DATA-DRIVEN AND AI-POWERED LECTURE HALL MANAGEMENT SYSTEM FOR OPTIMAL SCHEDULING, RESOURCE ALLOCATION, AND FUTURE PREDICTION IN UNIVERSITY OPERATIONS

A dissertation

presented to the Faculty of Computing

NSBM Green University

in partial fulfillment of the requirements for the degree of

BACHELOR OF SCIENCE IN MANAGEMENT INFORMATION SYSTEMS (SPECIAL)

by

Tharusha D. Egodage

September 2025

Declaration

I, the undersigned Egodagaha Gedara Tharusha Dilhara Egodage hereby declare that I am the sole author of this dissertation. To the best of my knowledge this dissertation contains no material previously published by any other person except where due acknowledgement has been made. This thesis contains no material which has been accepted as part of the requirements of any other academic degree or non-degree program, in English or in any other language. This is a true copy of the dissertation, including final revisions.

Date: 11/09/2025

Name (printed letters): EGODAGAHA GEDARA THARUSHA DILHARA EGODAGE

Signature: ………………………

This dissertation by Egodagaha Gedara Tharusha Dilhara Egodage has been approved by the Supervisor below, who recommended it be accepted by the faculty of Business in partial fulfillment of requirements for the degree of BSc in management information systems (special)

Supervisor's Details

Mr. Gayan Perera	…………………..	10/09/2025
First and Last name	Signature	Date

Abstract

DATA-DRIVEN AND AI-POWERED LECTURE HALL MANAGEMENT SYSTEM FOR OPTIMAL SCHEDULING, RESOURCE ALLOCATION, AND FUTURE PREDICTION IN UNIVERSITY OPERATIONS

Egodagaha Gedara Tharusha Dilhara Egodage

Lecture Hall Management System is an AI-based tool that would solve the problem of inefficiency in scheduling lecture halls in universities. This system can reduce the risk of scheduling conflicts, maximize the use of resources, and make sure that the academic resources are utilized efficiently by combining the **genetic algorithms** to optimize the allocation of halls and the **predictive analytics** to know the demand in advance. The system will also use automation and streamlining of the scheduling process, which will be flexible to dynamic academic requirements.

The study follows a quantitative approach, which uses historical data of the scheduling of NSBM Green University. The data is provided by the analysis of the lecture schedules, patterns of use of halls, and lecturer preferences, and processed with machine learning models to predict the future need of the lecture halls. The predictive model is based on the time-series analysis that utilizes the past data to be able to predict the peak times of demand to allocate the halls optimally. At the same time, genetic algorithms optimize the schedule, so that lecture halls are distributed without any clashes.

These results indicate that predictive analytics is a highly effective tool to increase the accuracy of lecture hall allocation, and the genetic algorithm is an efficient tool to create conflict-free schedules to fill halls. The automated feature of the system in decision-making aids in the effective execution of academic schedules which is advantageous to both the administrative personnel and lecturers.

The study contributes a lot to the sphere of AI-based scheduling systems and offers a scalable and data-driven solution to lecture hall management. It provides a basis on which future research can be conducted in the optimization of academic resource allocation in institutions of learning by emphasizing the need to make decisions based on data.

Keywords: Genetic Algorithms, Predictive Analytics, AI Scheduling, Lecture Hall Management, Optimization, Resource Allocation, Academic Scheduling, Machine Learning

Table of Contents

List Of Tables

List Of Figures

CHAPTER 1: INTRODUCTION

This chapter presents the background, objectives, research problems and motivation for creating an AI-based and data-driven lecture hall management system at NSBM Green University. The project considers significant weaknesses in current scheduling systems (UMIS) and further considers how to overcome them and create **Data-driven and AI-powered lecture hall management system.**

1.1 Chapter Overview

Chapter overview delineates the research's background and context, with an emphasis on the challenges involved in both scheduling lecture halls and the allocation of university resources. The problem with these current systems is articulated, particularly in the context of the limitations of the existing System (UMIS) utilized by NSBM Green University. The chapter provides the groundwork for the investigation by recognizing the research gap, as well as the motivation to pursue the development of a smarter, predictive, and more adaptive solution.

This research proposes a Data-driven and AI-powered lecture hall management system that combines **intelligent agents**, **genetic algorithms**, and **predictive analytics**. To provide a more effective and strategic approach to lecture hall management within the Faculty of Computing at NSBM Green University, the objective is to increase scheduling accuracy, optimize resource allocation, and enable real-time adaptation.

The aim, research questions and objectives are clearly outlined to provide guidance for the outcome of this study. The chapter also provides an overview of the proposed system architecture, as well as the resources and scope required for the undertaking. This chapter produces the groundwork for further chapters of the study that will engage with literature, design approach, system implementation, and evaluation.

1.2 Problem Background

University administrators face escalating demands to maximize their physical resource utilization, especially regarding lecture halls because of growing student populations combined with intricate timing needs of academic activities. Organizations must arrange their space correctly between lecture rooms alongside tutorial areas and test rooms and institutional activities including hackathons and symposiums to achieve academic stability. Most educational institutions maintain scheduling systems which consist of manual or semi-automated components despite their inability to adjust due to changing requirements. At NSBM Green University, the University Management Information System (UMIS) serves as the central platform for lecture hall allocation. The administrative functions of UMIS are supported but it does not include predictive planning or real-time functionality nor decision-making assistance capabilities. Due to its fixed operating mechanism and dependence on outdated assumptions the system creates multiple scheduling issues and wastes resources and results in elevated administrative workload.

The following (*Figure 1)* summarizes the critical issues affecting effective lecture hall scheduling in modern academic environments, specifically focusing on the challenges observed at NSBM.

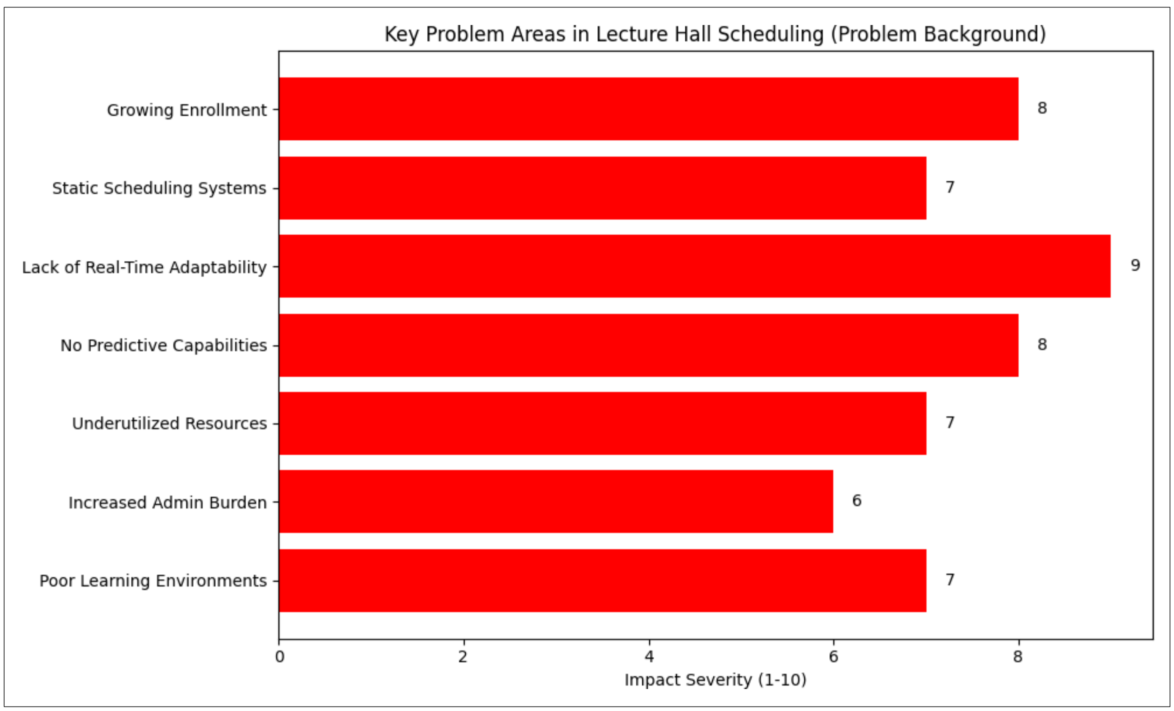


Figure 1: Key Problem Areas in Lecture Hall Scheduling (based on impact severity)

These shortcomings become particularly clear at busy times, such as the exam weeks, the peaks in enrollment, or where enrollment increases suddenly due to a curriculum change. In this situation, the existing system (and the knowledge of the people using the system) is unable to match resource use with demand. For example, lecture halls are sometimes over-reserved or available space is not completely utilized, which detracts from the learning experience and creates inefficient workflows.

Advancements in technology such as artificial intelligence (AI), predictive analytics, and intelligent agent systems present compelling alternatives to traditional scheduling approaches. These advances have disrupted industries ranging from transportation through healthcare to finance by allowing for smarter resource deployment and demand forecasting. But in the education sphere, their uptake in the management of resources is and remains relatively new and can offer much in the perspective of enhancement.[1][2]

There is a chance to completely transform how universities handle space by incorporating these technologies into lecture hall scheduling systems. Universities may drastically lower conflict, increase resource efficiency, and raise student happiness by utilizing past data, projecting future requirements, and facilitating adaptive decision-making.

1.2.1 Creating the Platform for Discussion /Argument

- Historically, the scheduling of lecture halls has been regarded as a logistical task rather than a strategic function in university operation. However, the recent evolution of educational administration and moving from an analog to a digital existence has helped moved this thinking. We've come to expect institutions to work more effectively, decreasing operational costs, and create modes of learning that are increasingly flexible. The COVID-19 pandemic laid bare the constraints of our legacy systems in the face of unprecedented change, creating a new environment for having systems that can facilitate managing and responding to change quickly.[3]

- Smart campus solution demand continues to surge at an accelerated rate throughout the international marketplace. MarketsandMarkets (2023) predicted that the smart campus market will grow to USD 25.3 billion by 2027 because it started at USD 8.6 billion in 2022 which means it will expand by 24.4% annually. AI alongside data analytics and IoT systems drive the expansion of the higher education market through operational efficiencies and student-centric enhancements.[4]

- In Sri Lanka, universities are going through the digital transition, however, a lot of systems, such as UMIS, still work in silos and do not have the intelligence to leverage predictive scheduling functionalities. Case studies from universities in Europe and Asia have provided evidence that AI-powered scheduling tools have alleviated scheduling conflicts and improved utilization of the hall by as much as 40%.[5][6]

  A hybrid lecture hall management system serves as a response to establish an effective solution. The proposed system uses global case study insights to develop a scalable solution which enforces current digital transformation standards and applies efficiently at NSBM Green University as well as other Sri Lankan educational institutions.[7]

1.3 Problem Statement

The modern educational framework presents universities with daunting challenges to properly handle their physical facilities for learning. The growing enrollments and increasingly complex program structures and timetable arrangements of institutions make traditional scheduling systems insufficient to handle current needs. Evaluating lecture hall management maintains both its logistical function and its direct influence on teaching effectiveness as well as learning outcomes and resource distribution. From a broad viewpoint and a field-specific angle the section presents an analysis of the issue.

