**WEB-BASED LECTURE HALL MANAGEMENT SYSTEM**

**RONY MWANGI NJOROGE**

**CT100/G/8673/20**

This project proposal was presented in partial fulfillment of Kirinyaga University's requirements for the award of the degree of Bachelor of Science in Information and Technology.

# DEDICATION

I would like to dedicate this artwork to the individuals who have inspired me and supported me throughout this journey. Their unwavering belief in my abilities has been the driving force behind my accomplishments, and I am forever indebted to their unwavering support.

# ACKNOWLEDGMENT

With a heart brimming with gratitude, I extend my deepest appreciation to the divine hand of God, the Almighty, for bestowing upon me the intellect, perseverance, and unwavering faith that have enabled me to embark on this academic journey and bring this research proposal to fruition. His divine blessings and constant guidance have been the pillars of my strength, illuminating my path towards knowledge and understanding.

I am also profoundly grateful to my esteemed lecturers, whose unwavering dedication to teaching and their expertise in their respective fields have been instrumental in shaping my academic foundation and nurturing my intellectual growth. Their passion for knowledge, their commitment to excellence, and their unwavering belief in my potential have inspired me to strive for academic excellence.

Furthermore, I would like to express my sincere gratitude to my fellow course mates, for their encouragement, and willingness to share their knowledge and insights throughout my academic journey. Their collaborative spirit, supportive nature, and shared passion for learning have created a stimulating and enriching learning environment that has greatly contributed to my academic growth and intellectual development.

# DECLARATION

I declare that this is my original work and has not been presented in any University for a degree or any consideration of any Certificate.

Student name: Rony mwangi njoroge

Signature………………………………..

This project documentation has been submitted with my approval as the project supervisor

Name: DR Ephuntus

Signature……………………………… Date………………

Department of Information Technology

# ABSTRACT

This project presents the design and development of a web-based university lecture hall management system. The system is designed to automate the process of lecture hall scheduling and allocation. The system has two main modules: a lecturer module and a student module. The lecturer module allows lecturers to view the availability of lecture halls, book lecture halls, their capacity and manage their bookings. The student module allows students to view the timetable for lecture hall bookings, and to receive notifications of any changes to the timetable. The system is implemented using a three-tier architecture, with a presentation tier, an application tier, and a data tier. The web-based university lecture hall management system is a valuable tool for universities and colleges. The system can help to improve the efficiency, transparency, and communication of lecture hall management.

Contents

[DEDICATION 2](#_Toc163388969)

[ACKNOWLEDGMENT 3](#_Toc163388970)

[DECLARATION 4](#_Toc163388971)

[ABSTRACT 5](#_Toc163388972)

[LIST OF FIGURES 9](#_Toc163388973)

[LIST OF TABLES 10](#_Toc163388974)

[CHAPTER 1: INTRODUCTION. 11](#_Toc163388975)

[1.0 introduction 11](#_Toc163388976)

[1.1Background of the Study 11](#_Toc163388977)

[1.2 Current/existing system 11](#_Toc163388978)

[1.4 Proposed system 13](#_Toc163388979)

[1.5 Purpose of the study 13](#_Toc163388980)

[1.6 Main Objective 13](#_Toc163388981)

[1.6.1 other objectives 13](#_Toc163388982)

[1.7 Justification 13](#_Toc163388983)

[1.8 Scope 14](#_Toc163388984)

[1.10 Significance of the study 15](#_Toc163388985)

[1.11 Operational definition of terms 16](#_Toc163388986)

[CHAPTER 2: LITERATURE REVIEW. 17](#_Toc163388987)

[2.0 Introduction. 17](#_Toc163388988)

[2.1 Online Halls Reservation System at the University of Peradeniya 17](#_Toc163388989)

[Figure 2.1 Online Halls Reservation System at the University of Peradeniya 18](#_Toc163388990)

[2.2 University of Otago, New Zealand Lecture room booking and timetabling 19](#_Toc163388991)

[Figure 2.2University of Otago, New Zealand Lecture room booking and timetabling 20](#_Toc163388992)

[2.3 Room and lecture hall scheduling in University of Hamburg Germany(Raumplanung) 20](#_Toc163388993)

[Figure 2.3 Room and lecture hall scheduling in University of Hamburg Germany( Raumplanung) 22](#_Toc163388994)

[2.4 Accruent EMS Room Booking System 22](#_Toc163388995)

[Figure 2.4 Accruent EMS Room Booking System 23](#_Toc163388996)

[2.5 Mazevo Room Booking System 23](#_Toc163388997)

[Figure 2.5 Mazevo Room Booking System 25](#_Toc163388998)

[2.6 GAPS/LACUNA'S 25](#_Toc163388999)

[2.8 Context Diagram /Conceptual framework 25](#_Toc163389000)

[Figure 2.8 Conceptual framework 25](#_Toc163389001)

[2.7 Summary 25](#_Toc163389002)

[3.0 INTRODUCTION 26](#_Toc163389003)

[3.1.1 Requirements gathering and analysis 26](#_Toc163389004)

[3.1.2 Planning 26](#_Toc163389005)

[3.1.3 Development 27](#_Toc163389006)

[3.1.4 Testing 27](#_Toc163389007)

[3.1.5 Deployment 27](#_Toc163389008)

[3.1.6 Feedback and retrospective 27](#_Toc163389009)

[3.2 CONCLUSION 27](#_Toc163389010)

[CHAPTER 4: DESIGN 28](#_Toc163389011)

[**4.0 INTRODUCTION** 28](#_Toc163389012)

[**4.1 REQUIREMENTS** 28](#_Toc163389013)

[**4.2 CONTEXT LEVEL DIAGRAM** 28](#_Toc163389014)

[**figure 4.1: context level diagram** 28](#_Toc163389015)

[**4.3 INPUT DESIGN** 29](#_Toc163389016)

[**Figure 4.2 Admin log in** 29](#_Toc163389018)

[**Figure 4.3: Student login** 29](#_Toc163389019)

[**4.4 PROCESS DESIGN** 30](#_Toc163389020)

[**4.5 DATABASE DESIGN** 31](#_Toc163389021)

[**Table 4.1 Admin table** 31](#_Toc163389022)

[**Table 4.2 student details** 31](#_Toc163389023)

[**Table 4.3 lecturer table** 32](#_Toc163389024)

[**4.6 OUTPUT DESIGN** 32](#_Toc163389025)

[**Figure 4.5: process design** 32](#_Toc163389026)

[**4.7 CONCLUSION** 32](#_Toc163389027)

[5.0 INTRODUCTION 34](#_Toc163389028)

[**5.1 Unit Testing** 34](#_Toc163389029)

[**5.2 Integration Testing** 34](#_Toc163389030)

[**5.3 System Testing** 35](#_Toc163389031)

[**5.4 Database Testing** 35](#_Toc163389032)

[**5.5 Implementation Requirements** 35](#_Toc163389033)

[**5.6 Coding Tools** 35](#_Toc163389034)

[**5.7 System Home Page and Other Relevant Pages** 36](#_Toc163389035)

[**Figure 5.1 home page** 36](#_Toc163389036)

[**Figure 5.2 Log in page** 37](#_Toc163389037)

[**5.8 Conclusion** 38](#_Toc163389038)

[CHAPTER 6: CONCLUSION AND RECOMMENDATIONS 39](#_Toc163389039)

