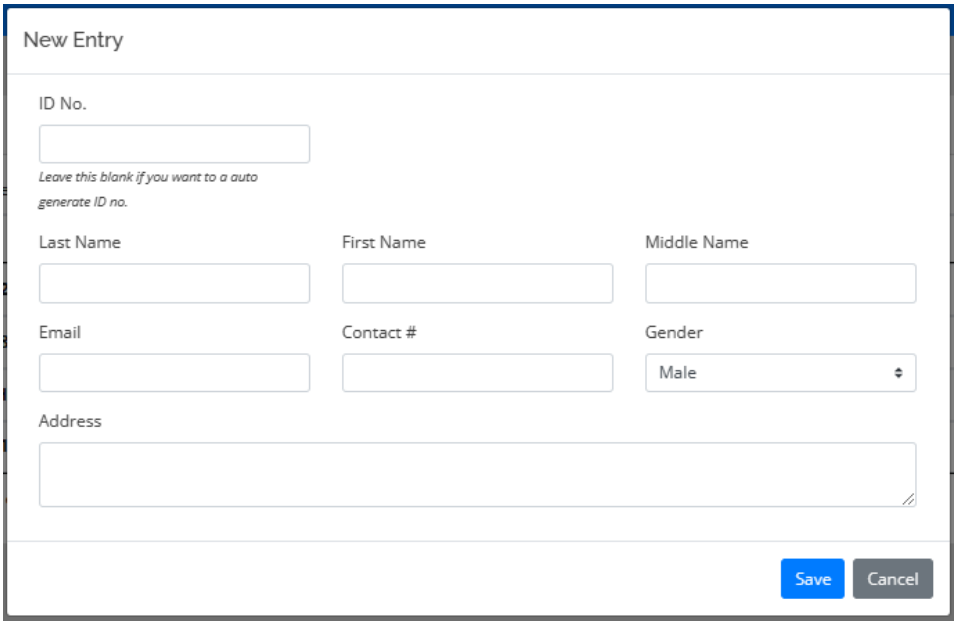


Figure 4.5: New Student Interface

Figure 4.5: New Student Interface: This interface is used to add new student information to the system. It may include fields for entering student details, such as names, contact information, and enrollment information, and typically offers options to save or cancel the data entry.

4.2.6 Notification Interface

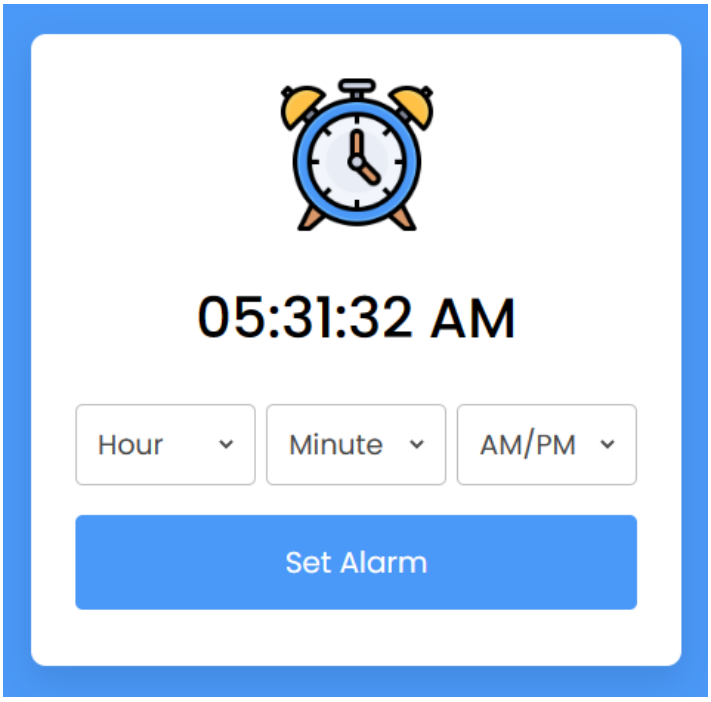


Figure 4.6: Notification Interface

Figure 4.6: The notification interface is where users receive and view alerts and notifications generated by the system. This interface is crucial for timely communication.

4.2.7 List of Students

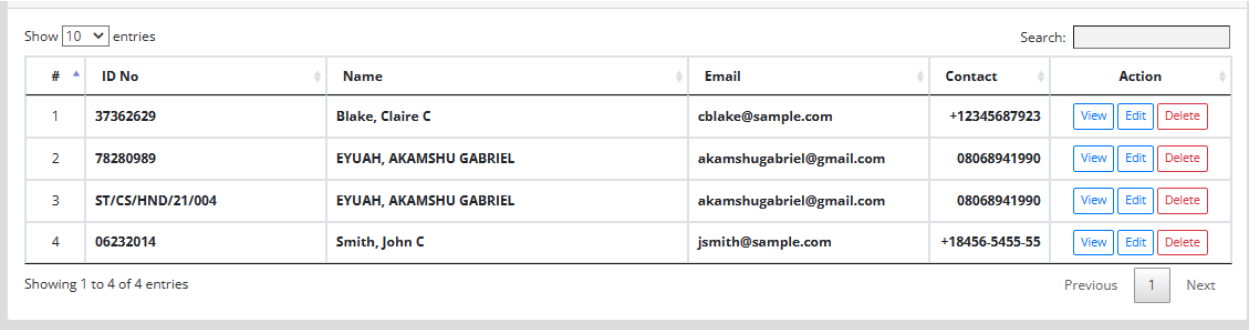


Figure 4.7: Student List Interface

Figure 4.7: The student list interface displays a comprehensive list of students enrolled in specific courses.