1.3.1 General Problem (High-Level Implications)

Across the higher education sector, inefficient lecture hall scheduling and resource allocation lead to a ripple effect of negative consequences. From the administrative side, scheduling is done poorly, causing delays, overbooking and necessitating frequent manual adjustments which constitutes time and human resources. On the academic side, the classrooms are always overcrowded, the venues are changed only a few minutes before commencement, or the learning environment is wholly unfit to be in all of which results in a poor quality of education for the students.

However, if that issue is dealt with via the fragmented, ad-hoc methods or the static systems, universities may find themselves:

- Increased operational costs due to reactive adjustments and poor resource utilization.
- Student satisfaction score, as well as student learning engagement.
- Negative impacts on faculty productivity and morale.
- Wasted investments in infrastructure, such as wasted investments in underused or mismatched lecture halls.

1.3.2 Specific Problem (Domain-Specific Shortcomings & Research Gap)

The current scheduling systems suffer from a lack of data-driven intelligent technologies according to the viewpoint of computing and information systems. The academic scheduling system at NSBM UMIS functions by following programmed rules with manual oversight although it does not possess required adaptive features for managing intricate academic operations. The systems currently in use at NSBM UMIS lack integration with three key components:

- **Predictive Analytics**: To anticipate future demand based on historical trends (e.g., recurring exam periods or fluctuating class sizes).
- **Artificial Intelligence (AI) Techniques**: Such as Genetic Algorithms for optimizing scheduling under multi-variable constraints.
- **Intelligent Agents**: That can make real-time decisions and adapt to last-minute changes in class or room availability.

1.4 Research Question

- **How can a data-based and AI-powered lecture hall management system improve responsiveness, efficiency, and accuracy of resource allocation in a changing university environment?**

- How can historical data and predictive analytics be applied to properly predict demand for lecture halls, especially during periods of peak demand such as semester exams or heavy enrollment for degree programs?

- What is some decision-support features that can be integrated into the system to assist university administrators in making real-time, data-informed scheduling decisions?

- How can genetic algorithms and intelligent agents improve the scheduling system's responsiveness and adaptability to variability in hall demand or availability?

- What are the prevalent quantitative (e.g., location, capacity) and qualitative (e.g., lighting, Wi-Fi, equipment) factors affecting booking of lecture halls, and how can these be best incorporated within scheduling decision-making?

- How is the proposed predictive and adaptive system more effective in scheduling and less so in resource conflicts compared to the existing UMIS system?

- How far does the proposed system handle real-time fluctuations in demand for lecture theaters, and what are its measurable gains in terms of resource utilization and user satisfaction?

1.5 Research Motivation

The research responds directly to increasing complexity alongside inefficiency in university lecture hall scheduling which can be observed in NSBM Green University. The current scheduling systems showed discrepancies with actual student and academic staff requirements because I witnessed firsthand poorly scheduled lectures and overcrowding and empty lecture halls. The unresolved issues would affect both educational quality and operational effectiveness in a negative way.

The recent development of Artificial Intelligence-based solutions including predictive analytics and genetic algorithms and intelligent agents delivers organizations with a prompt promising method to handle their educational resource management. Academic scheduling benefits little from these successful technologies since they primarily operate in different domains like logistics and healthcare and smart cities.

The research investigates both a critical local demand and advances knowledge in the domain of intelligent systems used in education The proposed hybrid data model specifically designed for NSBM seeks to create an adaptable framework that improves decision processes while forecasting demand patterns to establish intelligent learning spaces in Sri Lanka and international educational contexts.

1.6 Research Aim

The goal of this research is to design and deploy a data-driven and AI-enabled lecture hall management system that enhances the accuracy, efficacy, and responsiveness of resource allocation within the university setting. Using the integration of predictive analytics, genetic algorithms, and intelligent agents, the system should be capable of optimizing the scheduling of lecture halls, respond to real-time demand changes, and provide decision-support capability for university administrators. The goal is to overcome the limitations of existing static systems such as UMIS and help make NSBM Green University more intelligent and forward-looking in its handling of academic infrastructure.

1.7 Research Objectives

- To identify the most significant quantitative and qualitative factors such as hall capacity, location, equipment availability, and hall conditions that significantly affect lecture hall scheduling in the Faculty of Computing at NSBM Green University.
- To analyze and compare the shortcomings and limitations of the current lecture hall scheduling system (UMIS), with particular reference to its inability to predict, flexibility, and integration of intelligent decision-making features.
- To design and develop an intelligent lecture hall management system with predictive analytics, genetic algorithms, and intelligent agents for efficient scheduling and real-time decision support.
- To evaluate the performance, accuracy, and response of the proposed system against the existing UMIS platform in the context of scheduling efficiency, usage of resources, and user satisfaction.

1.8 Rich Picture of the Proposed Solution

- **Explanation of the Rich Picture Diagram**

1. **Introduction:**

At the beginning of your research, describe what the Rich Picture is for. As per the above diagram, the system of lecture hall management at NSBM Green University can be described visually as representing the stakeholders, activities, and technological resources involved in scheduling and resource allocation of the lecture halls.

2. **Stakeholders:**
    - **University Administrators:** They are responsible for the scheduling system and for optimal hall utilization. In the diagram, they are in the center of decision making since they act with the system to see recommendations and accept scheduling decisions.
    - **Lecturers:** They are the primary users of the system, and they enter the class timings and hall preferences for the allocation of the class. They directly influence scheduling conflicts or inefficiencies.
    - **Students:** They are in touch with the system to check out their class schedules and hall assignments. System usability and hall availability is suggested to enhance system usability based on the user feedback.
    - **University IT Staff:** They manage the system, ensuring it runs smoothly, solving technical problems and incorporating updates.
    - **Event Coordinators:** They ask for lecture halls for purposes such as exams, workshops or conferences. In the scheduling process we consider their needs.
3. **System Components:**
    - **Lecture Hall Database:** Contains all the information regarding available halls, such as their capacity, location and available resources (Wi-Fi, projectors, etc.). This is the basis for planning decisions.
    - **Predictive Analytics Engine:** Uses historical data to forecast peak demand periods, such as exam seasons or times of high student enrollment. This engine helps plan future hall requirements and optimize scheduling.
    - **Genetic Algorithm Module:** A key component in optimizing the hall assignments given various constraints such as class sizes, equipment requirements and lecturer availability. It imitates the process of natural selection to identify the optimal possible schedule.
    - **Intelligent Agent Module:** Dynamic agents that can change the schedule as required, in real time, in response to unforeseen changes such as last-minute cancellations or rebooking requests.
    - **Decision Support System:** Offers administrators data-driven recommendations and insights on hall availability, optimal scheduling, and resource allocation. It assures that the administrators make informed decisions.

4. **Processes:**
    - **Data Collection:** The system collects data from multiple sources, such as surveys, historical scheduling information and real-time usage data. Demand patterns and user preferences are also important to understand from this information.
    - **Scheduling Optimization:** The central process in which the system uses Artificial Intelligence tools (Predictive Analytics and Genetic Algorithms) to generate the most effective hall schedules and reduce conflicts.
    - **Real-Time Adaptation:** The system continuously monitors usage patterns and makes real-time adjustments, such as reallocating halls or rescheduling classes to adapt to unexpected changes.
    - **User Feedback:** Throughout the process, feedback is obtained from users such as students, lecturers, and administrators, which helps in making constant improvements and changes to the system to enhance the satisfaction of the users.
5. **External Factors:**
    - **University Events:** Hall availability and demand are impacted by events such as exams, special academic conferences, etc. The system should provide for these by rescheduling and reserving rooms.
    - **System Constraints:** Physical constraints (like room size and capacity) and technological limitations (such as Wi-Fi availability or system processing power) play a significant role in how the scheduling system operates.
6. **Information Flow:**
    - **Feedback Loops:** User feedback (students and lecturers) is fed back into the system on a continual basis to adapt and enhance scheduling performance.
    - **Data Flow:** Data is passed through the system, from lecture hall database to predictive and optimization engines, to eventual decisions by administrators and event coordinators.

State that the Rich Picture diagram is a way to see the complexity of the lecture hall management system and how the system components fit together. By explaining it, you can showcase the system's strengths, its predictive abilities and its adaptability to real-time changes. The diagram also shows how AI technologies (such as predictive analytics and intelligent agents) are embedded into your research.

1.9 Resource Requirements

Developing the AI-powered lecture hall management system requires essential hardware, software tools, and cloud services to ensure scalability, real-time responsiveness, and efficient deployment.

1.9.1 Hardware

Hardware Component	Purpose
Standard Laptop or Desktop (8GB+ RAM, i5/i7 or Ryzen 5/7)	For development, algorithm testing, and local server setup.
GPU (NVIDIA CUDA-enabled)	For training or testing predictive models if deep learning is integrated.
External Storage or Cloud Backup	Backup datasets and system files.
Internet Connectivity	Required for real-time system access, cloud integration, and user collaboration.

Table 1: Hardware Component

1.9.2 Software

Software / Tool	Purpose
Python	Python is primary used for system development and data analysis.
Flask / Django	The web framework is used for building the backend and serving the frontend UI.
Pandas, NumPy	Data manipulation, calculation and transformation tools.
Scikit-learn / XGBoost	For building and testing predictive models.
Matplotlib / Seaborn	Visualization tools for charts, graphs, and reporting.
SQLite / PostgreSQL	Database solutions to manage structured data (lecture halls, schedules, equipment, etc).
Draw.io	Diagramming tools for designing system architecture and workflows.
Git / GitHub	Version control for source code management and collaboration.
Jupyter Notebook / VS Code	Environments for exploratory development, ML modeling, and documentation.
Google Forms / Microsoft Forms	For conducting surveys and collecting stakeholder feedback.
Excel / CSV	Import/export of legacy data and analysis datasets.

Table 2: Software Component

1.9.3 Cloud Services

Cloud Tool / Service	Purpose
Google Colab / JupyterHub	Cloud-based development and testing of machine learning models.
Firebase / Firestore	Real-time database and authentication for system integration (student logins, Staff logins).
Amazon Web Services (AWS)	Scalable backend services (e.g., EC2 for hosting, S3 for data storage).
Google Drive	Backup and sharing of project files, datasets, and reports.
Heroku / Render / Railway	Simple deployment of web applications (Flask/Django) for demo/testing.

Table 3: Cloud Services

1.10 Project Scope

The scope of the study establishes the functional boundaries, technological focus, and expected outcomes of the system, in addition to what will not be included in the project. The areas that are in scope and out of scope for this study are explicitly stated in the table below.

In-Scope	Out-of-Scope
Designing and developing a prototype system for lecture hall scheduling.	Commercial-level deployment across the entire NSBM university.
Implementing predictive analytics using historical scheduling and hall usage data.	Handling unrelated university resources like staff scheduling or inventory management.
Applying Genetic Algorithms for timetable optimization.	Advanced deep learning models requiring high-performance GPU clusters.
Incorporating Intelligent Agents for adaptive decision-making.	Full mobile application development or offline desktop applications.
Collecting data via questionnaires, interviews, and historical datasets.	Integration with all external systems (e.g., another university student database or other faculties).
Evaluating the performance of the proposed system using simulated and historical data.	Real-time synchronization with every live timetable change across all faculties.
Providing admin interfaces and basic dashboards for visualization.	Building advanced BI (Business Intelligence) or enterprise-level analytics modules.