[**6.0 Introduction** 39](#_Toc163389040)

[**6.1 Conclusions** 39](#_Toc163389041)

[**6.1 Recommendation** 39](#_Toc163389042)

[**6.3 Future Work** 40](#_Toc163389043)

[**6.4 References** 41](#_Toc163389044)

[**6.5** Appendices 42](#_Toc163389045)

# LIST OF FIGURES

**Figure 2.1** **Online Halls Reservation System** at the University of Peradeniya16

**Figure 2.2 University of Otago, New Zealand Lecture room booking and timetabling** 18

**Figure 2.3 Room and lecture hall scheduling in University of Hamburg Germany**20

[**Figure 2.4** Accruent EMS Room Booking System](#_Toc155595279) 21

**Figure 2.5 Mazevo Room Booking System**22

**Figure 4.1 Context diagram** 28

**Figure 4.2 Admin log in**29

**Figure 4.3 lecturer log in**29

**Figure 4.4 Student log in**29

**Figure 4.5 process design**30

**Figure 4.6Admin dashboard**32

**Figure 5.1 home page**36

**Figure 5.2 Log in**37

**Figure 5.3 Student dashboard**37

**Figure 6.1 home page**42

# LIST OF TABLES

**Table 4.1 :Admin table** 31

**Table 4.2: student table** 32

**Table 4.3: lecturer table** 33

**Table 4.4: Admin table** 34

# CHAPTER 1: INTRODUCTION.

# 1.0 introduction

This introductory chapter **provided** a comprehensive overview of the web-based lecture hall management system (LHMS) project, outlining its significance, objectives, and scope. It **delved** into the background of lecture hall management challenges and the need for an automated solution. The chapter **also explored** the existing LHMS landscape, identifying key features and challenges of these systems. Additionally, it **introduced** the proposed LHMS, highlighting its unique features and potential benefits.

# 1.1Background of the Study

Traditionally, lecture hall management was a manual process. This was often time-consuming and error-prone, especially for large institutions with many lecture halls. A web-based university lecture hall management system could automate the process of lecture hall scheduling and allocation, providing a platform for lecturers and students to interact and share information. A web-based university lecture hall management system was a valuable tool that helped universities and colleges improve the efficiency, transparency, and communication of lecture hall management. It also could be used to enhance the teaching and learning experience for lecturers and students alike.

# 1.2 Current/existing system

Lecture hall management systems (LHMS) became increasingly popular in educational institutions, as they offered a number of benefits over traditional manual methods of hall allocation and scheduling. LHMSs could help improve efficiency, reduce conflicts, and enhance transparency. However, there were also a number of challenges that LHMSs faced, such as integration with existing systems, user interface and accessibility, data management and security, scalability and flexibility, real-time updates and conflict resolution, accessibility for students with disabilities, mobile compatibility, reporting and analytics, user feedback and continuous improvement, and cost-effectiveness.

**1.3** **Problem Statement**

Classroom allocation at many universities was previously a manual process that was coordinated by representatives from academic departments and staff from the registration office. This manual process was time-consuming, tedious, and prone to human error. Each representative submitted information about the courses their department offered for a given semester to the registration office, which might include specific classroom requests. Most departments were given a specific set of classrooms that they could schedule. The manual process required a lot of time and coordination to finalize the classroom allocation. However, the actual enrollment of a class was not known until the beginning of the semester, which was an important factor in selecting a classroom for a given course. This often led to changes in the classroom allocation early on in the semester. To make matters worse, there was no procedure in place to compare alternative allocation strategies. Therefore, in most cases, the allocation of classrooms was done based on previous semesters' allocations and the expertise of both the representatives and the registration staff.

# 1.4 Proposed system

This web-based lecture hall management system **was** a software application that **helped** universities and other educational institutions to manage the booking and scheduling of lecture halls. The system **was** typically hosted on a web server and **was** accessed by authorized users from any device with an internet connection.

# 1.5 Purpose of the study

The purpose of this study of the proposed web-based lecture hall management system **was** to evaluate the system’s potential to improve the efficiency and effectiveness of lecture hall management for universities and other educational institutions.

# 1.6 Objectives

# Main Objective

1. To develop a lecture hall management system that enhances the efficiency and effectiveness of lecture hall scheduling, allocation, and utilization.

# 1.6.1 other objectives

1. To develop an interface that captions available lecturer halls and their capacity.
2. **To develop a system that will enhance User Experience through Streamlined Interface.**
3. To develop a system that will improve communications by notifying stakeholders about updates.

# 1.7 Justification

The proposed web-based lecture hall management system **had** the potential to improve the efficiency and effectiveness of lecture hall management for universities and other educational institutions. Currently, lecture hall management **was** often a manual and time-consuming process. Administrative staff **had to** manually track lecture hall bookings and schedules, and students **had to** often contact administrative staff to book lecture halls. This **led** to errors and inefficiencies. The proposed system **would have automated** many of the tasks involved in lecture hall management, freeing up administrative staff to focus on other tasks and making it easier for students to book lecture halls. The system **would have also provided** real-time information about lecture hall availability, which **would have helped** to improve the efficiency of lecture hall utilization.

# 1.8 Scope

**The scope of the proposed web-based lecture hall management system was to provide a comprehensive solution for managing the booking and scheduling of lecture halls. The system was able to meet the needs of all stakeholders, including students, faculty, and administrative staff. The system had features like lecture hall booking, lecture hall scheduling, lecture hall management and user management**

**1.9 limitations**

**1. Financial Constraints: The design was achieved but not without some financial involvements. One had to pay for the computer time. Also, the typing and planning of the work had its own financial involvements.**

2. High programming Technique: The programming aspect of this project posed a lot of problematic bugs that took some days to solve. Problems such as database connections were encountered

3. Few Literature Sources: The topic, though seeming to be a common term, was not a popular topic to surf from the Internet. It had fewer literature sources.

# 1.10 Significance of the study

**The significance of the study of the proposed web-based lecture hall management system was that it had the potential to improve the efficiency and effectiveness of lecture hall management for universities and other educational institutions.**

At that time, lecture hall management was often a manual and time-consuming process. Administrative staff had to manually track lecture hall bookings and schedules, and students often had to contact administrative staff to book lecture halls. This led to errors and inefficiencies.

The proposed system aimed to automate many of the tasks involved in lecture hall management, freeing up administrative staff to focus on other tasks and making it easier for students to book lecture halls. The system also provided real-time information about lecture hall availability, improving the efficiency of lecture hall utilization.

Overall, the proposed system had the potential to improve the efficiency and effectiveness of lecture hall management for universities and other educational institutions. This would have benefited students, administrative staff, and faculty.

# ****1.11 Operational definition of terms****

**Lecture hall: A room specially designed for lectures, typically with tiered seating and a projector/screen.**

**Lecture hall booking:** The process of reserving a lecture hall for a specific date and time.

**Lecture hall scheduling:** The process of creating a timetable for the use of lecture halls.

**Lecture hall management:** The process of overseeing and maintaining lecture halls, including booking, scheduling, and maintenance.

**User management:** The process of creating, updating, and deleting user accounts, as well as assigning user roles and permissions.

**Integration with other systems:** The process of connecting the lecture hall management system to other systems, such as the student information system or course management system.