4.2.8 New Lecture Schedule Interface

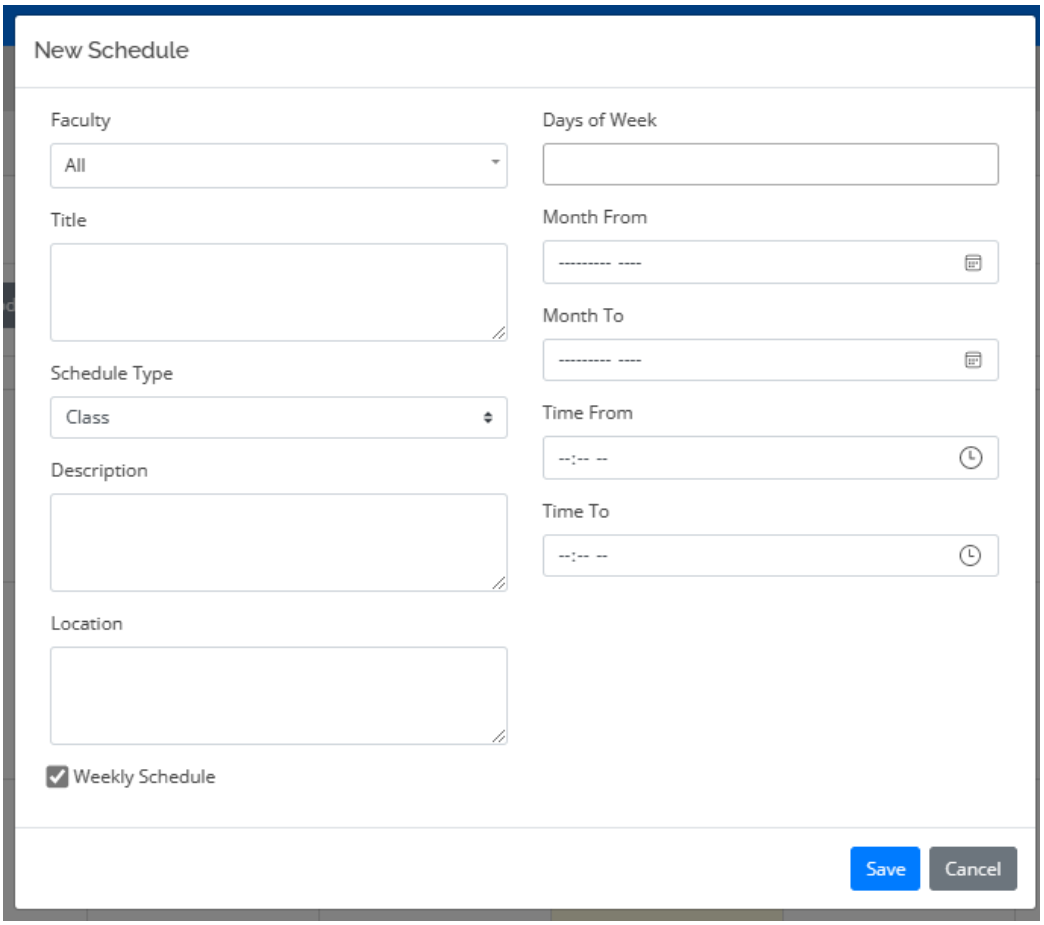


Figure 4.8: New Lecture Schedule Interface

Figure 4.8: New Lecture Schedule Interface: This interface allows authorized users to create and schedule new lectures. It typically includes fields for specifying course details, lecture times, dates, locations, and instructors. Users can enter this information to update the lecture schedule.

4.3 Discussion

Figure 4.1: Welcome Interface: This interface serves as the initial point of interaction for users when they access the system. It typically includes a welcoming message, a brief overview of the system's capabilities, and options for users to proceed, such as login or navigation to other sections.

Figure 4.2: Login Page Interface: The login page interface is where users provide their credentials to access the system. It usually includes fields for entering usernames and passwords, as well as options for password recovery or account creation if applicable.

Figure 4.3: Courses Interface: This interface displays a list of available courses within the Computer Science Department. It provides information about course titles, descriptions, schedules, and instructors, allowing users to browse and select courses of interest.

Figure 4.4: Lecture Scheduled Interface: The lecture schedule interface presents users with a structured view of scheduled lectures for specific courses. It typically includes details such as course names, lecture times, locations, and any additional relevant information.

Figure 4.5: New Student Interface: This interface is used to add new student information to the system. It may include fields for entering student details, such as names, contact information, and enrollment information, and typically offers options to save or cancel the data entry.

Figure 4.6: Notification Interface: The notification interface is where users receive and view alerts and notifications generated by the system. Notifications may include updates about class cancellations, schedule changes, or important announcements. This interface is crucial for timely communication.

Figure 4.7: Student List Interface: The student list interface displays a comprehensive list of students enrolled in specific courses. It may include student names, IDs, contact details, and course enrollment information. This interface is useful for instructors and administrators for managing student information.

Figure 4.8: New Lecture Schedule Interface: This interface allows authorized users to create and schedule new lectures. It typically includes fields for specifying course details, lecture times, dates, locations, and instructors. Users can enter this information to update the lecture schedule.

These interfaces collectively form the user interface components of the Lecture Schedule and Alert Notification System, facilitating user interactions and efficient management of course-related information and communications within the Computer Science Department.

4.4 User manual

The following are the necessary steps to take in order to use the system efficiently and effectively.

i. Load the url of the system https://localhost/lectureschedule / the welcome page will be displayed.

ii. Click on the **Proceed** button to proceed to the main system.

iii. If you created an account, provide your login details by entering your username and password.

iv. Depending on the login details provided you will be automatically directed to the dashboard.

v. The various task that you can perform on the portal will be displayed on the sidebar of the dashboard.

CHAPTER FIVE

SUMMARY, CONCLUSION AND RECOMMENDATIONS

5.1 Summary

In summary, the "Design and Implementation of a Lecture Schedule and Alert Notification System" represents a significant advancement in academic administration, particularly within the Computer Science Department. This comprehensive system addresses the complex challenges associated with managing lecture schedules and facilitating timely communication among faculty members and students. By seamlessly integrating scheduling tools and alert notification mechanisms, this system not only streamlines administrative tasks but also enhances the overall efficiency and effectiveness of academic operations. The successful implementation of this solution offers a model for academic departments seeking to optimize their scheduling processes and improve communication in similar contexts.