Table 4: Project Scope for Data-Driven and AI-Powered Lecture Hall Management System

1.11 Chapter Summary

This chapter introduced the background, motivation, and problem context for developing a data- driven and AI-powered lecture hall management system. It highlighted current limitations in university scheduling systems like UMIS and defined the research problem, aim, and objectives. Key technologies **predictive analytics**, **intelligent agents**, and **genetic algorithms** were outlined as core components of the proposed solution. Overall, the chapter sets the foundation for designing an intelligent, adaptive system tailored to the needs of the Faculty of Computing at NSBM Green University.

CHAPTER 2: LITERATURE REVIEW

This chapter reviews existing research and technologies related to AI-based and data-driven lecture hall scheduling, resource management, future prediction and intelligent decision-making in NSBM Green University. It critically examines data-driven methods such as **Predictive Analytics**, **Genetic Algorithms**, and **Intelligent Agents**, identifying their strengths and limitations in current systems (UMIS). The review serves to establish the foundation for the proposed research by highlighting gaps in adaptability, prediction, and decision support particularly within systems like UMIS that this study aims to address.

2.1 Chapter Overview

This chapter provides a complete review of existing literature in areas related to faculty of computing, lecture hall management, scheduling optimization and intelligent decision support systems in NSBM Green University. It tries to examine the important technologies, methodologies and frameworks employed for handling lecture hall allocation, automatic timetable generation and real time adaptability problems in university settings.

The literature review commenced by analyzing traditional as well as modern scheduling practice, and the way systems such as UMIS are currently used. Then it emphasizes a few advanced techniques like predictive analytics, genetic algorithms, and intelligent agents, their applications and their limitations. Using this analysis, the chapter utilizes this analysis to highlight several important gaps in existing solutions, in particular the severe degree of missing integration among decision support functions and predictive technologies.

Using critical evaluation to these findings, the chapter serves as the basis of the proposed research and the argument for a data-driven and AI-based system that adapts to the dynamic requirements of university resource management.

2.2 Conceptual Map of Literature Review

The literature review consists of the conceptual map, which illustrates the most important themes, technologies and research domains represented by the foundations of this study. It gives the knowledge of connections and contributions to the construction of the data-driven, AI-powered lecture hall management system of current studies, systems and methodologies.

In this section, the main topics of literature interested in are traditional and automated scheduling systems, the use of predictive analytics in predicting resource needs and future prediction, the use of genetic algorithms for timetable optimization and the usage of intelligent agents for real time decision. Moreover, it evaluates the constraints of existing systems such as UMIS and the lack of their ability to adjust to variable dynamics and to predict future scheduling conflicts.

Besides assisting in structuring the reviews in the literature into meaningful categories, the conceptual map also allows us to see how the different areas are relevant to the problem the literature has identified. It establishes the groundwork for the proposed hybrid solution by bridging the gap in existing systems using new technologies that allow for more flexible, accurate and intelligent scheduling capabilities in university environments.

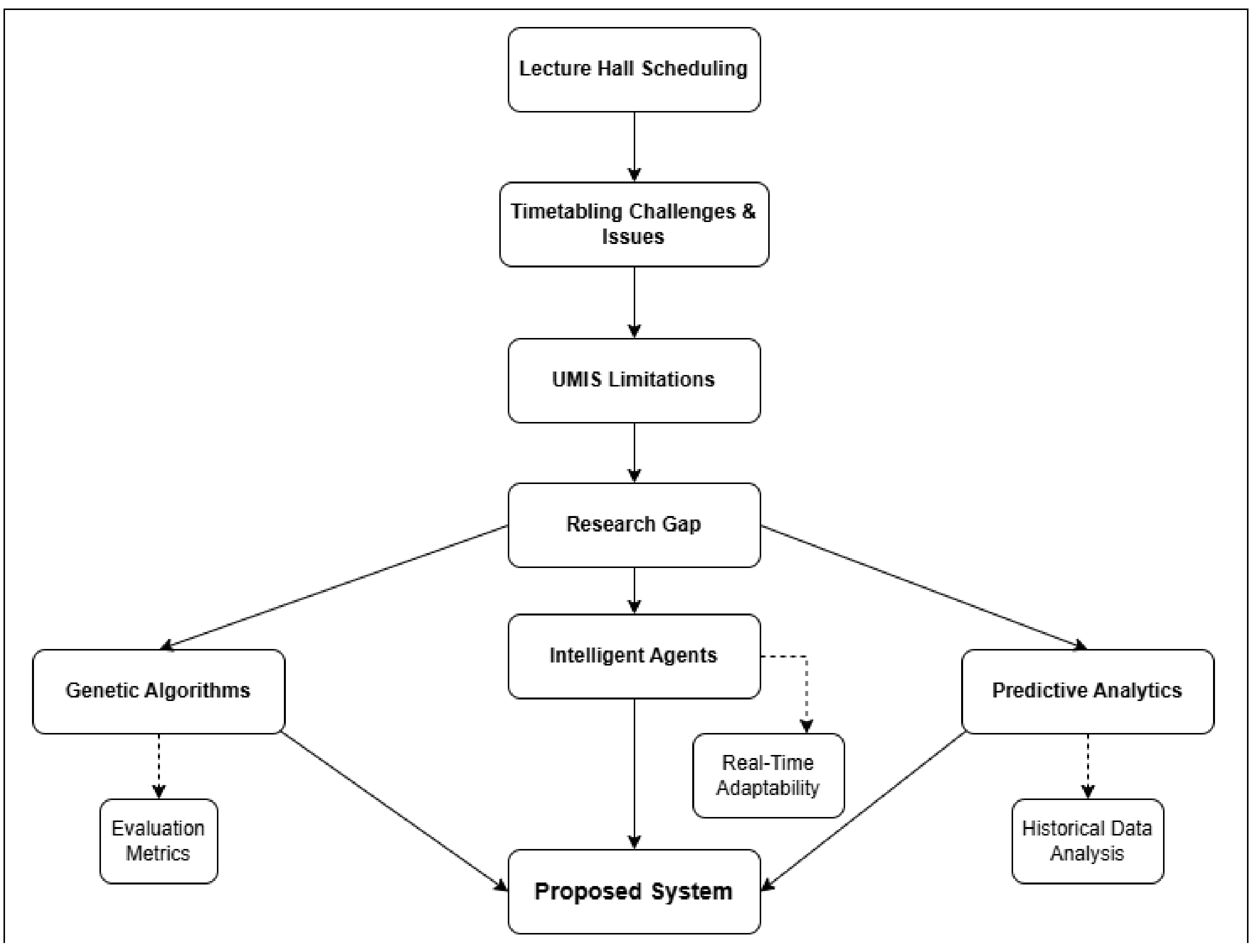


Figure 2: Conceptual Map illustrating key themes, technologies, and their relationship to the proposed system

2.3 Literature Review - Thematic Areas (Key Themes and Technological Foundations)

This section expands on specific literature themes that are directly aligned with this research problem, technological foundation, and proposed solution.

2.3.1 Traditional and Legacy Lecture Hall Scheduling Systems

Traditional university lecture hall scheduling systems usually employ manual or semi-automatic techniques using spreadsheets or minimal web platforms. They operate with fixed constraints and pre-programmed rules, allowing little room for dynamic response. The UMIS platform of NSBM Green University is an example where static scheduling procedures limit responsiveness to real-time fluctuations like changing student numbers or unexpected classroom demands.

It is found through research that initial rule-based and heuristic scheduling models were not scalable or flexible in large-scale academic environments [8]. The legacy systems cannot incorporate AI or predictive functionality, resulting in scheduling conflicts, inefficient use of resources, and an increased administrative burden [9]. Moreover, decision-making within such systems is still largely manual, and very little support exists for automatic planning or optimization [10].

While these systems comprised the basis of academic scheduling, they fall short of addressing modern dynamic requirements. Their limitations provide the motivation for exploring smart, data-driven solutions that can predict demand and shift in real-time.

2.3.2 Predictive Analytics in Educational Planning

Predictive analytics is a significant component of modern educational planning since it allows institutions to predict future trends in lecture hall usage, student enrollment, and course scheduling. By applying historical data, machine learning algorithms, and statistical techniques, universities can anticipate busy periods such as examination seasons or high-enrollment semesters and pre-allocated resources in advance. This allows for more effective utilization of space, removes scheduling conflicts, and reduces administrative workload [8].

Compared to static scheduling systems such as UMIS, predictive models bring flexibility and responsiveness, more suited to the dynamic academic environments of today. The systems allow administrators to make data-driven decisions, enabling the strategic planning of facilities and schedules. Research shows that predictive tools significantly improve the effectiveness of resource management in education, especially when combined with real-time responsiveness and intelligent automation [11]. Hence, predictive analytics is at the core of creating smart, effective, and scalable lecture hall management systems.

2.3.3 Genetic Algorithms and Intelligent Agents use for Lecture Hall Optimization

Integrating Genetic Algorithms (GAs) and Intelligent Agents in university timetabling has shown effective results in maximizing scheduling efficiency, flexibility, and conflict reduction. GAs, based on the principles of natural selection, perform especially well in optimizing schedules based on complex constraints like room availability, lecturer aversion, and student numbers. Because they can generate better solutions in each subsequent generation, they are particularly suitable for solving NP-hard problems like timetabling. Intelligent Agents complement this with real-time flexibility, dynamic decision-making based on changing needs and inputs.

For example, a GA system that was coupled with intelligent agents achieved a 70% reduction in scheduling conflicts and a 45% improvement in timetable generation time [12]. Another achievement was shown in an academic setting where an Evolutionary Algorithm-based scheduler produced optimal results with higher convergence rates and better constraint satisfaction than conventional GA techniques [13].

The synergy of the technologies allows institutions to efficiently allocate resources, handle schedule changes in real-time, and improve user experience for students and staff-something that just works beautifully within the goals of a data-driven and AI-driven lecture hall management system.

2.3.4 Real-World Implementations and Success Stories in AI-Driven Scheduling

The practical success of intelligent systems in academic scheduling highlights their transformative potential. For instance, Dasgupta and Khazanchi's IAEDS (Intelligent Agent Enabled Decision Support) system successfully managed dynamic academic scheduling using adaptive agents that continuously learn from historical and real-time data. This system not only reduced human intervention but also improved responsiveness to changes in demand and constraints [14].

One more fascinating example is the hybrid timetable that was installed using the assistance of Genetic Algorithms, which reduced the timetable building time by 45% and conflicts by 70%. In comparison with common methods, the system demonstrated measurable improvements in terms of speed and scheduling quality [12].

In Nigeria, an entirely computerized lecture scheduling system via Genetic Algorithms registered an extremely high rate of success. Over 84% of the participants agreed that the automatic way was more convenient because of efficiency and negligible clash rates, while 82.5% reported zero clashes-drastic compared to older manual systems [15].

These real-world applications show that genetic algorithms, intelligent agents, and predictive analytics are not only theoretical options but also proved to be effective in reality in optimizing education scheduling processes.