**Reporting:** The generation of reports on lecture hall usage, such as which halls are the most popular and which halls are underutilized.

# CHAPTER 2: LITERATURE REVIEW.

# 2.0 Introduction.

This chapter mainly considered the similar systems that were available on the internet. There were some online systems that used to handle the lecture hall scheduling and timetabling in universities. The manual timetabling system used in the institution was time-consuming and energy-sapping, leading to the loss of information, general apathy, clashes, and wastage of resources. The solution to these problems was a lecture hall scheduling with a web-based timetabling system. Computerization made it easy for a university to do its lecture hall scheduling and timetabling process effectively. The currently available system was difficult to customize according to the fluctuations of the allocation process. Having an implementation system was user-friendly, and fewer technical skills were needed to operate the system. The ability to search for locations and staff based on different criteria was an essential feature. There were many lecture hall scheduling and timetabling systems available all over the world. These systems were used to improve the efficiency of the scheduling process. The following were similar systems:

# 2.1 ****Online Halls Reservation System**** at the University of Peradeniya

Department of Computer Science (2005) developed as web based lecture hall management system of University of Peradeniya using, HTML, CSS, and servlets at the front end and PHP and MySQL at the back end.

The system allows users to book lecture halls and other facilities at the university online. Users can search for available facilities, view their booking history, and cancel bookings. The system also provides administrators with the ability to manage user accounts, add and remove facilities, and view booking reports.

The system has a number of strengths, including its convenience, efficiency, transparency, flexibility, and affordability. It is also easy to use, and it can be accessed from anywhere in the world.However, the system also has some weaknesses. For example, it is not fully integrated with other university systems, such as the student information system and the financial management system. This can lead to duplication of data and inefficiencies. Additionally, the system does not provide a way for users to track the status of their reservation requests in real time, and it does not provide a way for users to make reservations for multiple halls or facilities at the same time.

There are a number of ways to improve the Online Halls Reservation System. For example, the system could be integrated with other university systems to eliminate duplication of data and improve efficiency. Additionally, the system could be enhanced to provide real-time status tracking and multi-hall/facility reservations.

# Figure 2.1 ****Online Halls Reservation System**** at the University of Peradeniya

# 2.2 University of Otago, New Zealand Lecture room booking and timetabling

**Was developed by the University of Otago Information Technology Services (ITS) in (2006) using HTML, java script css and bootstrap at the front end and java PostgreSQL and spring board for back end.**

**The system is a web-based system that allows staff to book lecture rooms and other facilities online. The system also allows staff to create and manage timetables for courses and other events. It had the following features; Lecture room booking, Staff can search for available lecture rooms and book them online. The system allows staff to book lecture rooms for a variety of purposes, such as classes, meetings, and events. Timetabling: Staff can create and manage timetables for courses and other events. The system allows staff to create timetables for a variety of purposes, such as weekly classes, semester-long courses, and one-off events.**

**The Lecture Room Booking and Timetabling System at the University of Otago provides users with the ability to View the availability of lecture rooms and other teaching spaces, book lecture rooms and other teaching spaces, generate and view timetables and Manage bookings and timetables.**

**The Lecture Room Booking and Timetabling System at the University of Otago has the following strengths: It is comprehensive and feature-rich, providing users with all the tools they need to book and manage lecture rooms and other teaching spaces, as well as to generate and view timetables and It is easy to use, with a user-friendly interface that makes it easy for users to book lecture rooms and other teaching spaces, generate and view timetables, and manage bookings and timetables. However the system has some weaknesses such as It is not fully integrated with the University of Otago's student information system, which means that users have to enter some information into the Lecture Room Booking and Timetabling System that is already stored in the student information system and It does not provide a way for users to book lecture rooms and other teaching spaces for multiple semesters at the same time.**

**The Lecture Room Booking and Timetabling System at the University of Otago could be improved by addressing the following gaps: Integration with the University of Otago's student information system, Multi-semester booking and real-time status tracking.**

# ****Figure 2.2University of Otago, New Zealand Lecture room booking and timetabling****

# ****2.3 Room and lecture hall scheduling in University of Hamburg Germany(Raumplanung)****

**The room and lecture hall scheduling system at the University of Hamburg, Germany was developed in (2005) by the university's IT department. The system is called Raumplanung (room planning) and is used by students, faculty, and staff to book lecture halls and other facilities for classes, meetings, and events. The front end was developed using HTML, CSS, PHP and java script while the back end was created by the use of PHP and MySQL.**

**The system had the following features; Check the availability, Carry out preliminary planning and make arrangement with lecturer , Make booking request, Wait for confirmation and Check booking entry.**

**The strengths of this system lie in its centralized nature and ease of use. The online form makes I easy for users to book rooms after checking the availability. However there are some weaknesses such as Not fully integrated with other university systems and Real-time status tracking.**

**The Room and Lecture Hall Scheduling system at the University of Hamburg is a comprehensive and feature-rich system, but there are some areas where it could be improved. One gap is that it is not fully integrated with other university systems, such as the student information system and the financial management system. This could be improved to improve efficiency and reduce duplication of data**

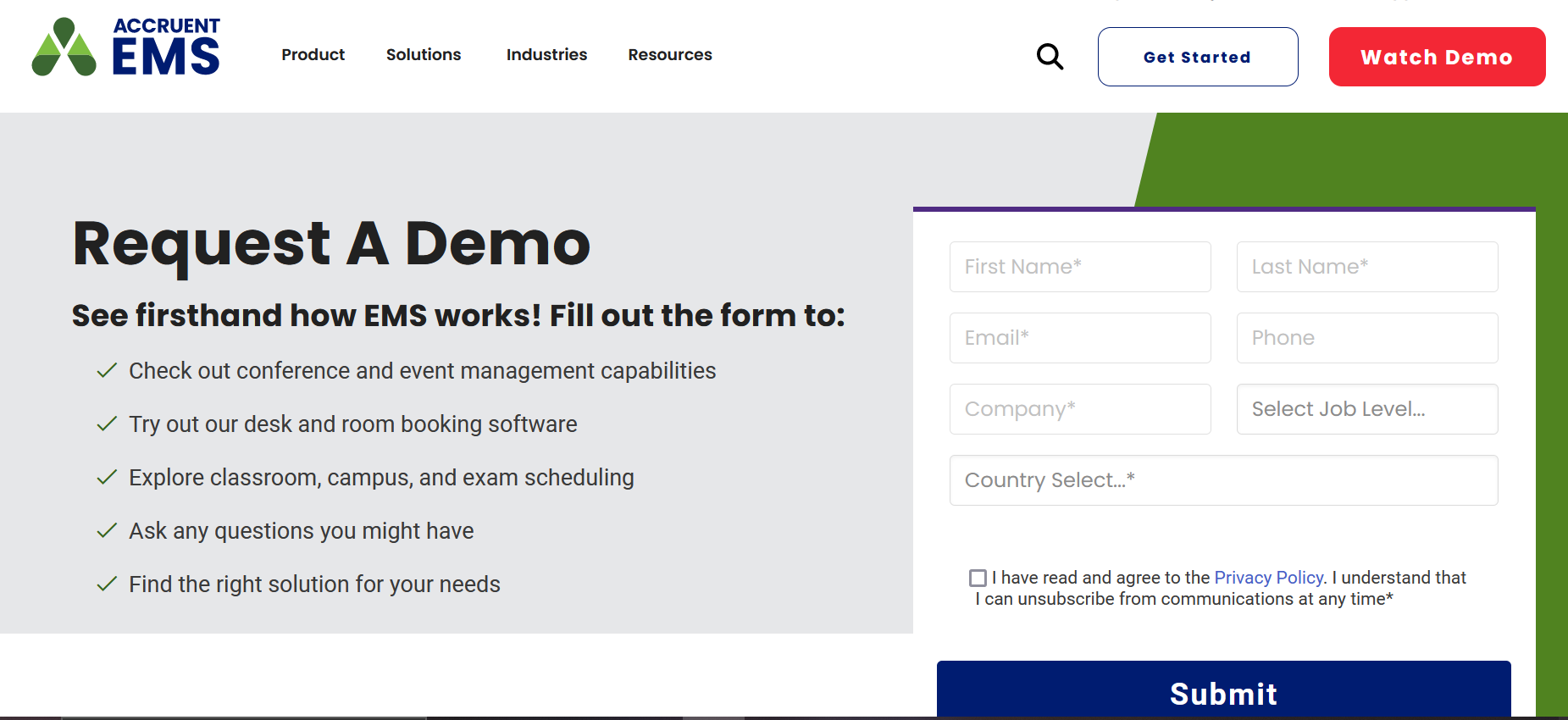
# ****Figure 2.3 Room and lecture hall scheduling in University of Hamburg Germany(** **Raumplanung)****