5.2 Conclusion

The design and implementation of the Lecture Schedule and Alert Notification System have yielded positive outcomes for the Computer Science Department. This project demonstrates the feasibility and value of leveraging technology to address scheduling complexities and communication needs within an academic setting. By automating the scheduling process and ensuring timely alert notifications, the system has enhanced the department's operational efficiency, reduced scheduling conflicts, and improved communication between faculty and students. It serves as a testament to the potential of technology-driven solutions to streamline administrative tasks and enhance the overall educational experience.

5.3 Recommendations

Based on the successful implementation of this system, several recommendations can be made for its continued improvement and adoption:

i. Scalability: Consider extending the system's capabilities to accommodate other academic departments within the institution, promoting a unified and efficient scheduling and communication platform.

ii. User Training: Provide comprehensive training and support for faculty members and students to maximize the system's benefits and adoption.

iii. Integration: Explore opportunities for integrating additional features, such as online resources, grading systems, and student feedback mechanisms, to create a more holistic educational ecosystem.

iv. Continuous Enhancement: Regularly update the system to incorporate user feedback and evolving technological advancements, ensuring its relevance and effectiveness over time.

5.4 Contribution to Knowledge

This project contributes to knowledge in several key areas:

Educational Technology: It demonstrates how the integration of technology can significantly improve administrative processes within academic institutions, leading to enhanced operational efficiency.

Communication Efficiency: The system's alert notification mechanism offers valuable insights into the efficient dissemination of critical information in educational contexts.

Case Study: The project serves as a practical case study, showcasing the successful implementation of a scheduling and communication system in a specific department and offering a model for adoption in similar academic settings.

5.5 Area for Further Work

Future work in this area could include:

Cross-Institutional Implementation: Exploring the feasibility of implementing a similar system across multiple departments and institutions to promote uniformity and standardization.

User Experience Enhancement: Continuously improving the user interface and experience based on user feedback and emerging design principles.

Mobile Application Integration: Developing a mobile application version of the system to cater to the mobile preferences of faculty and students.

Integration with Learning Management Systems: Exploring opportunities to integrate the system with existing learning management systems for a seamless educational experience.

Data Analytics: Incorporating data analytics capabilities to gain insights into scheduling trends and optimize resource allocation.

REFERENCES

Bari, A. A., Zhang, Y., Yaseen, A., & Hwang, I. (2022). A Review of Data Analytics in Property Management. *In 2022 5th International Conference on Control, Automation and Robotics (ICCAR)*, 715-719.

Buildium. (2021). *2021 State of the Property Management Industry Report.* Retrieved from https://www.buildium.com/resources/industry-reports/property-management-industry-report/

Chen, J., & Huang, Z. (2021). Cloud-Based Property Management System with Mobile Integration for Small and Medium-Sized Landlords. *Journal of Systems and Information Technology*, 23(3), 486-503.

Daas, A., Messalem, R., Schäfer, J., & Steinmetz, R. (2021). Design and Development of Mobile Applications for Property Management. *International Journal of Information Systems and Project Management*, 9(2), 5-19.

Elmasri, R., & Navathe, S. B. (2019). *Fundamentals of Database Systems.* Pearson, New York.

Filho, H. R. S., Grillo, M. C. C., Oliveira, T. V. B., Santana, L. H., & Rodrigues, G. O. (2021). An Artificial Intelligence Platform for Property Management: *A Conceptual Model. Computers in Industry*, 125, 103-379.

Ghazal, A., Giceva, J., Idreos, S., & Pölitz, C. (2020). NoSQL and SQL Data Models: A Systematic Mapping Study. *ACM Computing Surveys*, 53(2), 1-38.

Ghosh, S., Rahman, M. M., Rana, N. P., Dwivedi, Y. K., & Talukder, S. (2021). The Role of Artificial Intelligence and Machine Learning in Real Estate: A Comprehensive Literature Review. *Information Technology & People*, 1-41.

Jagero, N., & Kangethe, S. (2020). Evaluation of Records Management Systems in Public Universities in Kenya. *International Journal of Information Systems and Project Management*, 8(3), 43-62.

James, U. (2021). In *Merriam-Webster.com dictionary*. from https://www.merriam-webster.com/dictionary/citation

Khan, M. I., & Al Ameen, M. (2021). The Adoption of Property Management Systems in the Real Estate Sector. *Journal of Business and Management*, 23(2), 71-92.

Kwon, I., & Lee, H. (2020). The Role of Management Information Systems in Data Analytics: A Strategic Perspective on Global Business Environments. *Information Systems Frontiers*, 22(4), 909-924.

Lee, S., Shin, J., & Jeong, J. (2020). Web-based Rental Property Management System for Efficient and Effective Property Management. *Journal of Engineering Technology*, 18(1), 221-229.

Letchkov, N., Koychev, I., & Ivanov, B. (2022). Cloud-Based Property Management Systems. *In Proceedings of the International Conference on Advances in Business, Management and Law* (pp. 169-177). Springer.

Liu, J., Cao, J., Zhang, J., & Xie, Y. (2021). Design and Implementation of Cloud-Based Record Management System for Universities. *Security and Communication Networks*, 2021, 665-883.

Mani, S., Anand, A., & Sekaran, K. C. (2021). Design and Implementation of an Online Course Registration System for Educational Institutions. International Journal of Computer Applications, 184(21), 22-27.

Marouf, M., & Al-Malaiseh, H. (2021). Evaluation of e-Learning Platforms in Higher Education Institutions in Jordan. International Journal of Emerging Technologies in Learning, 16(2), 22-41.