2.4 Research Gap

2.4.1 Limitations of the UMIS System and Its Underlying Technologies

The University Management Information System (UMIS) currently in use at NSBM Green University provides basic administrative functionality, like booking lecture halls. Its technology stack and underlying architecture, however, are specialized and limited in nature. UMIS is primarily PHP web-based with rudimentary relational databases, which serve well enough for static data management but are not adequate for dynamic, real-time scheduling needs.

The system does not have embedded predictive analysis or optimization code and relies on administrative-defined constraints and updates being done manually. This results in rigid scheduling that cannot adapt to evolving student enrollment, room states, or academic event schedules. As described in previous research, this type of system is susceptible to resource inefficiency, increased administrative burden, and scheduling conflicts [16],[9].

Furthermore, the current system does not have the capability to integrate AI-based models such as Genetic Algorithms or Intelligent Agents. Without these, UMIS is always reactive but never proactive. This technological deficit underscores the need for a decision-support system that makes decisions in real-time, forecasts outcomes based on data and adjusts automation to improve efficiency and scalability [14].

2.4.2 Operational Challenges in the Current UMIS Lecture Hall Scheduling System

The UMIS at NSBM Green University includes baseline scheduling functions yet operational problems reduce its productivity. The system lacks the ability to automatically adapt to evolving academic requirements across different academics' periods. The scheduling issues that emerge in the non-real-time adaptive system stem from class rescheduling while rooms become unavailable unexpectedly or when student enrollment changes [14].

Intelligent automation stands as a critical absence in this system. The absence of adequate learning environments through UMIS results in fewer effective lecturing sessions as well as reducing student satisfaction. According to previous research traditional scheduling systems fail to maximize resource usage within complex educational settings [9]. Without AI integration UMIS lacks the ability to provide essential predictive insight which would help forecast high-demand periods and adjust capacity accordingly [12].

UMIS faces limitations when it comes to integrating qualitative resource attributes into its scheduling model such as equipment in rooms and Wi-Fi quality and lighting quality. The lack of integration between UMIS and AI technologies leads to spaces that are less than optimal for teaching and learning which results in worsening student satisfaction and decreased teaching effectiveness. An intelligent, responsive, farsighted solution becomes essential because of these system limitations.

2.4.3 Comparative Analysis: UMIS vs. Advanced Scheduling Systems Worldwide

The NSBM Green University Management Information System (UMIS) is a simple, half-automated system for administrative processes of academia, including scheduling lecture halls. UMIS lacks severely in terms of flexibility, intelligence, and scalability compared to advanced international systems.

Across the globe, academies have adopted sophisticated solutions with Genetic Algorithm (GA) and Intelligent Agent technology to optimize performance and responsiveness in academic timetable systems. The contemporary systems utilize evolutionary computation to generate schedules with the best fit, self-solving hard problems like room under availability, lecturers' conflicting responsibilities, or high-occupancy time slots [12]. In contrast, UMIS is founded on static rule logic and requires careful manual surveillance, which makes it reactive instead of proactive.

In addition, intelligent agent-based systems like the IAEDS model continuously check and update schedules in real-time without any kind of human intervention, a feature that UMIS completely lacks [14]. International systems also employ predictive analytics, allowing them to forecast room demand based on historical trends and adjust resource allocation accordingly [13].

Overall, while UMIS does basic scheduling, it does not possess the intelligence, automation, and forecasting that mark existing, internationally applied systems.

2.4.4 Global Responses & Mitigation Strategies for Academic Scheduling Challenges

To overcome the drawbacks of conventional systems like UMIS, intelligent technologies like predictive analytics, Genetic Algorithms (GAs), and Intelligent Agents have been introduced by universities worldwide. These technologies offer automatic, optimized, and real-time scheduling along with adaptability to dynamic academic environments. For example, GA-based timetable systems with significantly reduced conflict rates and optimizing the usage of resources for large institutions have been implemented in universities [13].

Globally, Intelligent Agent-based systems have been adopted to aid decentralized decision-making. They continuously track input data, for instance, class size changes or last-minute cancellations, and make decisions independently without administrative intervention [14]. These agent-based models have maximized system flexibility and lessened the workload on human schedulers.

To mitigate current deficiencies in UMIS, the progressive introduction of AI-supported systems is suggested. These include incorporating machine learning to forecast demand, introducing intelligent agents for conflict resolution in real time, and introducing qualitative resource parameters in the schedule matrix. Not only do these align local systems with international best practices, but they are also future-proof university systems by making them scalable and adaptive [12].

2.5 Technological Analysis

In this section, we will discuss the technological components that make up the proposed **Data-Driven and AI-Powered Lecture Hall Management System**. The technological analysis is carried out on the applicability and justification of the key components of the implementation: **algorithms, system design,** and **workflow**, in order to ensure that the selected solutions are adequate to solve the problems of the current scheduling system at NSBM Green University.

2.5.1 Algorithmic Analysis

The major algorithms selected for this system are Genetic Algorithms and Predictive Analytics. **Genetic Algorithms** (GA) are very suitable for solving complex optimization problems; therefore, they are suitable for the lecture hall scheduling problem. These algorithms are based on the process of natural selection and are able to optimize schedules over a series of iterations while respecting constraints such as hall size, equipment availability, and class times. The GA method enables efficient search in the space of possible solutions, with the guarantee of a reduction in scheduling conflicts.

Furthermore, Predictive Analytics is invaluable in predicting the future demand for lecture halls (especially during peak examination periods, or when the number of students enrolled varies throughout the year). By using historical data, the system can predict these peak times and allocate resources in advance to prevent scheduling conflicts. These algorithms are well-suited to our research because they can process large amounts of data and make decisions that optimize multiple factors. However, we are aware that Genetic Algorithms can become computationally more complex with data size. To reduce this, we will use optimization techniques that cut down the processing time.

2.5.2 Design Analysis

The system architecture is designed to scale, configure and be easy to use. At the core of the system is the Predictive Analytics Engine, Genetic Algorithm Module and Intelligent Agent System. This modular design enables individual parts to be upgraded or replaced as new technologies become available, providing long-term usability. The system is also equipped with a user-friendly user interface that will allow the administrators to navigate and manage the scheduling tasks with ease.

The design has been aimed at enhancement of the existing UMIS system with real-time data processing and prediction capabilities which were not available in the existing system. It makes the system flexible enough to adjust to changing schedules, such as last-minute cancellations of classes or requests for extra rooms. Further, the versatility of the design allows for the accommodation of future demand and effective operation as the university becomes larger.

2.5.3 Workflow Analysis

The workflow of the proposed system is very well defined and is very efficient. First, some data concerning the availability of lecture halls, timetables and historical usage data will be gathered. The system will then use Predictive Analytics to predict demand at high times. Genetic Algorithms will be used to optimize the schedule based on all the constraints, and the Intelligent Agent System will make changes to the schedule in real time to adapt to changes even though they are unforeseen.

The workflow reduces manual intervention, which is not the case with the existing UMIS process where human intervention is significant. This manual-thinning results in more administrative efficiency as well as fewer scheduling mistakes. In addition, the system includes decision-support functions for administrators, in the form of data-driven suggestions for hall allocation.

2.6 Reflection

The literature discussed in this chapter has given significant inputs on the present status of academic timetable systems, their technological underpinnings, and the potential promised by AI-based systems. Through the study of both traditional systems such as UMIS and intelligent scheduling systems implemented globally, it could be distinctly understood that although base-level development has been achieved, a large gap exists as far as flexibility, instantaneous decision-making, and forecast planning are concerned.

A key takeaway is the contrast between rigid systems and smart, dynamic models driven by Genetic Algorithms and Intelligent Agents. These technologies have been shown to minimize scheduling conflicts, optimize resource utilization, and reduce administrative intervention. However, most applications remain fragmented too often emphasizing either optimization or real-time responsiveness, but not often both in conjunction with predictive analytics and qualitative factors.

This consideration also highlights the importance of filling that gap in a local setting. While international solutions are appreciated, they are not always applicable due to differences in institutional needs and technological readiness. This study, therefore, promotes a tailored, hybrid system that not only includes predictive and adaptive features but also explicitly matches the resource constraints and administrative procedures of NSBM Green University.

Lastly, this reflection informs and augments the rationalization of the proposed system design so that not only is it innovative, but also contextually applicable and realistically viable.

CHAPTER 3: Research Methodology

The chapter provides information on how the research was carried out while creating the proposed lecture hall management system in Faculty of Computing, NSBM Green University. The methods section states the research approach taken the design employed and lists the ways data was collected and analyzed. Since the approach is both academically strong and practical, the resulting system is created to meet actual needs and is based on proper research. This chapter also mentions the instruments, ethical considerations, and the rationale behind the chosen methods.

3.1 Introduction

This chapter details how the study used a particular method to explore and introduce a lecture hall management system powered by data and AI. With more students and programs at NSBM Green University's Faculty of Computing, it is now obvious that we require a smarter and more flexible approach to academic planning. To meet this need, this research develops a system that makes use of predictive analytics, genetic algorithms, and intelligent agents during the scheduling process.

In this study, the approach of Design Science Research (DSR) is used, since it stresses developing and testing novel solutions to real-life challenges. With this project, the DSR approach is a good choice since it allows for developing a working model gradually and encourages in-depth investigations. To begin, the rationale for using this method is explained and connected to what the research is trying to accomplish and its philosophical bases.

Both human experience and numerical results are gathered and examined using this strategy. Having two sides to the approach is valuable since it covers the different aspects of UMIS, highlights where it is lacking, and shows the outcomes of the proposed solution. The information in this chapter includes research design, choosing the methodology, data collection techniques, analysis, tools for building the system, and ethical matters. The aim of using this method is to ensure that the system provides technical comfort and suits the requirements of stakeholders where it will be used.

3.2 Research Paradigm

The researchers depend on the research paradigm to frame both the ideas and strategies for the study. Saunders' Research Onion model is used here to explain what ontological and epistemological positions were taken to deal with challenges in lecture hall scheduling using data-driven and AI-powered system. For the study to remain on track, picking the right paradigm is necessary so it supports the objectives, research questions, and the nature of the problem.

It follows positivism because this study uses measurable measures, examines data objectively, and relies on statistical analysis to decide on outcomes. Also, deductive reasoning has helped to analyze and apply optimization and predictive analytics theories in academic scheduling. Most of the strategy depends on using the following procedures: surveys, observations, and reviewing data from the past, to improve and evaluate the planned solution.

The analysis also makes use of a single quantitative method and reviews how the system functions over a specific period of time. All the decisions in the project are thought out based on studies and needs in the real world of managing the university's resources.

3.2.1 Philosophy

The research uses positivism as its main philosophy, since this matches the quantitative emphasis and importance of objectivity and being generalizable. Positivism holds that the world is orderly, and facts can be noticed and measured, and the researcher is expected to remain impartial during the study. In connection with this study, the goal is to analyze lecture hall use, locate any inefficiencies, and suggest an AI-assisted solution by using real data such as how the halls are booked, how many students can fit inside, and their availability.

Moreover, positivism is a good choice as the study introduces data from surveys and old management records, which can be evaluated statistically and with the use of predictive tools. As a result, positivism is useful because it enables us to observe and investigate things in a reliable manner.