# ****2.4 Accruent EMS Room Booking System****

**Accruent EMS is developed by Accruent (2002).Accruent EMS is built on a modern, scalable architecture that can be deployed on-premises or in the cloud. The system is also compatible with a variety of other software applications, such as Microsoft Outlook and Microsoft Teams.**

**Accruent EMS is a comprehensive software solution that helps organizations manage their meeting rooms effectively. It provides a user-friendly interface for employees to book meeting rooms, check availability, and manage their reservations. The system also offers a variety of features to help organizations optimize their space utilization and streamline their meeting processes.**

**Accruent EMS is a comprehensive space and resource management software that offers a variety of benefits and drawbacks. On the positive side, Accruent EMS provides real-time room availability, online room booking, resource management, and reporting and analytics. These features streamline space utilization and meeting processes, making it easier for employees to find meeting spaces, book rooms, manage shared resources, and gain valuable insights into space usage. However, Accruent EMS also has some limitations. It can be costly to implement and maintain, and its complexity may necessitate employee training. Additionally, the software lacks features like parking space booking and calendar integration, has limited software integrations, and can be slow at times, making support from Accruent challenging.**

****

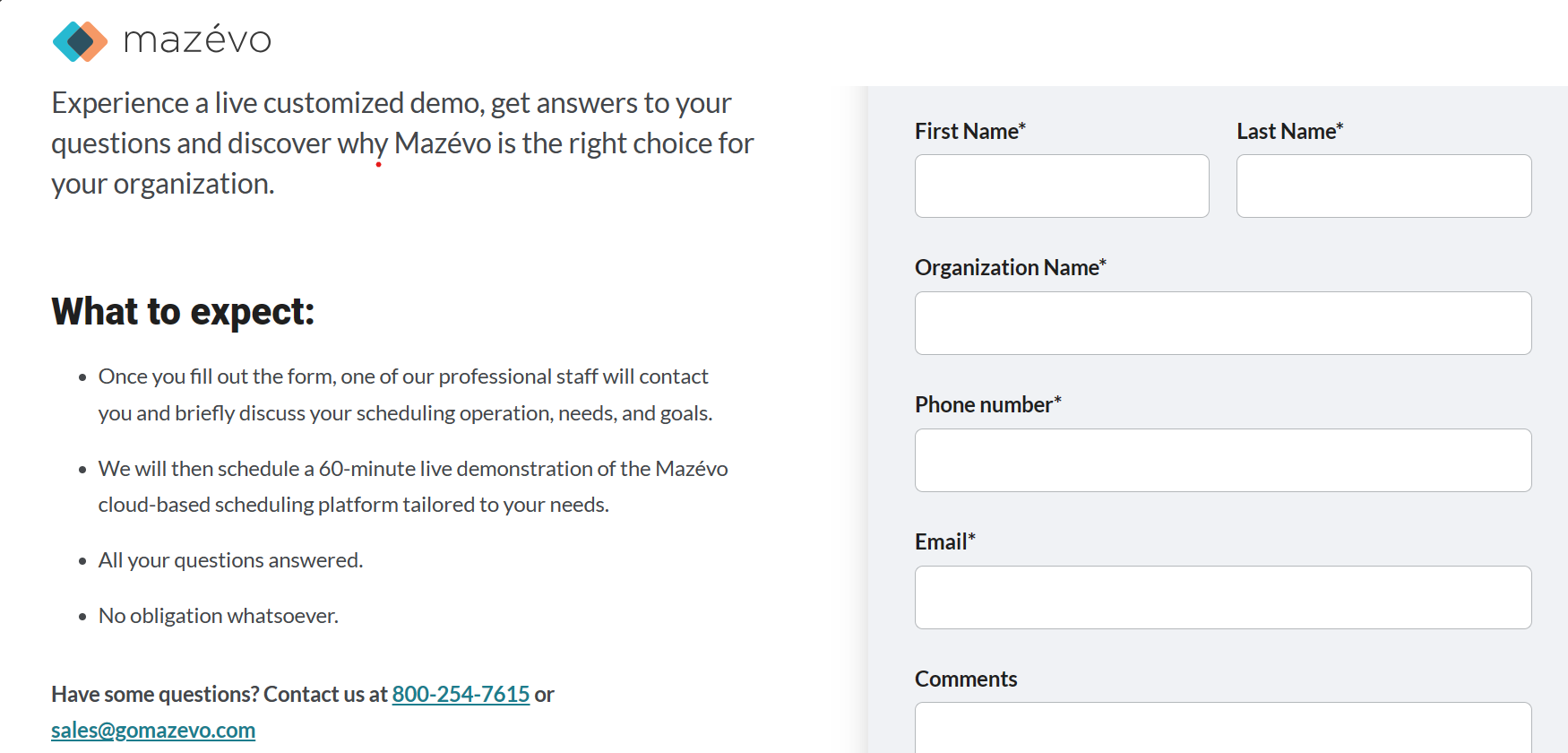
# ****Figure 2.4 Accruent EMS Room Booking System****

# ****2.5 Mazevo Room Booking System****

**Dean Evans (2019.), the founder of Dean Evans and Associates and creator of the EMS Scheduling Software system, developed Mazévo. The specific technologies used to develop Mazevo Room Booking System are not publicly available. However, based on the system's features and capabilities, it is likely that it was built using a combination of cloud-based technologies, such as Amazon Web Services (AWS) or Microsoft Azure, and web development technologies, such as HTML, CSS, JavaScript, and PHP. The system may also use other technologies, such as databases, API s, and third-party integration, to provide its functionality.**

**Mazevo offers built-in room request and approval processes that handle all event details such as room setups, AV, and catering. It also allows users to keep track of all event charges if they charge for their space and services.**