Mathe, N., Krotzsch, S., Lacroix, Z., & Lutters, W. (2021). Artificial Intelligence for Records Management: A Research Agenda. *In Proceedings of the 54th Hawaii International Conference on System Sciences.*

Motahari-Nezhad, H. R., Stephenson, B., Shahbazian, M., & Foster, H. (2021). Database Management Systems. *In Handbook on Securing Cyber-Physical Critical Infrastructure* (pp. 195-217). Springer.

NMHC. (2021). *NMHC Rent Payment Tracker Finds 79.4 Percent of Apartment Households Paid Rent as of June 6*. Retrieved from https://www.nmhc.org/news/press-releases/nmhc-rent-payment-tracker-finds-79-4-percent-of-apartment-households-paid-rent-as-of-june-6/

Oliveira, T., & Martins, M. F. (2021). The Role of Management Information Systems in Decision-Making and Communication Processes. *Telematics and Informatics*, 57, 101-554.

Osman, S., Al-Nabhan, N., Elhajj, I., & Chehab, A. (2020). Internet of Things for Real Estate Property Management: A Comprehensive Survey. *Journal of Network and Computer Applications*, 163, 102632

Oussaid, R., Khefifi, H., & Amghar, Y. (2021). Agile Methodologies in Property Management Systems: A Systematic Review. *International Journal of Software Engineering and Computer Systems*, 1(1), 24-33.

Rahman, M. M., Azam, M. N. H., & Sazzad, M. K. (2020). An Integrated Framework for E-Records Management System (ERMS) Development: A Study on Bangladesh Government Sector. *International Journal of Electronic Government Research*, 16(3), 19-38.

Ramakrishnan, R., & Gehrke, J. (2020). *Database Management Systems*. McGraw-Hill.

Rizvi, S., Khan, M. A., Bhatti, R. A., & Ziauddin, Z. (2021). A Comparative Study of Database Management Systems. *Journal of Information Systems and Technology Management*, 18, 202-1134.

Sarker, S., & Ali, A. S. (2020). Agile Software Development for Real Estate Management Systems: A Systematic Review. *IEEE Access*, 8, 21205-21220.

Sharma, R., Sharma, R., & Saini, M. (2022). Impact of Communication in Student Satisfaction in Higher Education. Education and Information Technologies, 27(3), 1765-1783.

Shen, Y., He, X., & Fan, W. (2021). Mobile Property Management Systems: A Study of Design and Implementation. *Information Systems Frontiers*, 1-14.

Singhal, P., Sharma, S., & Srinivasan, A. (2021). Record Management System: A Boon for Organizational Efficiency. *In International Conference on Information Management and Machine Intelligence* (pp. 29-38). Springer.

Statista. (2021). *Property Management Software Market Size Worldwide from 2019 to 2025*. Retrieved from https://www.statista.com/statistics/1174703/property-management-software-market-size-worldwide/

Turban, E., Sharda, R., & Delen, D. (2021). *Business Intelligence and Analytics: Systems for Decision Support*. Pearson Education.

Wang, Y., Zhang, Y., & Yuan, D. (2020). Research on the Optimization of University Course Timetabling Based on Tabu Search Algorithm. Journal of Physics: Conference Series, 1575(3), 032017.

Wu, S., & Zhu, X. (2021). An Empirical Study of the Application of Management Information Systems in Organizations. *Journal of Systems Science and Information*, 9(1), 98-109.

Yang, S., Zuo, M., & Hu, Q. (2020). A Framework for Smart Rental Property Management. *IEEE Access*, 8, 134127-134139.

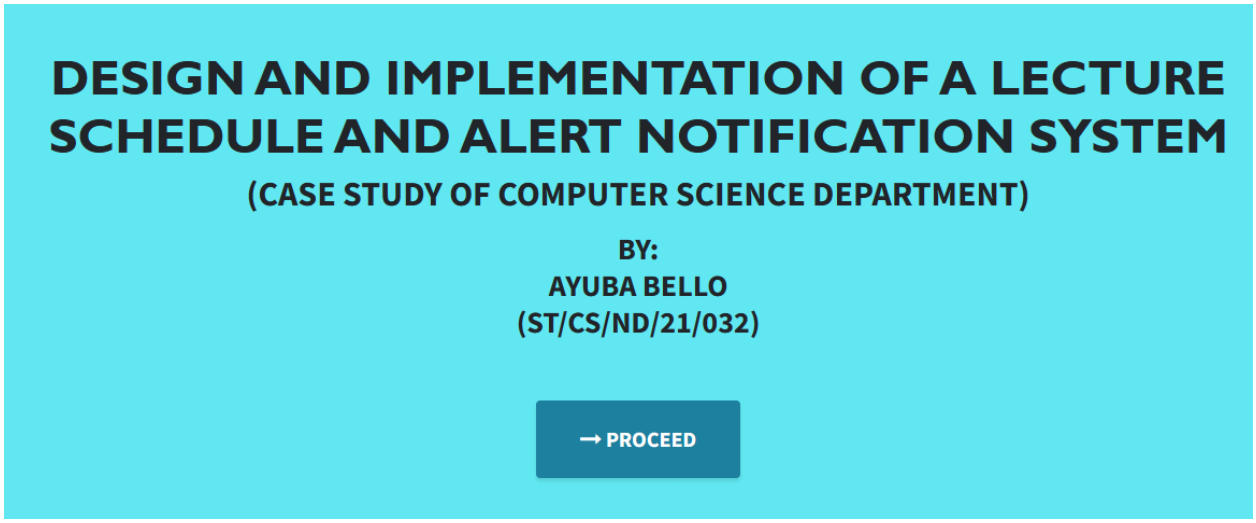
Yau, T. W., Yiu, K. T., & Leung, S. W. (2021). Enhancing the Tenant-Management Communication Process through Mobile Technology in Residential Property Management. *Journal of Real Estate Research*, 43(1), 49-78.