Interpretivism and pragmatism were also looked at, but it was decided they weren't suitable enough for the study. In-depth qualitative research such as ethnographic studies makes Interpretivism more appropriate, because it puts emphasis on personal experiences and subjective understanding. To keep the study clear and consistent, pragmatism, which would have worked for any type of research, was missing in this quantitative research.

Through adopting a positive worldview, the examination conducted in this research is methodical, repeatable, and objective while assessing the impact of data and AI on automating the management of a lecture hall in Faculty of Computing at NSBM Green University.

3.2.2 Approach

Since the study is quantitative, a **Deductive Research approach** was used to test the study's theory-based assumptions. A deductive method starts by using general theories and principles, creates hypotheses or assumptions from these, and studies them by analyzing collected data.

For this research, the theory is built upon resource optimization, predictive analysis, and decision-making supported by artificial intelligence (AI). With this as a base, the study looks into whether using **predictive models**, **genetic algorithms**, and **intelligent agents** can improve the way lectures are scheduled at NSBM Green University. The purpose of the study is to establish if the use of these tools leads to more efficient timetables, cope with paperwork more easily, and improve space usage.

While the inductive system of approach serves as an introductory process and is suitable for coming up with theories from qualitative observations, the deductive approach, however, permits this research to go from a general model data-driven decision-making in educational resource management to definite testable outcomes. Such approach provides a guided approach to hypothesis testing, thus, contributing to the objectivity and scientific rigor of the research.

3.2.3 Strategy

The research uses survey methods because it follows a quantitative and deductive research approach. Surveys serve as an effective method to collect standardized data from large amount of student groups while being well-suited for identifying issues and expectations in structured environments such as lecture hall management. The main goal of this strategy involves obtaining quantifiable data from multiple stakeholders who include students and academic staff and administrative personnel at NSBM Green University.

The study uses structured questionnaires to evaluate the operational difficulties of UMIS and to understand user perceptions and preferences regarding data-driven and AI-enhanced scheduling features. The research strategy enables hypothesis testing and statistical validation of results which is essential for assessing the proposed system's effectiveness.

- **Why Use the Survey Strategy:**
    - **Wider Reach -** The survey method allows researchers to gather data from numerous participants during a short period of time.
    - **Standardization -** The standardized approach enables quantitative analysis and comparison through uniform responses.
    - **Stakeholder Inclusion -** The system allows different user groups including administrators and lecturers and students to provide their inputs.
    - **Efficiency -** Requires fewer resources and provides quicker feedback compared to interviews or experiments.
    - **Data Validity -** The method produces data that is valid for statistical analysis and hypothesis testing purposes.

➢ **Why Not Use Other Strategies?**

- **Case Study -** The study I reviewed was narrow in its scope and was not suitable for reaching conclusions that could be used in other cases.
- **Experimental Approach:** This method does not allow enough flexibility for learning about people's opinions and is only useful for some kinds of research.

In brief, use of the survey approach is well suited for achieving the goals of the research because it ensures detailed examination and useful feedback on issues and suggested improvements with lecture hall resources.

3.2.4 Methodological Choice

The research uses a **Mono method quantitative approach** which depends on a structured questionnaire as its primary data collection tool. The research objectives support the selection of quantitative methods because they need to measure user perceptions and system limitations and scheduling inefficiencies in the current UMIS framework at NSBM Green University. The quantitative method serves as the base to test hypotheses and validate assumptions through statistical tools.

The mono-method strategy enables researchers to obtain measurable data that allows for comparison between different elements while supporting the deductive reasoning approach used in this study. The analysis of closed-ended questions enables researchers to detect patterns and connections between system usability and resource allocation problems and stakeholder satisfaction.

➢ **Why use the Mono Method Quantitative approach?**

- It makes it possible to test a hypothesis by looking at real data.
- Can work with large groups of people, which increases the accuracy of the results.
- Enables you to evaluate both user comments and the system results in a fair way.
- Suitable for places where learning is organized, for example, in classrooms.

➢ **Why Not use a Qualitative approach?**

- **Qualitative:** The evaluation of system-wide problems required exclusion of interviews because they produce subjective results that cannot be easily expanded to multiple cases.

In summary, research design aligns with the mono method quantitative approach because it produces data that remains consistent and analyzable and suitable for system enhancement planning.

3.2.5 Time Horizon

This research adopts a **longitudinal time horizon approach**, as it involves the analysis of data collected over a continuous 12-month period (from January to December 2024). The dataset includes comprehensive lecture hall timetables records, laboratory sessions (lab name, date, time), exam-period scheduling conflicts, special event usage (such as academic conferences), and patterns of frequent and infrequent lecture hall utilization at the Faculty of Computing, NSBM Green University.

Following the longitudinal approach reveals when and why things happen in seasons, how demand changes, and when recurring scheduling issues occur, which is vital for making a successful AI-based system. In contrast to doing a cross-sectional study, this method lets us see the ups and downs of scheduling and discover problems with the system, so the proposed solution can address these issues better and with data. Because of the extended period, the results are still important and useful for new system processes related to analytics and making informed decisions.

Overall, looking at data over a long period (longitudinal) of time both enables accurate predictions and matches the study's goal to help solve problems in academic scheduling. With all the academic-year-based trend analysis, the research makes sure the system deals with real-life complexities and is flexible enough for every possible type of scheduling demand, resulting in a smarter and future-oriented approach for managing lecture halls.

3.2.6 Techniques and Procedures

This research adopts systematic methods and data are used to collect information and shape the design of an AI-powered lecture hall system. The methods used ensure a comprehensive understanding of both current operational challenges and potential improvements.

- **Primary Data Collection Techniques:**
    - **Structured Questionnaires:** Questionnaires are prepared so that students, instructors, and staff can all give consistent and measurable responses about how they use the systems, their scheduling issues, and how pleased they are.
    - **Observations:** informal observations of lecture hall usage trends and scheduling conflicts to corroborate user-reported experiences.
    - **Historical Data Analysis:** Collecting and analysis of timetable data from 2024, including,
        - Lecture and lab usage logs.
        - Exam period hall conflicts.
        - Emergency and ad-hoc event scheduling.
        - Identification of most and least used halls.
        - Lecture Attendance Records.
        - Feedback Logs from Staff or Students.

- **Data Analysis Tools:**
    - **Microsoft Excel:** Used for initial data tabulation, filtering, and visualization of patterns and conflicts.
    - **Python (pandas, NumPy, matplotlib):** Python are used for data analysis and preparation to assist with predictive modeling.
    - **Machine Learning Tools (TensorFlow):** When dealing with large data and making models that can adjust over time, TensorFlow is a fine choice.

Integrating many different sources of data and tools in this study helps to build the suggested system on a firm empirical basis, supporting both smart scheduling and the right use of any resources involved.

3.3 Methodology Selection Criteria

Identifying the right research methodology is the basis of developing strong, valid, and objective research. The methodology adopted in this research is to develop a data-driven and AI-powered lecture hall management system in NSBM Green University should not only be able to accommodate the intricate dynamics of academic scheduling but also enable predictive and decision-support functions. It is based on this multifaceted nature that the selection of appropriate methodological approach was driven by a combination of both theoretical concerns and practical viability.

In this section the criteria on which the choice of the research design is justified is outlined, namely its philosophical alignment (positivism), deductive reasoning, and quantitative orientation. These decisions were determined by the characteristic of the issue - the evidence needs to be quantifiable, and the data should be gathered on a massive scale and across several groups of stakeholders. More so, the methodology had to support the combination of historical data analysis, user feedback, and technical system assessment.

The methodology was also evaluated for its ability to:

- Keep in line with research objectives and questions.
- Facilitate the utilization of systematic instruments like survey and statistical analysis.
- Promote time-effective, but thorough data gathering.
- Permit hypothesis confirmation by means of dependable, measurable outcomes.
- Be scalable to incorporate predictive elements like genetic algorithms and intelligent agents.

Through these criteria, the study makes sure that the selected methodology is able to fill the gap between the academic needs and the practical deployment of an AI-enhanced lecture hall scheduling solution.

3.3.1 Selection Criteria

The methodology that will be used in this study- which is mainly quantitative and deductive- is purposely picked to meet the main research aims, as well as sufficiently addressing the research questions that will be asked. The research proposes to plan and verify a data-driven and AI-enhanced lecture hall administration framework with predictive timetable and smart asset appointment abilities. Achieving these objectives requires the application of a methodology that will allow the gathering of scale, measurable data that can be statistically analyzed and upon which a model can be developed to enhance scheduling efficiency.

The quantitative method and specifically the structured survey enable the researcher to obtain standardized feedback across various groups of stakeholders like the students, lecturers and the administrators. This will be essential in establishing pain points in the existing system (UMIS), user expectations, and identifying practical requirements that has to be considered in system development. Moreover, the application of this method makes the results generalizable and statistically valid, which can be used in decision-making on institutional levels.

Methodology also allows the inclusion of historical timetables and utilization data to permit trend analysis and implementation of machine learning approaches such as predictive analytics and genetic algorithms. Such methods need clean, structured data usually found in quantitative research. As such, the given approach will provide consistency among the research plan, data requirements, and technical aspirations of the suggested solution.

3.3.2 Relevance of Quantitative Approach

The character and volume of this investigation, that focuses on optimizing the control of lecture halls with the help of predictive analytical tests and AI, dictates the requirement of the quantitative approach because of its adequacy in terms of objective and numerical study. Quantitative techniques would enable the gathering of tangible data involving a wide range of stakeholders including students, lecturers and administrative personnel by use of structured questionnaires and subsequent statistical verification of inefficiencies in the scheduling systems, usage of halls and satisfaction with the currently existing systems such as UMIS.

The major rationales to choose this method are:

- **Objective Measurement**: It works with empirical analysis that is needed to develop predictive models and optimization tools.
- **Structured Data Collection**: Questionnaires offer a standard method of obtaining feedback on different user groups.
- **Trend Identification**: Historical timetable data may show when the busiest times are, patterns of conflicts and halls which are not being used enough.
- **Tool Compatibility**: The data could be analyzed with the help of Microsoft Excel and this aspect could be used to test models and prove hypotheses.
- **Generalizability:** The insights are applicable to the whole Faculty of Computing, which supports the utility of results.

Although qualitative approaches provide valuable context information, they are not scalable or standardized, which is required to train algorithms and model systems. Therefore, whereas qualitative feedback could be used in subsequent rounds the quantitative method would be most useful in meeting the fundamental requirements of the research, which are the optimization of the system, the predictive validity and the scaling of the resource allocation enhancements.

3.3.3 Evaluation of Design Method

The study embraces survey strategy as the basic approach of data collection, and this approach fits well into quantitative nature of the research. Surveys will allow the researcher to collect standardized data on a vast and wide range of stakeholders in the Faculty of Computing comprising of administrators, lecturers and students. The method can be objective, scalable and efficient in terms of capturing information that is pertinent to the lecture hall scheduling practices, resource usage and system expectations.