**Its strengths include a user-friendly interface, clear visibility for stakeholders, and efficient room request and approval processes. Additionally, it facilitates cost tracking for organizations that charge for space and services. On the other hand, the system has been criticized for its over-sensitivity to room specifications, which may lead to the disqualification of certain events based on arbitrary parameters. This could hinder the flexibility of scheduling and potentially limit the range of events that can be accommodated.**

****

# ****Figure 2.5 Mazevo Room Booking System****

# ****2.6 GAPS/LACUNA'S****

**This literature review on lecture hall management systems was limited in a number of ways. First, there was a lack of research on the use of machine learning to improve the efficiency and effectiveness of lecture hall booking. Second, there was a lack of research on the integration of lecture hall booking systems with other university systems, such as the student information system and the course management system. Third, there was a lack of research on the accessibility and usability of lecture hall booking systems for users with disabilities.**

# ****2.8 Context Diagram /Conceptual framework****

lecturer

system

student

# ****Figure 2.8 Conceptual framework****

# ****2.7 Summary****

**Web-based lecture hall scheduling and timetabling systems were very effective for universities because they handled a large number of students and lecturers in various ways. They automated the entire allocation process. The above section mentioned several systems, each with different features.**

**CHAPTER 3: METHODOLOGY**

# ****3.0 INTRODUCTION****

**This chapter presented the systematic strategies, techniques, and steps that the developer used to design, develop, and implement the innovative system. It offered a detailed road map of how the theoretical concepts were transformed into a functional, user-friendly, and efficient management tool. From data collection methods and analysis procedures to system design and testing protocols, this chapter acted as the blueprint for success. It not only showed the thorough understanding of the project’s requirements but also the commitment to delivering a high-quality, web-based Lecture Hall Management System that met the needs of the users. This chapter guided the reader through the exciting journey of discovery, innovation, and technological advancement.**

**3.1 Agile Methodology**

**The Agile methodology was an iterative and incremental approach to software development that emphasized flexibility, collaboration, and customer satisfaction. It was based on the Agile Manifesto, which valued individuals and interactions, working software, customer collaboration, and responding to change.**

# ****3.1.1 Requirements gathering and analysis****

The developer met with stakeholders to understand their needs and pain points. The developer also reviewed existing lecture hall management systems to identify best practices and areas for improvement. Once the developer had a good understanding of the requirements, they prioritized them and developed a high-level plan for the project.

# ****3.1.2 Planning****

The developer created a detailed plan for each sprint, including the requirements to be implemented, the tasks to be completed, and the estimated time for each task. The plan also included a schedule for the sprint and a list of resources needed.

# ****3.1.3 Development****

During the sprint, the developer worked to develop and deliver the requirements that were selected in the planning meeting. The developer used a variety of development tools and technologies, such as Bootstrap and Python Django. The developer also worked closely with stakeholders to ensure that the system met their needs.

# ****3.1.4 Testing****

Once the developer developed the requirements, they tested them to ensure that they met the requirements and were free of defects. The developer used a variety of testing methods, such as unit testing, integration testing, and system testing.

# ****3.1.5 Deployment****

Once the developer tested the requirements and confirmed that they met the requirements, they deployed them to production. This involved making the system available to users and providing support.

# ****3.1.6 Feedback and retrospective****

At the end of each sprint, the developer gathered feedback from users and stakeholders. This feedback was used to improve the system and the development process. The developer also held a retrospective meeting to discuss what went well during the sprint and what could be improved.

# ****3.2 CONCLUSION****

**This chapter introduced the Agile methodology and how it was used to develop the web-based lecture hall management system (WLHMS). The Agile methodology was a flexible, responsive, quality-focused, collaborative, and continuous improvement approach to software development. It was a good fit for the WLHMS project because it allowed the team to respond quickly to changes in the requirements and to learn from their mistakes. The Agile methodology was used to develop the WLHMS in a way that was efficient, effective, and met the needs of the users.**

# CHAPTER 4: DESIGN

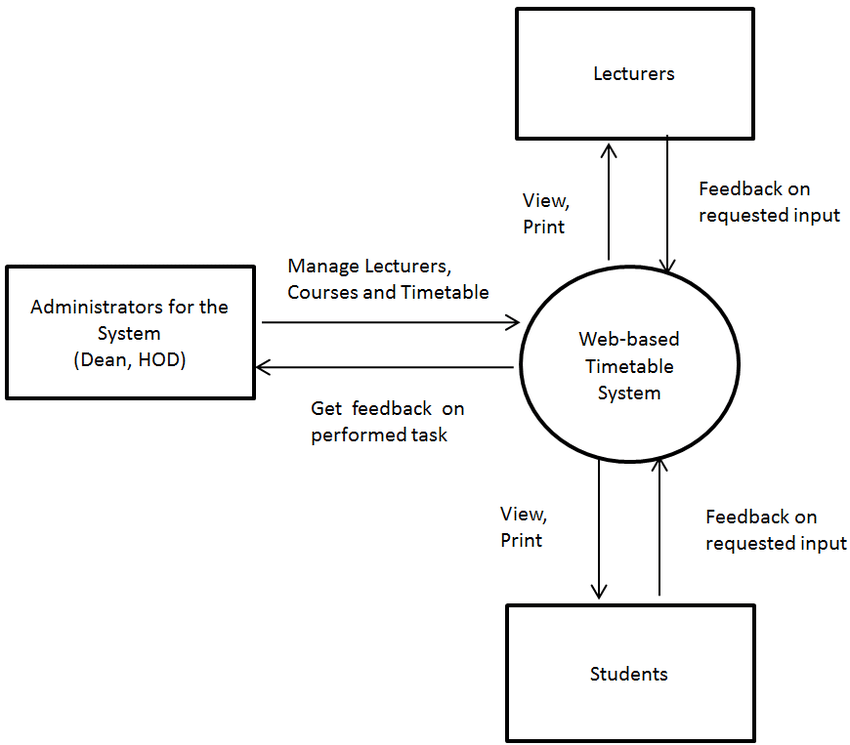
### **4.0 INTRODUCTION**

**This chapter described what the chapter would do and the functionalities it would provide, therefore the lecture hall management system could be able to create, store and retrieve data from the data base. The system enabled the administrator to input data of all lecture halls available in the institution.**

### **4.1 REQUIREMENTS**

**Requirements included interacting with the users of the system and find out what they wanted the system to do. This help came out with the user requirements, operational that is functional and non functional requirements and system requirements**

### **4.2 CONTEXT LEVEL DIAGRAM**

**This showed interaction between system and other external factors with which the system is designed to.** S**ystem context diagram can be useful in understanding how the system worked, they are the processes used in a project to the agreement of the scope a context diagram show the entire system and how it is supposed to work as shown in figure 4.0**

### **figure 4.1: context level diagram**

### **4.3 INPUT DESIGN**

**Admin log in**

**The systems admin will be required to access the system using an username and confidential password as shown in figure 4.3.**

USERNAME

PASSWORD

### 

LOG IN

### **Figure 4.2 Admin log in**

**STUDENT LOG IN**

LOG IN

USERNAME

PASWWORD

All student registered in the institution will be required to use the system by logging in

### **Figure 4.3: Student login**

### **4.4 PROCESS DESIGN**

**A use case diagram to show activities the actors and relationships as shown below.**

**ADMIN**

**student lecturer**  Figure 4.4process design

### **4.5 DATABASE DESIGN**

**The blue prints for storing data in a system are provided by database design. Furnishes access with the data it needs to link the data in the tables as necessary, it ensures the accuracy and the integrity of your information while supporting it.**