Zhang, J., Li, S., Yang, W., & Wang, X. (2020). Big Data Analytics in Property Management: *A Review. Smart Cities*, 3(4), 974-991.

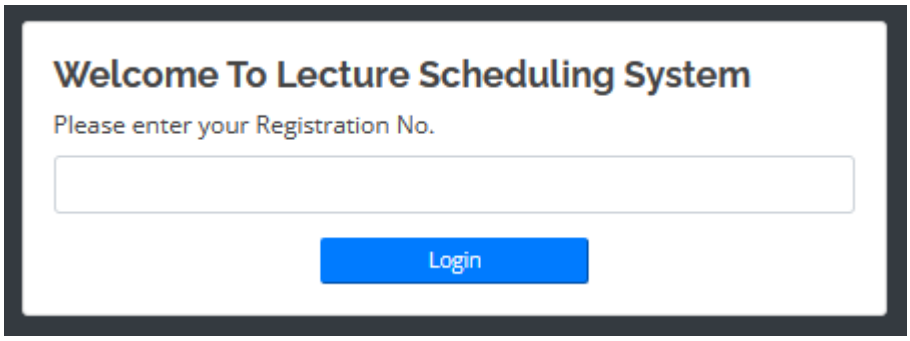
Zillow. (2021). *Zillow Group Consumer Housing Trends Report 2021*. Retrieved from https://www.zillow.com/report/2021/consumer-housing-trends/