Other methods like interviews and ethnography were also thought of but finally not chosen due to the following reasons:

- **Interviews**: Although they provide more qualitative information, interviews are time-consuming and not feasible to use when a vast number of people need to give their input. Also, interviews can lead to interviewer bias and subjectivity which constrains the standardization required to analyses quantitatively.
- **Ethnographic studies**: Ethnographic research (long-term observations in the field) was identified to not be suitable because of the lack of access to long-term institutional environments as well as the time scales necessary to gain significant insights.

The survey strategy has several advantages over these options which include:

- Effective gathering of information amongst many people (students, lecturers, staff).
- Commonality with statistical analysis packages.
- Limited researcher impact, resulting in objectivity.

Overall, the survey methodology fits well within the research purpose, justifies the deductive methodology and will guarantee obtaining credible data required to guide the design and validation of a predictive, AI-enabled lecture hall management system.

3.4 Method of Data Collection

The gathering of appropriate and precise data is the core element of success of any empirical study, especially the one aiming towards designing a data-driven and AI-enhanced framework of optimizing the scheduling of lecture halls. In this section, the researcher will describe the methods that will be used to get both secondary and primary data that will aid in meeting the research objectives. The main gathering was conducted in the form of structured questionnaires to the important stakeholders such as students, lecturers, and administrative staff of the Faculty of Computing at NSBM Green University. These are the stakeholders that are directly part of the existing scheduling ecosystem and that suffer the most constraints and inefficiencies of the incumbent.

Also, the usage history of lecture halls, timetable constraints, examinations periods, and special events during the previous academic year were gathered to aid in pattern mining and forecasting models. Such a two-pronged strategy will help the research to not only provide the subjective experiences of the users of the system but also collect the objective data needed to train and validate the proposed AI-enhanced scheduling models. Qualitative feedback in combination with quantitative metrics would enable a more comprehensive overview of how well the system is functioning and what its weaknesses are and, therefore, serve as a firm basis towards creating an intelligent, adaptive solution.

3.4.1 Data Type

In this research, both primary and secondary data types are leveraged to obtain a comprehensive understanding of current scheduling inefficiencies and the feasibility of a predictive AI-powered lecture hall management system. Primary data provides first-hand information from the system's users, while secondary data provides access to past context and allows for trend analysis, which is especially crucial for predictive model training and testing.

- **Primary Data Collection (Quantitative)**

  - **Structured Questionnaires:** Administrator, lecturers and students received these questionnaires in order to gain insight into the experience of using UMIS.
  - **Survey Metrics:** Emphasis on the frequency of use of lecture halls, satisfaction, and inefficiency perceptions, and preference on predictive features.
  - **Stakeholder Representation:** Ensures diverse input across all major users of the system.
  - **Data Analysis Tools:** Tools such as excel and SPSS will be used to analyze collected data to identify patterns and be used to guide predictive modeling.

- **Secondary Data Collection (Historical/Institutional Records)**
  - **Lecture Timetables (2024):** Complete details and data of lecture timetables throughout the Faculty of Computing.
  - **Lab Session Logs:** Books for scheduled lab utilization to identify overlaps or inefficiencies.
  - **Exam Period Scheduling:** Records of conflicts and overbooking among lecture halls during peak-demand study periods.
  - **Event-Based Usage:** Statistics for special study or cultural event use (symposium, workshops).
  - **Utilization Trends:** Identification of busiest and least busy halls by frequency of use and scheduling density.

These two types of data are complementary to each other and allow having a holistic, data-informed picture of the existing constraints and assist in the creation of the optimized lecture hall management system.

3.4.2 Sampling Framework

The sampling framework defines the outline or framework of gathering pertinent data of the major stakeholders of the Faculty of Computing, NSBM Green University. It revolves around collecting the opinions of the individuals who have direct contact with the lecture hall scheduling system at the university to recognize the drawbacks of the current solution and learn more about the possibility of utilizing a data-driven solution that employs AI.

Considering the situational constraint faced in obtaining a list of the whole university population, the study will use a non-probability sampling technique, that is, purposive sampling. This will facilitate purposive gathering of information as the opinions of people directly involved in the process of scheduling such as students, lecturers and administrative staff will be sought. This is the way to make sure that the collected information is topical, knowledgeable, and corresponds to the goals of this study. To make findings more solid, and to have a wider representation,

- A total of **250 participants** will be surveyed, composed of:
  - **150 students**
  - **70 lecturers**
  - **30 administrative staff**

This sample size will have a wide range of coverage in terms of the user roles, which will give a balanced opinion in terms of the people who have different experience with the current UMIS system. The purposive approach justifies the informative purpose of this study to feed a predictive and adaptive scheduling model by prioritizing on depth, relevance, and utility of stakeholder responses.

3.4.3 Questionnaire Design and Development

In this study, the questionnaire was well structured, and it was intended to be the chief instrument in the collection of data. In order to quantify the perceptions and experiences of different stakeholders who will be using the current lecture hall scheduling system (UMIS), it was designed in such a way that perceptions and experiences of different stakeholders were to be checked on a 5-point Likert-scale that would consist of choices like: Strongly Disagree to strongly Agree. This scale is amenable to analysis, and it retains depth of responses.

Questions were categorized according to the theme sections under which they were grouped in line with the objectives of the research to ensure that question flow and understandability is logical and understandable. Such themes are:

- **System Usability and Accessibility**
- **Perceived Scheduling Conflicts**
- **Availability and Suitability of Lecture Halls**
- **Expectations for Predictive Features and AI Integration**
- **Satisfaction with Current Administrative Efficiency**

Every section is aimed at certain revelations of the problems and requirements of users in interacting with existing scheduling mechanisms. After the questionnaire design, it was pilot tested on a sample of 10 respondents (students and staffs) in order to assess the clarity of the questionnaire, remove ambiguity and sharpen the wording of the questions. The feedback on the pilot test received was followed by slight corrections so that all factors are simple to comprehend and lay in accordance with the analytical aims of the research.

This universally applicable structure would improve the validity, reliability and relevance of the questionnaire with regard to the predictive analysis and data-driven objectives of the study.

3.4.4 Instrumentation and Administration of the Questionnaire

The questionnaire was conveniently distributed using online media, particularly through Google Forms, so that students, lecturers, and administrative staff in the Faculty of Computing in the NSBM Green University could easily access this questionnaire. This online method was more effective in terms of the prompt collection of data, the monitoring of answers in real-time, and a few logistics involved.

The participants also went through the questionnaire that was in English language since this is the language of instruction at the university, hence understanding and making it clear to all the participants. It was identified that participation was fully voluntary, and the respondents could leave without providing any explanation. To ensure honest and not biased answers, full anonymity was provided, with no details of a particular person being captured.

The survey would be open over a period of two weeks, and several reminders would be made through the faculty mailing lists and academic coordinators. This method allowed achieving a good response rate and adapting to the academic schedule of the participants.

The administration process facilitated by a familiar digital database tool and a well-documented statement of ethical safeguards also favored a facilitating, respectful, and non-wasting process of data collection.

3.4.5 Reliability, Validity, and Generalizability

Reliability

In order to determine the reliability of the questionnaire a Cronbach analysis was checked. This is a statistical indicator which calculates internal consistency between the items of a survey. Any Cronbach's Alpha $\alpha > 0.7$ is considered reliable in any social science research. The overall reliability score values of the important part of the questionnaire in this study were above this mark and thus this indicates that the items utilized in this study were steady and gauged their target construct in similar ways.

Validity

- **Content Validity**

  academic experts and industry professionals who had a level of expertise in information systems in addition to experience in administering universities were consulted and their advice sought when in finding ways of reviewing the questionnaire. Their reviews were used to perfect phrases of questions, relevance and grouping of themes. This procedure was effective because it helped to address the intended subject matter of the survey properly and in correspondence with the objectives of the research.

- **Construct Validity**

  In order to demonstrate construct validity, the questionnaire questions were based on pre-tested and published questionnaires of several studies regarding the scheduling systems, management of educational resource management, and AI-based decision support tools. The approach strengthened the validity and consistency of latent constructions being observed.

Generalizability

Although the investigation was carried out with a large size of the population and diversity of the population included in the Faculty of Computing, there are limitations in the generalizability of the study. The sampling is conducted on one institution only-NSBM Green University which restricts the scope of inference of the same results to other faculties or even to other universities with a different operational framework or demographic status. However, the number of respondents and the variety of them (students, lecturers, administrators) justifies the representative insights into the research in this framework.

3.5 Research Methodology Execution Workflow

3.5.1 Problem Identification

The problem that is identified in this research is related to the inefficiency of the existing lecture hall scheduling system (UMIS) in NSBM Green University. This system faces difficulties such as double-booked lecture halls, rooms not being used to their full potential and lack of predictive scheduling ability. The issue was identified from the feedback of stakeholders, including administrators, lecturers, and students, who were frustrated with the timeliness issues, rigidity and outdated system features. By examining the problems of the current system, the requirements for an AI-based solution to optimize scheduling, enhance the utilization of resources, and offer predictive capabilities became apparent.

3.5.2 Relevance Justification

This research is of great importance especially in the context of fast-growing universities such as NSBM Green University. The research is relevant since it is intended to overcome the limitations of conventional scheduling system. Existing systems like UMIS lack the functionality to predict demand in the future or adapt to the scheduling dynamically based on real-time changes. The advent of artificial intelligence (AI), in the form of predictive analytics and genetic algorithms, will deliver a more efficient, flexible and scalable solution. This research will not only benefit NSBM but also will help other educational institutions having similar scheduling challenges, thus adding value to the larger smart campus technologies.

3.5.3 Comparative Analysis and Gap Justification

A comparative study between the current UMIS system and more sophisticated AI-based systems showed huge gaps in functionality. UMIS only supports basic scheduling work and misses advanced features such as predictive scheduling, real-time adjustment and decision-supporting capabilities. On the other hand, more advanced systems, especially those that are used in universities in Europe and Asia, employ AI technologies such as predictive analytics and genetic algorithms to enhance scheduling efficiency and minimize conflicts in resource distribution. The results from this global system showed a decrease of up to 40% in scheduling conflicts and a considerable increase in the utilization of resources. The discrepancy observed between the state-of-the-art systems and UMIS justifies the need for a better and intelligent scheduling system in NSBM Green University.

3.5.4 Define and Finalize Objectives

The primary goal of this research was to develop and deploy an AI-driven lecture hall management system that optimizes scheduling, enhances resource allocation, and offers predictive analytics to enable administrators to make informed decisions based on data. Specific objectives are:

- Developing an artificial intelligence (AI) based system using predictive analytics to stimulate lecture hall demand.
- Genetic Algorithms for scheduling optimization problems, taking into account different constraints (rooms capacity, equipment requirements, preferences of lecturers)
- Developing a user-friendly interface for administrators, lecturers and students to interface with the system and view relevant data.
- Experimentation with the system on real test cases to measure the system's impact on scheduling accuracy, resource utilization, and user satisfaction.

3.5.5 Design / Development / Data Management and Handling

In this research, the development phase has been designed in modular system architecture with major AI components like predictive analytics (demand forecasting) and genetic algorithms (optimizing schedules). The data management process included gathering historical scheduling data from UMIS at NSBM as well as real-time data on class size, equipment requirements, and lecture hall usage. This data was processed beforehand in the creation of predictive models that were used in real-time decision-making for scheduling. The system is scalable for growing volumes of data as the university expands and is integrated with existing infrastructure.