**Admin Table**

**The tables show how the administrators’ data is stored in the database.**

### **Table 4.1 Admin table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** |  | **description** |
| **ID** | **Int (5)** | **NOT NULL** | **Primary key** |
| **User name** | **Varchar (45)** | **NOT NULL** | **Name of admin** |
| **password** | **Varchar (45)** | **NOT NULL** | **Password of admin** |

**Student Table**

**The table below illustrates the design of student’s data as stored in the database.**

### **Table 4.2 student details**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** |  | **description** |
| **Reg No** | **Varchar (45)** | **NOT NULL** | **Primary key** |
| **Student name** | **Varchar(45)** | **NOT NULL** | **Full name of student** |
| **password** | **Varchar(45)** | **NOT NULL** | **Password of the student** |

**Lecturer Table**

**The table below illustrates the design of lecturer’s data as stored in the database.**

### **Table 4.3 lecturer table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** |  | **description** |
| **ID** | **Int (5)** | **NOT NULL** | **Primary key** |
| **Email** | **Varchar (45)** | **NOT NULL** | **Email of lecturer** |
| **password** | **Varchar (45)** | **NOT NULL** | **Password of lecturer** |

### **4.6 OUTPUT DESIGN**

**The output design is the process of designing the format, layout and presentation of information that is displayed to users of the system. The goal of output design is to provide users with clear and relevant information that I easy to read and understand. The figure below shows the sketch of an admins page.**

**MAKE ANNOUNCMENT**

**ADD /DELETE**

**HALL**

**ADD /DELETE**

**STUDENT**

**ADD**

**/DELETE**

**SCHEDULE**

### **Figure 4.5: admins dashboard**

### **4.7 CONCLUSION**

This chapter thoroughly explored the design and development of the web-based lecture hall management system. It covered the database structure, explaining tables and how they worked together. It also looked at different user roles like lecturers, students, and administrators, and how they used the system. The chapter mentioned the tools and technologies we used and how they made collaboration or maintenance easier for future developers. It emphasized making the system user-friendly, showing interactions with a use case diagram. Overall, this chapter set a strong foundation for understandinghow the system worked, making it useful for everyone involved.

**CHAPTER 5**: **SYSTEM TESTING AND IMPLEMENTATION**

# ****5.0 INTRODUCTION****

This chapter marked a crucial milestone in developing the web-based lecture hall management system. Here, the developer carefully tested the system's functionality, ensuring it met all design requirements and operated efficiently. Following successful testing, the developer delved into the implementation phase, carefully deploying the system to its designated environment, made it readily accessible for its intended users.

### **5.1 Unit Testing**

The developer commenced the testing process with unit testing. This phase meticulously focused on individual components of the codebase, dissecting models, views, and forms in isolation. Test cases were meticulously crafted to verify each component's behavior under various conditions. These conditions included valid and invalid inputs, along with exploration of edge cases. To streamline this process, the developer leveraged the built-in Django testing framework unittest. Additionally, the project remained open to the adoption of third-party testing frameworks like pytest if needed for more comprehensive testing.

### **5.2 Integration Testing**

Following unit testing, the focus shifted towards integration testing. This phase aimed to assess how different components collaborate and interact with each other. The developer simulated user interactions and examined the flow of data between various components like models, views, and templates. Tools like Selenium proved valuable during this stage, enabling automated browser testing to uncover potential integration issues.

### **5.3 System Testing**

After ensuring the functionality of individual components and their integration, system testing was conducted. This comprehensive evaluation assessed the overall functionality of the system from a user's perspective. The developer actively participated in manual testing of various features, mimicking the actions of lecturers and administrators. This rigorous process unearthed any bugs, usability shortcomings, and ensured the system aligned with the predetermined requirements.

### **5.4 Database Testing**

Data integrity and consistency within the database were paramount. To guarantee this, the developer implemented database testing. This phase involved meticulously testing the application's data manipulation capabilities, encompassing CRUD operations (Create, Read, Update, Delete) performed through Django's ORM (Object-Relational Mapper). The tests verified proper data validation and error handling mechanisms to prevent data corruption.

### **5.5 Implementation Requirements**

For a successful deployment, several implementation requirements were addressed. The developer configured a web server, such as Apache or Nginx, to effectively host the Django application. Finally, the culmination of the development process involved deploying the application to a designated production server or a cloud platform, making it accessible to the intended users.

### **5.6 Coding Tools**

Throughout the development process, the developer employed various tools to streamline coding and collaboration. Git, a version control system, was implemented to meticulously track code changes. This facilitated reverting to previous versions if necessary. Furthermore, the developer actively considered using an Integrated Development Environment (IDE) like Visual Studio Code. These IDEs provide valuable functionalities like code completion and debugging tools, significantly enhancing the development workflow.

### **5.7 System Home Page and Other Relevant Pages**

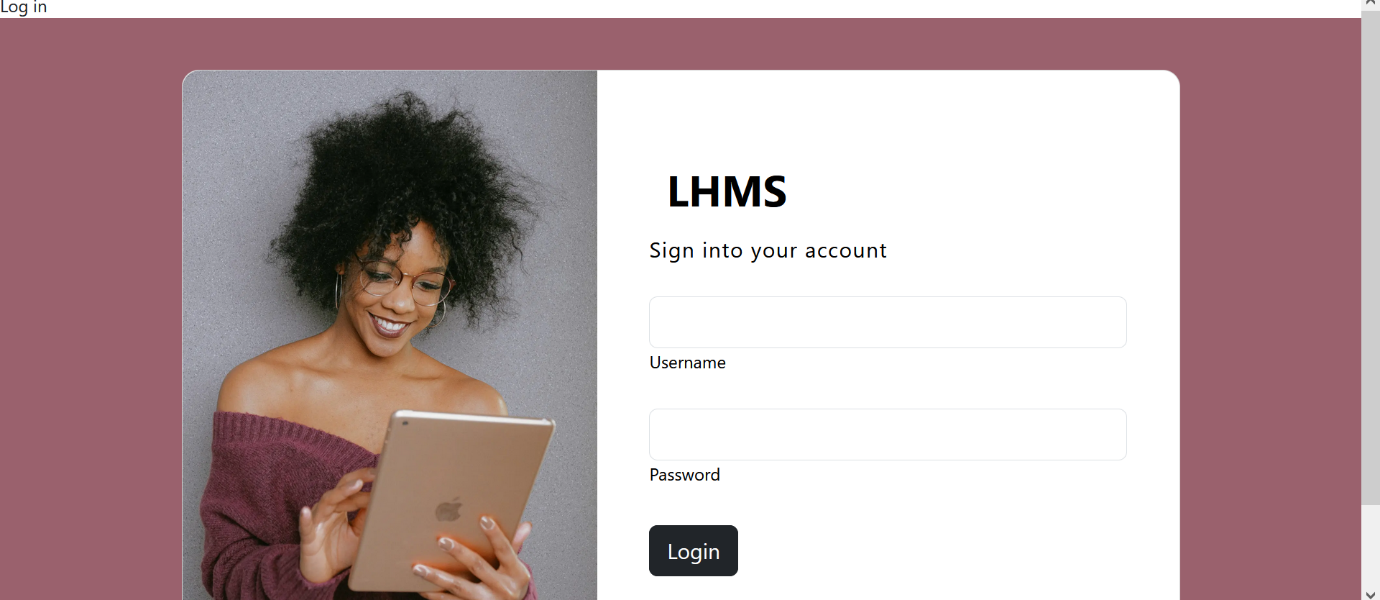
The system's home page was meticulously designed to provide users with a clear overview of the application and its functionalities.