APPENDIX A

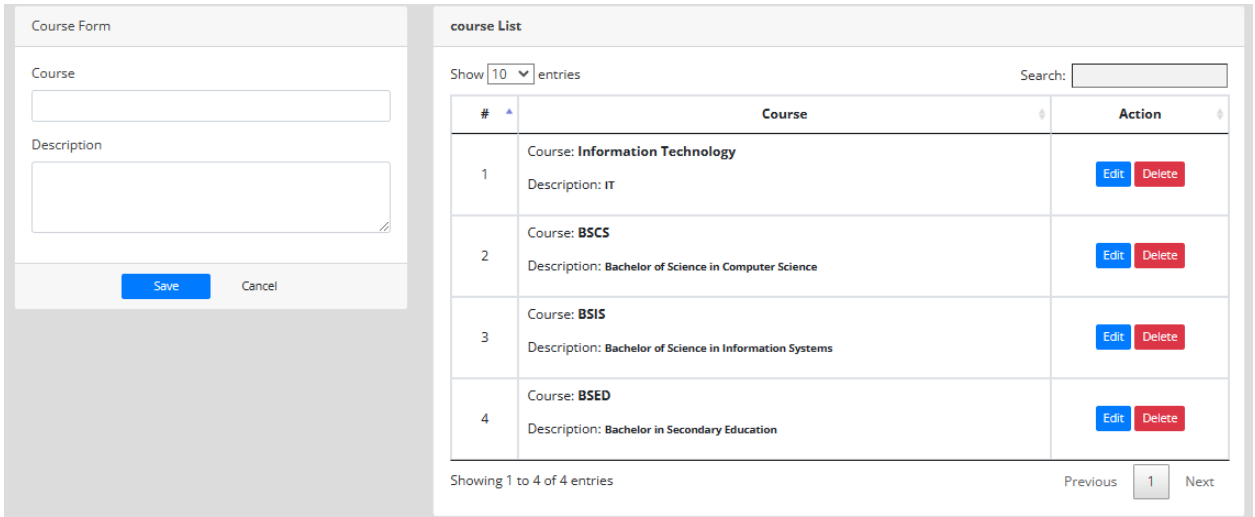
Welcome interface



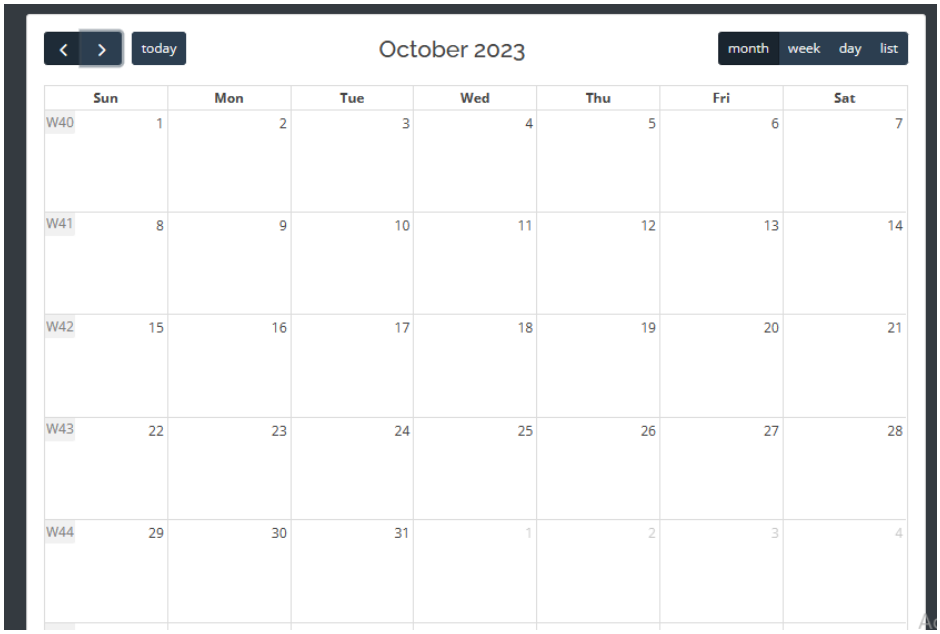
Login Interface



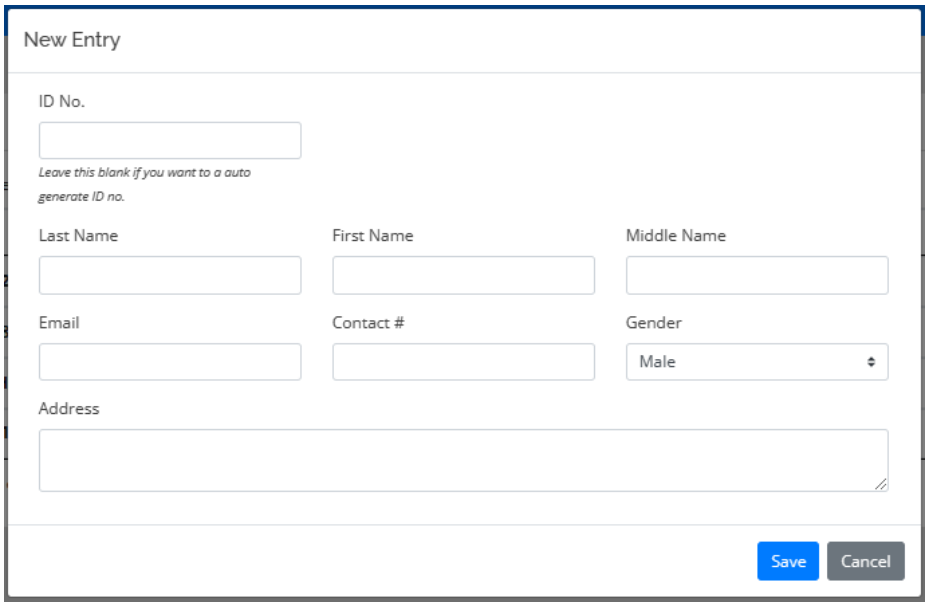
Courses Interface



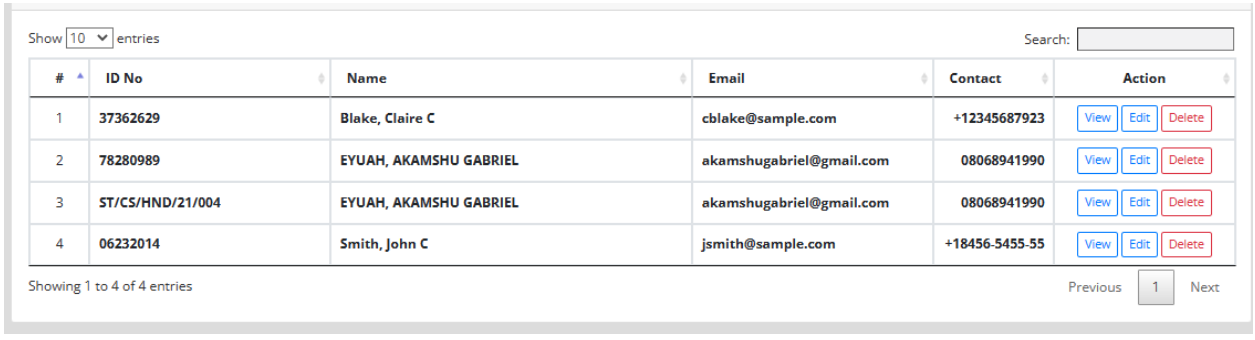
Lecture Scheduled Interface



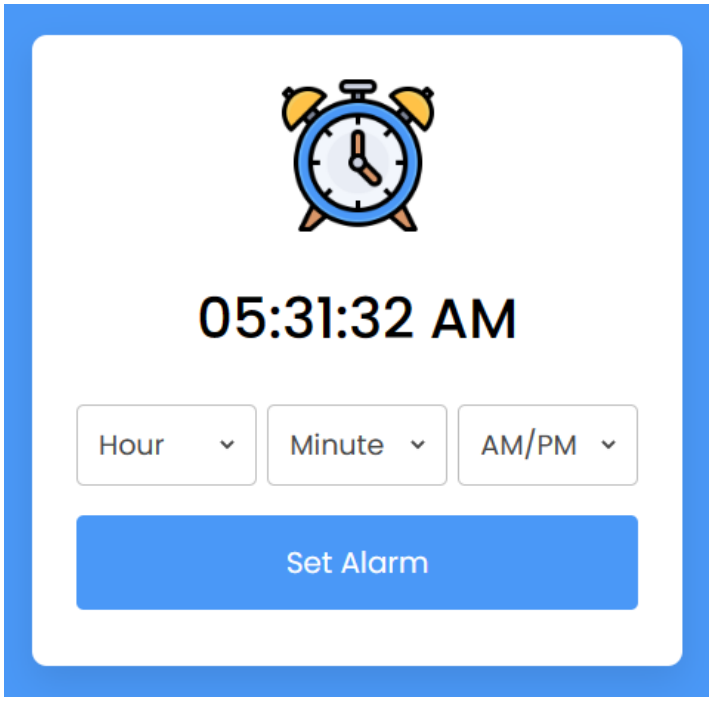
New Student Interface



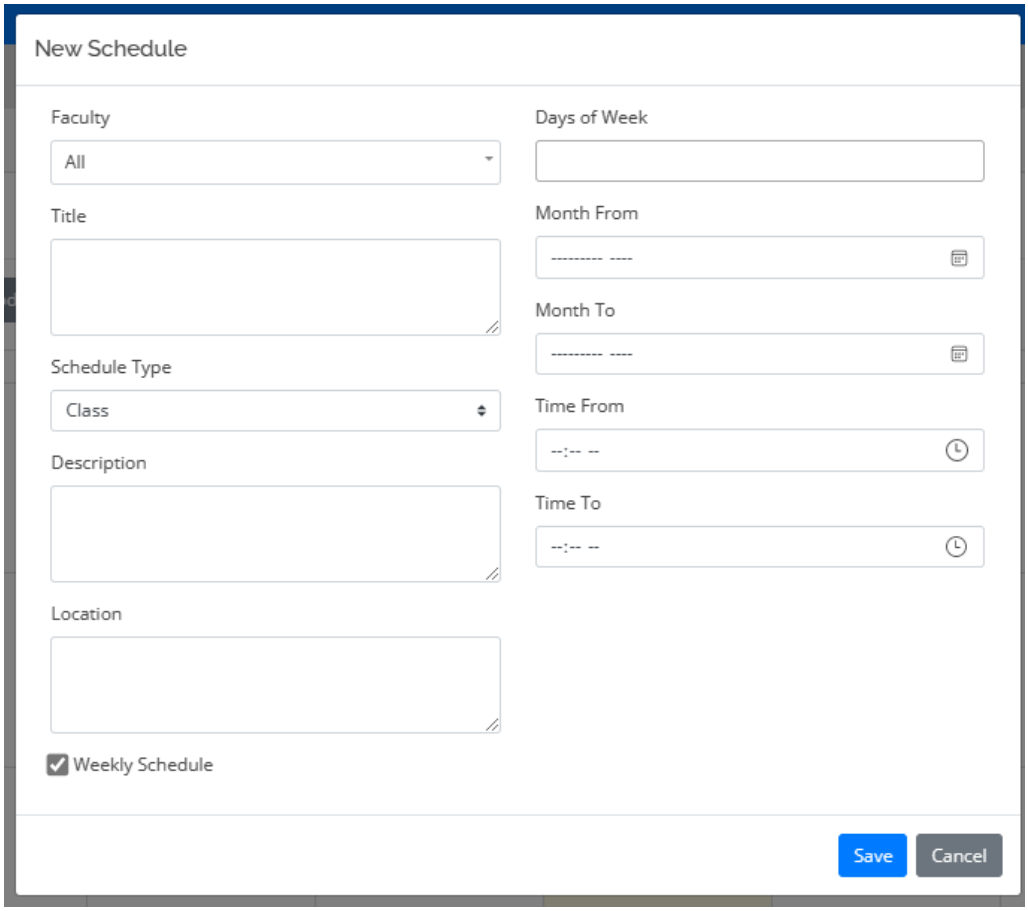
List of Students



Notification Interface



New Lecture Schedule Interface



APPENDIX B

PROGRAM CODE

```
<!DOCTYPE html>
<html lang="en">

  <head>

    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-
to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">

    <title>Antenatal Information System</title>

    <!-- Bootstrap Core CSS -->
    <link href="vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">

    <!-- Custom Fonts -->
    <link href="vendor/font-awesome/css/font-awesome.min.css" rel="stylesheet"
type="text/css">
    <link
href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,700,300ital
ic,400italic,700italic" rel="stylesheet" type="text/css">
    <link href="vendor/simple-line-icons/css/simple-line-icons.css"
rel="stylesheet">

    <!-- Custom CSS -->
    <link href="css/stylish-portfolio.min.css" rel="stylesheet">

  </head>

  <body id="page-top" style="background:rgb(79, 229, 240);">


    <!-- Header -->
    <header class="masthead d-flex">
      <div class="container text-center">

        <h1 class="mb-2" style="font-size: 45px; font-weight: bolder; font-
family: 'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS', sans-serif; text-
transform: uppercase;"><span style=" margin-top:15px;"> DESIGN AND IMPLEMENTATION
OF A LECTURE SCHEDULE AND ALERT NOTIFICATION SYSTEM </span> <br>  </h1>
        <h2 class="mb-3" style="">(CASE STUDY OF COMPUTER SCIENCE
DEPARTMENT)</h2>
        <h3>BY: <br> AYUBA BELLO<br>(ST/CS/ND/21/032)</h3> <br>


        </h3>

        <br>
```

```
        <strong><a class="btn btn-primary btn-xl js-scroll-trigger" href="care/"
style="font-size: 20px;"><span class="fa fa-long-arrow-right"></span>
PROCEED</a></strong>
              </div>
      <div class="overlay"></div>
    </header>



    <!-- Scroll to Top Button-->
    <a class="scroll-to-top rounded js-scroll-trigger" href="#page-top">
      <i class="fa fa-angle-up"></i>
    </a>

    <!-- Bootstrap core JavaScript -->
    <script src="vendor/jquery/jquery.min.js"></script>
    <script src="vendor/bootstrap/js/bootstrap.bundle.min.js"></script>

    <!-- Plugin JavaScript -->
    <script src="vendor/jquery-easing/jquery.easing.min.js"></script>

    <!-- Custom scripts for this template -->
    <script src="js/stylish-portfolio.min.js"></script>

  </body>

</html>

<!DOCTYPE html>
<html lang="en">
<?php
session_start();
include('admin/db_connect.php');
ob_start();
ob_end_flush();
?>
<head>
  <meta charset="utf-8">
  <meta content="width=device-width, initial-scale=1.0" name="viewport">

  <title>School Department Scheduling System</title>


<?php include('./header.php'); ?>
<?php
if(isset($_SESSION['login_id']))
header("location:index.php");

?>

</head>
<style>
    body{
        width: 100%;
        height: calc(100%);
        position:fixed;
    }
```

```
    #main{
        width: calc(100%);
        height: calc(100%);
        display:flex;
        align-items:center;
        justify-content:center
    }
    #login{

    }


</style>

<body>

  <main id="main" class=" bg-dark">
        <div id="login" class="col-md-4">