3.5.6 Evaluation and Communication

The system was tested both functionally (to verify that the basic functionality such as scheduling and demand forecasting was working correctly) and non-functionally (to evaluate the performance of the system under high load, usability, and scalability). Different testing methodologies such as load testing, stress testing, and usability testing were used to assess the effectiveness of the system and its ability to cope with real-world usage scenarios. The findings were then presented to stakeholders using interactive dashboards and reports which allowed the administrators, lecturers and students to provide feedback for further system enhancements. In addition, the performance of the system at various prediction accuracy and the improvement of system resources were analyzed and published to the University for future optimization.

3.6 Constraints and Considerations

The research was properly designed to reach the intended goals, but several limitations were also identified that needed to be circumvented along the line. These constraints take both logistical, technological, and ethical forms and all of them determine the mode of data collection, processing, and interpretation. These limitations are imperative in order to put the results into perspective and counter the fact that the dimension of the research has not been correctly served.

Complications in obtaining complete information on lecture hall usage to constraints of time in carrying out pilot survey and expert review, the study process was forced to be subjected to numerous practical matters. Furthermore, the necessity to use voluntary participation and online data collection procedures added limitations in terms of the sample segmentation and the accuracy rates of the data. Next, a combination of predictive technologies, such as Genetic Algorithms and intelligent agents, also posed development challenges that needed testing and improvement in a loop.

By discussing these factors openly, this section achieves realistic perception of the results obtained in the research and it has a structure that can be used in perfecting other studies carrying the same nature in future. It also reemphasizes the significance of flexibility and critical reflections in effecting sound academic studies especially in dynamic and institution-based settings such as in higher education.

3.6.1 Time Constraints

One of the limitations in the conducting of the research was time, considering the academic program and the first-then last structure of the project. Though the study was conducted for a few months, its data collection period was restricted to the specific time frame since after according to the university calendar, during the exam period, and availability of the staff.

Key time-related limitations included:

- **Academic Calendar Issues**: Data collection was required to be structured around semester vacations, test weeks and University events.
- **Stakeholder Availability**: Students, lecturers, and members of the administrative staff did not have a lot of time to take part in surveys or give feedback because of academic commitments.
- **Pilot Testing and Development**: The designing of the questionnaire and the conducting of the pilot and getting into the revisions took a significant source of the time available.
- **Historical Data Compilation**: Although data on the history was retrieved in UMIS records on the year 2024, due to time constraints, the data could not be covered on a larger basis of years of analysis.

Despite these limitations, effective data was collected by adopting strategic planning. The results updated knowledge of fair lecture hall usage and its inefficiency of management, which grants the basis of the case under the concept of the proposed AI-powered system. Future work may be, however, facilitated with a longer time frame to conduct a more in-depth process of validity and comparison.

3.6.2 Financial Constraints

This was a research project that had quite significant financial constraints and this affected some of the major areas related to the project conduct. First of all, the budgetary limitations did not allow other faculties (Business, Engineering, or Science) so the study was restricted to the Faculty of Computing of NSBM Green University. The study sample was limited only to the faculty of engineering, and such a limitation will not be able to give the larger picture of what is going on in the university in terms of managing lecture halls; a situation that could have been achieved by extending the study to other faculties though this was not possible given the time and resources available.

Furthermore, at the design and implementation stages of the project, much free and Open-source software was used to obviate proprietary software costs. Data was collected on platforms like Google Forms, whereas Python, Pandas, and Matplotlib were utilized in data analysis and visualization. Statistical calculations were also done on Microsoft Excel. For machine learning will use TensorFlow. These tools provided efficient and affordable solutions to data processing and modeling thus fulfilling the quality of the conducted research in spite of financial limitations.

The absence of funding limited the possibility of field research as well as conducting on-site interviews or on-site workshops with the involvement of external stakeholders. Travel charges, printing costs and other software costs had to be kept to the minimum. However, the study was effectively completed with the help of properly calculated utilization of the university resources available, peer assistance, and organization. In subsequent studies, external funding or inter-college interventions might assist in eliminating these monetary issues and expand the range of the study.

3.6.3 Data Access Constraints

Availability of applicable and comprehensive information was a major concern during the process of carrying out this study. This was not always easy as due to the project being specific to the Faculty of Computing at NSBM Green University and did not concentrate much on the other internal systems like the UMIS; it was not easy to get greater datasets from it. The constraints that were noticed were as follows:

- **Administrative Permissions:** Access to UMIS data necessitated institutional policy findings. Gaining access to such permissions meant there were several stages of clearance, and they slowed aspects of the research.
- **Incomplete/Unstructured Data:** Not all data on the use of the lecture halls was registered or digitalized, especially when it comes to non-academic uses (workshops or ad hoc requests).
- **Data Confidentiality:** Sensitive information like class specific feedback, employee schedules, and internal dispute resolutions could not be provided to preserve the privacy of the institution, and a more detailed analysis could not be done.
- **Limited Real-Time Data Access:** Restricted real-time data access the UMIS platform available at present does not allow API integration or real-time data access, which limits the ability of dynamic data modeling.

The research, in order to overcome these limitations, used:

- Stakeholders designed structured questionnaires and surveys to cover missing information.
- During peak usage periods, observational data will be obtained.
- Use of publicly accessible timetables and schedules.

In spite of such obstacles, the study successfully modified its methodology and got it working on the available data ecosystem without compromising validity and relevance.

3.6.4 Ethical Considerations

The research ethics were extensively adhered to during the research in an attempt to safeguard the rights of the participants and the authenticity of study.

Voluntary Participation

All the participants in the survey process, be they a student, lecturer, or an administrative employee, did so by their initiative. To illustrate, none of the students were contacted by classroom announcements under which they were required to attend the meeting but rather links to the survey were forwarded via email and faculty discussion forums, so each of the participants had a choice in regard to time and frequency of their response.

Data confidentiality

No name, student ID, or any such information was asked. The responses were saved under a locked, password-protected cloud drive. An example is that data were analyzed jointly to find out trends such that individual data could not be isolated to come up with a meaning.

Ethical Approval

Although the university did not require this type of study to pass through a formal ethics board on such a level, the work was carried out under the supervision of the research supervisor, who was supportive of following the NSBM ethical standards of research. In case future growth concerns more than one faculty, or external stakeholders, official approval procedures will be put into motion.

Informed Consent

The purpose of the study, the anticipated duration of the research, the possible risks and the rights of the participants were clearly and in written form explained to respective participants before they were admitted to the questionnaire. They had to confirm an informed consent agreement box only after which they could start the survey. To take an example, there was a short preamble at the head of the Google Form that clearly mentioned these terms.

3.7 Conclusion

The chapter defined the systematic approach that was to be used in implementing a system of lecture hall management using data and AI at NSBM Green University. The choice of positivist research philosophy and deductive approach was determined by the purpose of the study to conduct controlled tests on the hypothesis, using previous theories and real-life facts. This premise guaranteed objectivity and coherence during the research.

It was based on quantitative mono-method design, the main data collection method was structural surveys. This method helped to obtain a big volume of homogeneous information about the students, academic personnel, and administrators. Planning technologies were also used giving the ability to establish the patterns of usage, timing discrepancies, and efficiency of the used resources in the current application UMIS.

Included in the chapter was also the description of the methods and processes of data collection, such as the distribution of the questionnaires through Google Forms and using such tools as Microsoft Excel and Python to analyze data. The need to ensure that all the aspects of the ethics of research are adhered to were duly followed, like informed consent, anonymity, and voluntary participation of the research.

Limited financial resources, no access to wider faculty data and limited time were identified as limitations. They were alleviated via good planning, application of open-source software, and choice of focus on Faculty of Computing as the main case study.

Overall, the methodology adopted in this research was coherent, justified, and fully aligned with the study's objectives. It not only supports the reliability and validity of the findings but also paves the way for a smooth transition to the next chapter, where the collected data will be analyzed and interpreted to inform the design and evaluation of the proposed solution.

CHAPTER 4: System Requirement Specification

This chapter provides the technical basis of the proposed system describing its main requirements and structural elements. It takes the research objectives and findings of the preceding chapters and translates them into specifications of the system in detail that serves as the guideline in development. The chapter starts with the identification of the main stakeholders and their functions in the system and then the research data operationalization into the specific system needs is offered there.

Then it contains a collection of system modeling diagrams like **use case, class, activity and sequence diagrams** that show users' interface with the system and how the internal elements interact. The architecture that is being suggested is also addressed and explains the interaction between various layers and technologies to provide the needed functionality. Lastly, all the functional and non-functional requirements are listed and categorized in the chapter so that the expectations of users and technical standards can be fulfilled.

4.1 Chapter Overview

The chapter provides a description of the system requirements of the proposed data-driven and AI-powered lecture hall management system. It outlines the non-functional and functional expectation of the system, describes the stakeholders that will be involved and displays the behavior of the system in different diagrams namely the use case, class, activity, and sequence diagrams. The chapter starts with a stakeholder analysis and a process of operationalization, which links data-gathering methods (questionnaires and datasets) to the research objectives, maintaining traceability and justifying all the system requirements.

Moreover, the chapter introduces the proposed system architecture and expounds on non-functional and functional requirements to give a clear picture on the scope of the system, system performance expectations and usability requirements. The specifications are the basis of the later design and development stages. The general objective is to make sure that the suggested solution efficiently tackles the existing shortcomings of lecture hall scheduling and resource optimization, especially in the case of the Faculty of Computing at NSBM Green University.

4.2 Stakeholder Analysis

The stakeholder analysis is very crucial to determining individuals and groups that will engage or be affected by the proposed Data-Driven and AI-Powered Lecture Hall Management System. Their needs and expectations can be understood to make sure that the system is planned to be user-centric and operationally relevant.

The table below identifies the key stakeholders, their roles, interests and the degree of influence in the system:

Stakeholder	Role	Interest	Influence
System Administrator	Manages system configuration, access control, and scheduling oversight	High system performance, error-free scheduling, data accuracy	High
Faculty Members / Lecturers	Input schedules, updates lecture plans, views hall availability	Easy-to-use interface, real-time updates, reduced clashes	High
Students	View lecture schedules, receive updates, and give feedback	Accurate timetables, fewer clashes, accessible platforms	Medium
University IT Staff	Ensures infrastructure, maintenance, and security	System stability, integration with UMIS, low maintenance burden	High
University Management (Deans/Heads)	Make strategic decisions using reports and insights	Analytics-driven decision-making, efficient space utilization	High
Event Coordinators	Books halls for workshops, exams, or external events	Real-time booking capabilities, clash-free scheduling	Medium
External Reviewers / Auditors	Evaluate system performance, compliance, or efficiency	Transparent reporting, usage statistics, audit logs	Low to Medium

Table 5: Stakeholder Analysis for Data-driven and AI-Powered Lecture Hall Management System

Key Insights:

- The administrators, lecturers, and IT staff are high-priority stakeholders because they are the ones who operate and maintain the system directly.
- The main users that contribute to system usability are students and event coordinators.
- Space planning and policymaking require management of universities to rely on data analytics and reports.