Figure 5.1 below shows the home page of the system.



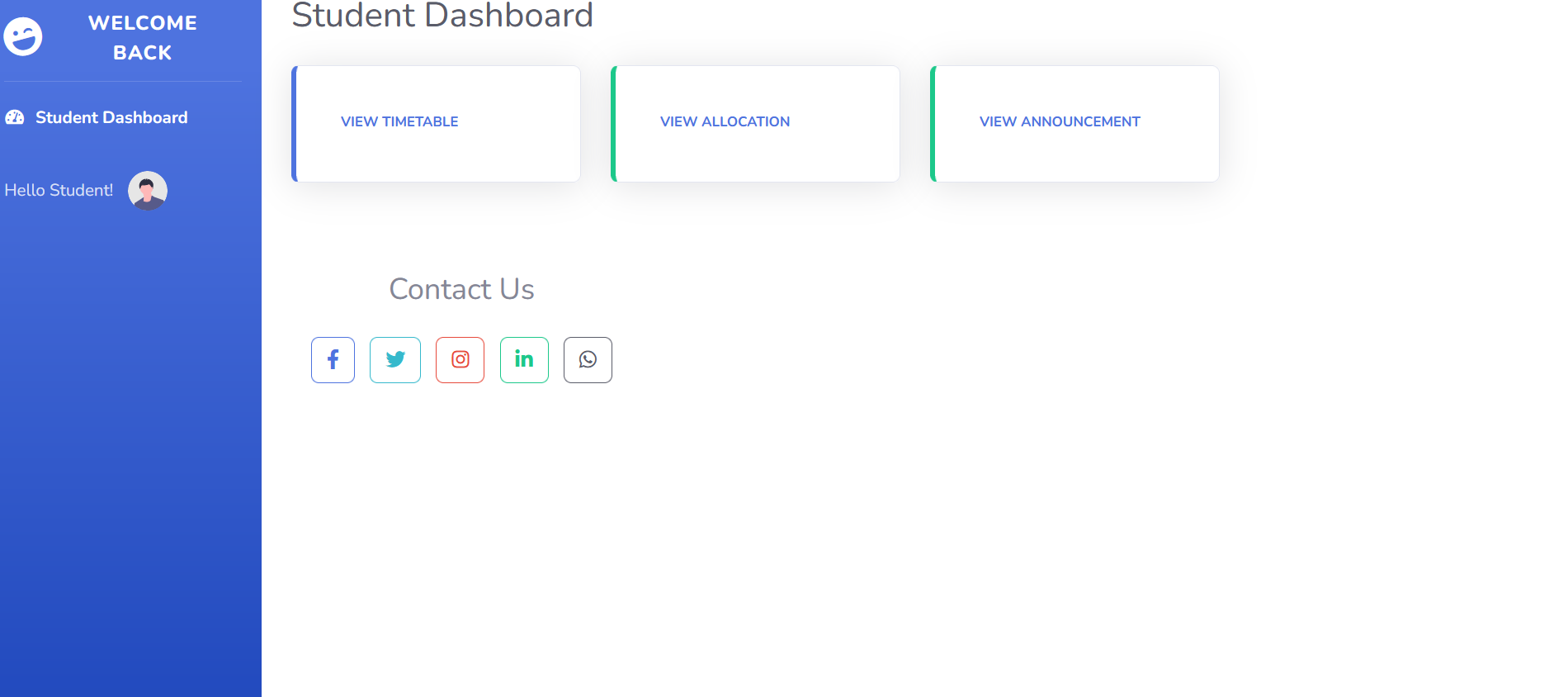
### **Figure 5.1 home page**

The developer prioritized a user-centric approach, ensuring a clear and intuitive layout. Furthermore, other essential pages were developed to deliver the core functionalities of the system. These pages included: Login and Logout functionalities for secure user access.



### **Figure 5.2 Log in page**

A dedicated students dashboard section enabling students to view relevant information .



**Figure 5.3 Students dashboard**

Responsive design principles were meticulously considered throughout the development process. This ensured that the web application functioned seamlessly across various devices, encompassing desktops, tablets, and mobile phones.

### **5.8 Conclusion**

By adhering to a comprehensive testing approach and meticulously following proper implementation procedures, the developer ensured that the lecture hall management system was robust, functional, and fully prepared for deployment. This rigorous testing process instilled confidence that the system would effectively meet the needs of its users.

# CHAPTER 6: CONCLUSION AND RECOMMENDATIONS

### **6.0 Introduction**

This chapter concluded the documentation of the web-based lecture hall management system developed using Django and Bootstrap. Here, the developer recapped the project's achievements, explored potential areas for improvement, and outlined possibilities for future development.

### **6.1 Conclusions**

The development process successfully culminated in a functional and user-friendly web-based lecture hall management system. The developer strategically leveraged Django's robust framework to construct the backend, and Bootstrap's responsive design capabilities to create the frontend. This combination ensured a seamless user experience for lecturers, regardless of the device they were using (desktops, tablets, mobiles).

### **6.1 Recommendation**

Introducing the new` Lecture Hall Management System! It was made to make life easier for schools and colleges. Our system helped you organize your lecture halls better. It showed you which halls were free and how many people they could fit. It was easy for teachers and staff to use. Plus, it gave you updates in real-time, helpful reports, and let you talk to everyone easily. With our system,one could make better choices, use your lecture halls well, and make learning awesome for students and teachers. It was a smoother, more successful way of managing lectures!

### 

### **6.3 Future Work**

The developer suggested several enhancements for future iterations of the system. First, the developer recommended implementing enhanced security features to further strengthen the system's security measures. This included adding user authorization levels and implementing two-factor authentication to provide an extra layer of protection for user accounts. Second, the developer proposed integrating advanced reporting and analytics functionalities into the system. These features would have provided valuable insights for administrators and lecturers, such as generating reports on course enrollment, student attendance trends, and quiz results, enabling data-driven decision-making.

### **6.4 References**

Agile Alliance. (n.d.). Agile methodology. Retrieved from <https://www.agilealliance.org/>

Beck, K. (2002). Agile software development: Principles, patterns, and practices. Addison-Wesley Professional.

Bootstrap Themes. (n.d.). Retrieved from <https://getbootstrap.com/docs/5.2/getting-started/introduction/>

Cohn, M. (2009). Agile estimating and planning. Addison-Wesley Professional.

Driscoll, J. (2023). Django for beginners: Dive into web development with Python. Packt Publishing Ltd.

The Hitchhiker's GuidetoPython.(n.d.).Retrieved from <https://docs.djangoproject.com/en/4.2/howto>

Mazevo Room Booking System. Retrieved from <https://www.gomazevo.com/>

Wu, J., Ding, S., & Fan, J. (2019). Design and implementation of a course resource management system based on Django framework. Journal of Physics: Conference Series, 1187(1), 012052.