            <div class="card">
                <div class="card-body">

                    <form id="login-form" >
                      <h4><b>Welcome To Lecture Scheduling System</b></h4>
                        <div class="form-group">
                            <label for="id_no" class="control-label">Please enter
your Registration No.</label>
                            <input type="text" id="id_no" name="id_no"
class="form-control">
                        </div>
                        <center><button class="btn-sm btn-block btn-wave col-md-4
btn-primary">Login</button></center>
                    </form>
                </div>
            </div>
        </div>


  </main>

  <a href="#" class="back-to-top"><i class="icofont-simple-up"></i></a>

</body>
<script>
    $('#login-form').submit(function(e){
        e.preventDefault()
        $('#login-form
button[type="button"]').attr('disabled',true).html('Logging in...');
        if($(this).find('.alert-danger').length > 0 )
            $(this).find('.alert-danger').remove();
        $.ajax({
            url:'admin/ajax.php?action=login_faculty',
            method:'POST',
            data:$(this).serialize(),
            error:err=>{
                console.log(err)
        $('#login-form
button[type="button"]').removeAttr('disabled').html('Login');
```

```
            },
            success:function(resp){
                if(resp == 1){
                    location.href ='index.php';
                }else{
                    $('#login-form').prepend('<div class="alert alert-danger">ID
Number is incorrect.</div>')
                    $('#login-form
button[type="button"]').removeAttr('disabled').html('Login');
                }
            }
        })
    })
</script>
</html>

<!DOCTYPE html>
<html lang="en">

<?php session_start(); ?>
<head>
  <meta charset="utf-8">
  <meta content="width=device-width, initial-scale=1.0" name="viewport">

  <title>School Faculty Scheduling System</title>


<?php
  if(!isset($_SESSION['login_id']))
    header('location:login.php');
 include('./header.php');
 // include('./auth.php');
 ?>

</head>

<body>
  <?php include 'topbar.php' ?>
  <?php include 'navbar.php' ?>
  <div class="toast" id="alert_toast" role="alert" aria-live="assertive" aria-
atomic="true">
    <div class="toast-body text-white">
    </div>
  </div>
  <main id="view-panel" >
      <?php $page = isset($_GET['page']) ? $_GET['page'] :'home'; ?>
    <?php include $page.'.php' ?>


  </main>

  <div id="preloader"></div>
  <a href="#" class="back-to-top"><i class="icofont-simple-up"></i></a>

  <div class="modal fade" id="uni_modal" role='dialog'>
    <div class="modal-dialog modal-md" role="document">
```

```
      <div class="modal-content">
        <div class="modal-header">
        <h5 class="modal-title"></h5>
      </div>
      <div class="modal-body">
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-primary" id='submit'
onclick="$('#uni_modal form').submit()">Save</button>
        <button type="button" class="btn btn-secondary" data-
dismiss="modal">Cancel</button>
      </div>
      </div>
    </div>
  </div>
  <div class="modal fade" id="confirm_modal" role='dialog'>
    <div class="modal-dialog modal-md" role="document">
      <div class="modal-content">
        <div class="modal-header">
        <h5 class="modal-title">Confirmation</h5>
      </div>
      <div class="modal-body">
        <div id="delete_content"></div>