National Institute of Standards and Technology (NIST). Cybersecurity Framework. Retrieved from <https://www.nist.gov/cyberframework>

OWASP Top 10 Web Application Security Risks. Retrieved from <https://owasp.org/Top10/>

Stanford University. (n.d.). Event Management System (EMS). [Description of system]. Retrieved from [URL removed]

University of California, Berkeley. (n.d.). Space Management - Room Scheduling & Reservations. [Description of system]. Retrieved from [URL removed]

University of Otago. (n.d.). Lecture room booking and timetabling system. Retrieved from <https://www.otago.ac.nz/study/study-timetables>

### **6.5** Appendices

**Appendix 1: System Home page**



**Figure 6.1: System Home page**

**Appendix 2 : Sample codes**

**Home page**

<!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>LHMS</title>

    {% load static %}

    <link rel="icon" type="image/x-icon" href="{% static 'assets/favicon.ico' %}">

    <!-- Font Awesome icons (free version)-->

    <script src="https://use.fontawesome.com/releases/v6.3.0/js/all.js" crossorigin="anonymous"></script>

    <!-- Google fonts-->

    <link href="https://fonts.googleapis.com/css?family=Catamaran:100,200,300,400,500,600,700,800,900" rel="stylesheet">

    <link href="https://fonts.googleapis.com/css?family=Lato:100,100i,300,300i,400,400i,700,700i,900,900i" rel="stylesheet">

    <!-- Core theme CSS (includes Bootstrap)-->

    <link href="{% static 'css/styles.css' %}" rel="stylesheet">

</head>

<body id="page-top">

    <!-- Navigation-->

    <nav class="navbar navbar-expand-lg navbar-dark navbar-custom fixed-top">

        <div class="container px-5">

            <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarResponsive" aria-controls="navbarResponsive" aria-expanded="false" aria-label="Toggle navigation"><span class="navbar-toggler-icon"></span></button>

            <div class="collapse navbar-collapse" id="navbarResponsive">

                <ul class="navbar-nav ms-auto">

                    <li class="nav-item"><a class="nav-link" href="{% url 'login2' %}">Student</a></li>

                    <li class="nav-item"><a class="nav-link" href="{% url 'login3' %}">Lecturer</a></li>

                </ul>

            </div>

        </div>

    </nav>

    <!-- Header-->

    <header class="masthead text-center text-white">

        <div class="masthead-content">

            <div class="container px-5">

                <h2 class="masthead-heading mb-0">WELCOME TO</h2>

                <h3 class="masthead-subheading mb-0">UNIVERSITY LECTURE HALL MANAGMENT SYSTEM</h3>

            </div>

        </div>

        <div class="bg-circle-1 bg-circle"></div>

        <div class="bg-circle-2 bg-circle"></div>

        <div class="bg-circle-3 bg-circle"></div>

        <div class="bg-circle-4 bg-circle"></div>

    </header>

    <footer class="py-5 bg-black">

        <div class="container px-5">

            <p class="m-0 text-center text-white small">Copyright &copy; Rony Designs</p>

        </div>

    </footer>

    <!-- Bootstrap core JS-->

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js"></script>

    <!-- Core theme JS-->

    <script src="{% static 'js/scripts.js' %}"></script>

</body>

</html>

**Student log in page**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <tit>Log in</title>

    {% load static %}

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN" crossorigin="anonymous">

</head>

<body>

    <section class="vh-100" style="background-color: #9A616D;">

        <div class="container py-5 h-100">

          <div class="row d-flex justify-content-center align-items-center h-100">

            <div class="col col-xl-10">

              <div class="card" style="border-radius: 1rem;">

                <div class="row g-0">

                  <div class="col-md-6 col-lg-5 d-none d-md-block">

                    <img src="https://mdbcdn.b-cdn.net/img/Photos/new-templates/bootstrap-login-form/img1.webp"

                      alt="login form" class="img-fluid" style="border-radius: 1rem 0 0 1rem;" />

                  </div>

                  <div class="col-md-6 col-lg-7 d-flex align-items-center">

                    <div class="card-body p-4 p-lg-5 text-black">

                      <form method="post">

                        {% csrf\_token %}

                        <!-- {{ form.as\_p }} -->

                        <div class="d-flex align-items-center mb-3 pb-1">

                          <i class="fas fa-cubes fa-2x me-3" style="color: #ff6219;"></i>

                          <span class="h1 fw-bold mb-0">LHMS</span>

                        </div>

                        <h5 class="fw-normal mb-3 pb-3" style="letter-spacing: 1px;">Sign into your account</h5>

                        <div >

                          <!-- <input type="text" id="form2Example17" name="username" class="form-control form-control-lg" /> -->

                          <label  for="">{{ form.Username.label }}</label>{{ form.username}}

                        </div>

                        <div >

                          <!-- <input type="password" id="form2Example27" name="password" class="form-control form-control-lg" /> -->

                          <label  for="">{{ form.Password.label }}</label> {{ form.Password}}

                        </div>

                        <div class="pt-1 mb-4">

                          {}

                          <a href="{% url 'student' %}" class="btn btn-primary btn-user btn-block">

                          Login

                          </a>

                        </div>

                      </form>

                    </div>

                  </div>

                </div>

              </div>

            </div>

          </div>

        </div>

      </section>

      <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL" crossorigin="anonymous"></script>

</body>

</html>

**Lecturer login page**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="utf-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <meta name="description" content="">

    <meta name="author" content="">

    <title>Login</title>

    {% load static %}

    <!-- Custom fonts for this template-->

    <link href="{% static 'vendor/fontawesome-free/css/all.min.css' %}" rel="stylesheet" type="text/css">

    <link href="https://fonts.googleapis.com/css?family=Nunito:200,200i,300,300i,400,400i,600,600i,700,700i,800,800i,900,900i"

        rel="stylesheet">

    <!-- Custom styles for this template-->

    <link href="{% static 'css/sb-admin-2.min.css' %}" rel="stylesheet">

</head>

<body class="bg-gradient-primary">

    <div class="container">

        <!-- Outer Row -->

        <div class="row justify-content-center">

            <div class="col-xl-10 col-lg-12 col-md-9">

                <div class="card o-hidden border-0 shadow-lg my-5">

                    <div class="card-body p-0">

                        <!-- Nested Row within Card Body -->

                        <div class="row">

                            <div class="col-lg-6 d-none d-lg-block" style="background-image: url('https://mdbcdn.b-cdn.net/img/Photos/new-templates/bootstrap-login-form/img1.webp'); background-size: cover; background-position: center; border-radius: 1rem 0 0 1rem;">

                                <img src="https://mdbcdn.b-cdn.net/img/Photos/new-templates/bootstrap-login-form/img1.webp" alt="login form" class="img-fluid" style="opacity: 0; height: 100%;" />

                            </div>

                            <div class="col-lg-6">

                                <div class="p-5">

                                    <div class="text-center">

                                        <form action="">

                                            {% csrf\_token %}

                                            {{ form.as\_p }}

                                        <h1 class="h4 text-gray-900 mb-4">LHMS LOG IN</h1>

                                    </div>

                                    <form class="post>

                                        <div class="form-group">

                                            <input type="text" id="form2Example17" name="username" class="form-control form-control-lg"

                                                id="exampleInputEmail" aria-describedby="emailHelp"

                                                placeholder="username...">

                                        </div>

                                        <div class="form-group"