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-primary" id='confirm'
onclick="">Continue</button>
        <button type="button" class="btn btn-secondary" data-
dismiss="modal">Close</button>
      </div>
      </div>
    </div>
  </div>
  <div class="modal fade" id="viewer_modal" role='dialog'>
    <div class="modal-dialog modal-md" role="document">
      <div class="modal-content">
              <button type="button" class="btn-close" data-dismiss="modal"><span
class="fa fa-times"></span></button>
              <img src="" alt="">
      </div>
    </div>
  </div>
</body>
<script>
   window.start_load = function(){
    $('body').prepend('<di id="preloader2"></di>')
  }
  window.end_load = function(){
    $('#preloader2').fadeOut('fast', function() {
        $(this).remove();
      })
  }
 window.viewer_modal = function($src = ''){
    start_load()
    var t = $src.split('.')
    t = t[1]
    if(t =='mp4'){
```

```
      var view = $("<video src='"+$src+"' controls autoplay></video>")
    }else{
      var view = $("<img src='"+$src+"' />")
    }
    $('#viewer_modal .modal-content video,#viewer_modal .modal-content
img').remove()
    $('#viewer_modal .modal-content').append(view)
    $('#viewer_modal').modal({
            show:true,
            backdrop:'static',
            keyboard:false,
            focus:true
          })
          end_load()

}
  window.uni_modal = function($title = '' , $url='',$size=""){
    start_load()
    $.ajax({
        url:$url,
        error:err=>{
            console.log()
            alert("An error occured")
        },
        success:function(resp){
            if(resp){
                $('#uni_modal .modal-title').html($title)
                $('#uni_modal .modal-body').html(resp)
                if($size != ''){
                    $('#uni_modal .modal-dialog').addClass($size)
                }else{
                    $('#uni_modal .modal-
dialog').removeAttr("class").addClass("modal-dialog modal-md")
                }
                $('#uni_modal').modal({
                  show:true,
                  backdrop:'static',
                  keyboard:false,
                  focus:true
                })
                end_load()
            }
        }
    })
}
window._conf = function($msg='',$func='',$params = []){
     $('#confirm_modal #confirm').attr('onclick',$func+"("+$params.join(',')+")")
     $('#confirm_modal .modal-body').html($msg)
     $('#confirm_modal').modal('show')
  }
   window.alert_toast= function($msg = 'TEST',$bg = 'success'){
      $('#alert_toast').removeClass('bg-success')
      $('#alert_toast').removeClass('bg-danger')
      $('#alert_toast').removeClass('bg-info')
      $('#alert_toast').removeClass('bg-warning')

    if($bg == 'success')
```

```
      $('#alert_toast').addClass('bg-success')
    if($bg == 'danger')
      $('#alert_toast').addClass('bg-danger')
    if($bg == 'info')
      $('#alert_toast').addClass('bg-info')
    if($bg == 'warning')
      $('#alert_toast').addClass('bg-warning')
    $('#alert_toast .toast-body').html($msg)
    $('#alert_toast').toast({delay:3000}).toast('show');
  }
  $(document).ready(function(){
    $('#preloader').fadeOut('fast', function() {
        $(this).remove();
      })
  })
  $('.datetimepicker').datetimepicker({
      format:'Y/m/d H:i',
      startDate: '+3d'
  })
  $('.select2').select2({
    placeholder:"Please select here",
    width: "100%"
  })
</script>
</html>
```