An accurate perception of the expectations of each stakeholder will ensure that the proposed system will provide specific features that will facilitate the satisfaction of users and organizational objectives.

4.3 Operationalization Process

Operationalization process is concerned with the transformation of the research objectives into actionable data which confers the adequacy of the proposed system requirements which are to be successfully collected, analyzed and integrated. This is the step where the researcher maps the data gathering methods (e.g. survey) with the research aims and evaluates the findings to confirm the requirements chosen. The specifics of lecture hall capacity, exam capacity, and hall characteristics of the Faculty of Computing are part of this process and contribute to developing a system according to the available resources.

4.3.1 Lecture Hall Details

Faculty of Computing at NSBM Green University has diverse lecture halls with varying capacities, which are significant in the context of familiarizing the present-day pattern of scheduling and resource allocation. The table below gives a division of the lecture halls available, the capacity of lecture halls, and the respective capacity of exam halls.

Floor	Lecture Hall			Lab		
	Hall No	Capacity	Exam Capacity	Lab No	Capacity	Exam Capacity
Ground Floor	C2-009	350	118	C2-007	50	40
	C2-002	250	80	C2-008	50	40
	C2-003	120	50			
	C2-005	50				
	C2-006	50				
B1 Floor	C2-L101	175	70	C2-L106	50	40
	C2-L102	175	70	C2-L107	50	40
	C2-L104	50				
	C2-L105	50				
	C2-L110	150	78			
	C2-L109	Network Lab				
B2 Floor	C2-L202	50		C2-L204	50	40
	C2-L203	50		C2-L205	50	40
1st Floor	C2-105	200	80	C2-103	70	40
	C2-106	100	40			

Figure 3: Lecture Hall Capacities and Exam Capacities in the Faculty of Computing

4.3.2 Mapping Data to Research Objectives:

- **Research Objective 1: Test efficiency of existing scheduling system (UMIS).**
  - **Data Gathering Method:** Surveys and interviews with stakeholders such as lecturers, administrators, and students. This will present the constraints and deficiencies to the existing system, such as use of lecture halls and scheduling issues.
  - **Mapped Data:** Feedback on hall use, contentment with existing system, and scheduling problems (e.g. hall is overbooked or occupied with a party).
  - **Lecture Halls Details:** This will entail the evaluation of rates of current use of various lecture halls. As an illustration, big lecture halls (C2-009, capacity 350) might be in demand whereas smaller lecture halls such as C2-003 (capacity 120) or C2-005 (capacity 50) may not be fully utilized during peak timings.
- **Research Objective 2: Forecasting of future scheduling requires AI and predictive analytics.**
  - **Data Gathering Method:** Historical data on lecture hall bookings, exam periods, and hall utilization rates.
  - **Mapped Data:** Evaluation of past schedules, lectures, exam and lab activities, to determine high demand times, under-utilization and schedule overlaps. This information will be used in the predictive design of halls allocation.
  - **Lecture Halls Details:** As an example, information about a hall C2-009 (350 capacity) during exams might reveal the necessity to introduce flexible scheduling solutions. In a similar fashion, the use of labs (e.g., C2-007 and C2-008) in specialized sessions will be considered.
- **Research Objective 3: Design an AI-powered scheduling system with real-time adaptability.**
  - **Data Gathering Method:** The real time data will be collected via live systems and user (lecturers and administrators) feedback to determine flexibility of the system.
  - **Mapped Data:** The idea is to make sure that the system is flexible to unexpected schedule changes, e.g. special events, emergencies (e.g., workshops, hackathons). This will involve the analysis of real time data of halls booking and adjusting accordingly.

4.3.3 Results Review and Validity Justification:

Following the accumulation of the necessary data, the results will be confirmed by matching them with the goals of the research. The project will also be reviewing the information provided by different stakeholders, including the record of the usage of the lecture halls (e.g., C2-009 and C2-105), so that the suggested AI-driven system will fit the real-world issues. The validity and applicability of the data gathering procedure are also enhanced by referring to the real data on the Faculty of Computing lecture hall that assists in narrowing down the model to the requirements and limitations of the neighborhood. This process will ensure that the requirements are informed by current and actionable data which will result in a dependable, effective and flexible scheduling system in the Faculty of Computing.

4.4 System/Model Analysis

4.4.1 Use Case Diagrams with Specifications

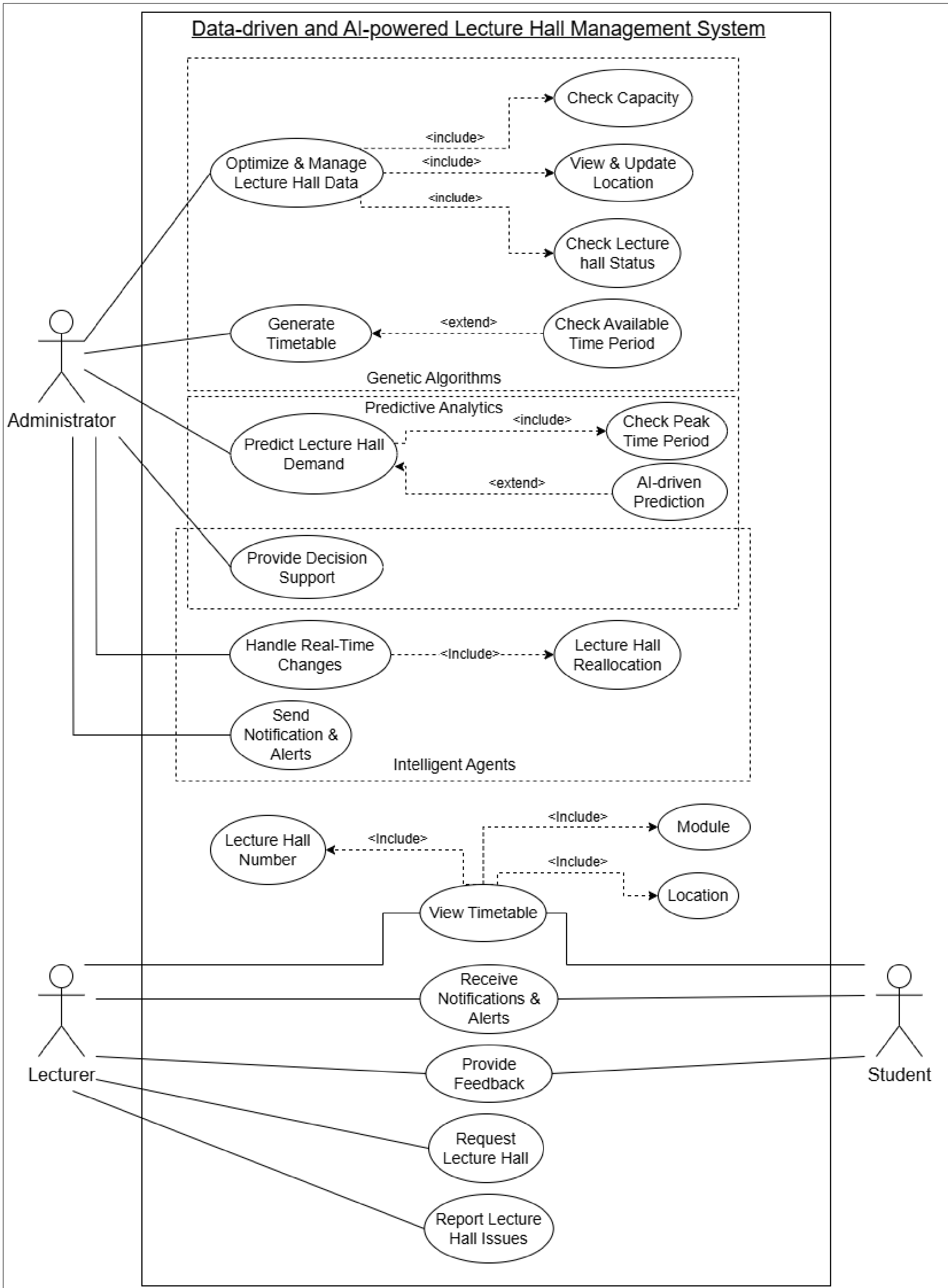


Figure 4: Use Case Diagram of the Data-Driven and AI-Powered Lecture Hall Management System

4.4.1.1 Use Case Analysis of the AI-Powered Lecture Hall Management System

1. **Actors in the System**

- **Administrator**: The main user who manages the overall system, makes decisions, and interacts with various processes like scheduling and allocating lecture halls.
- **Lecturer**: Module leaders or faculty members who interact with the system to check their schedules, request rooms, and report issues.
- **Student**: Students who need to access their schedules, receive notifications, and report issues.

2. **Administrator Use Cases**

The administrator bears the greatest responsibility. This is what they do in the system:

- **Optimize & Manage Lecture Hall Data**
  - Checking on the capacity of halls: Making sure that halls are large enough in terms of the classes.
  - Location viewing and updating: The administrator is able to view and update the location of every hall within the university.
  - Status check: This will assist the admin in determining whether the halls are available, occupied or require some repairs.
- **Generate Timetable**
  - **Genetic Algorithms**: A process that is applied to streamline the arrangement of classes.
  - **Predictive Analytics**: This predicts future demand in lecture halls using previous data to enable the system to make more informed decisions proactively.
- **Providing Decision Support**

  The administrator applies data and algorithms to aid decision making such as on whether a class ought to be rescheduled or transferred to another hall depending on the prediction of demand.

- **Handle Real-Time Changes**
  - **This aspect enables the admin to make corrections on the fly**: To illustrate, in case of a last-minute cancellation or when there is a requirement to transfer a class to another hall, the system will automatically take care of these.
  - **Send notifications and alerts**: This sends instant notifications to the individuals concerned (lecturers, students) when something changes.

- **Lecture Hall Reallocation**

  The process assists the administrator to control the timing, and reasons of assigning a hall to a different activity or class.

- **Intelligent Agents**

  These are automatic systems that assist the administrator:

  - **Check peak time periods**: Automatically forecast and notify the administrator when some periods are busier than others to assist with planning.
  - **AI-based predictions**: Use more intelligent predictions, which can be adjusted over time.

3. Lecturer Use Cases

The Lecturer will engage the system in the following ways:

- **View Timetable**
  Lecturers will be able to view their timetables and know when and where to hold a class.
- **Notification and Alerts Receipt**
  Lecturers are informed of any change in their time schedule or place of holding the classes to keep them in touch.
- **Provide Feedback**
  Some of the ways that lecturers can give feedback on the system is to ask it to improve or to complain that there is a problem with the timeline or the hall assignment.
- **Request Lecture Hall**
  The lecturer can request a lecture hall through the system when he/she requires a different room or special resources (such as additional equipment).
- **Report on Lecture Hall Issues**

In case of issues with the lecture hall (e.g. broken equipment, and lecture hall not available), lecturers can discuss these problems, to be followed up.

4. Student Use Cases

The system is primarily used by the student when engaging with the timetable and staying up to date:

- **View Timetable:**
  Students are able to see and read their respective lectures schedules that contain lecture timetables, halls, etc.
- **Get Notifications and Alerts:**
  The students are informed of changes in their timetable, such as the cancellation of lectures or change in lecture halls.

4.4.2 Class Diagram

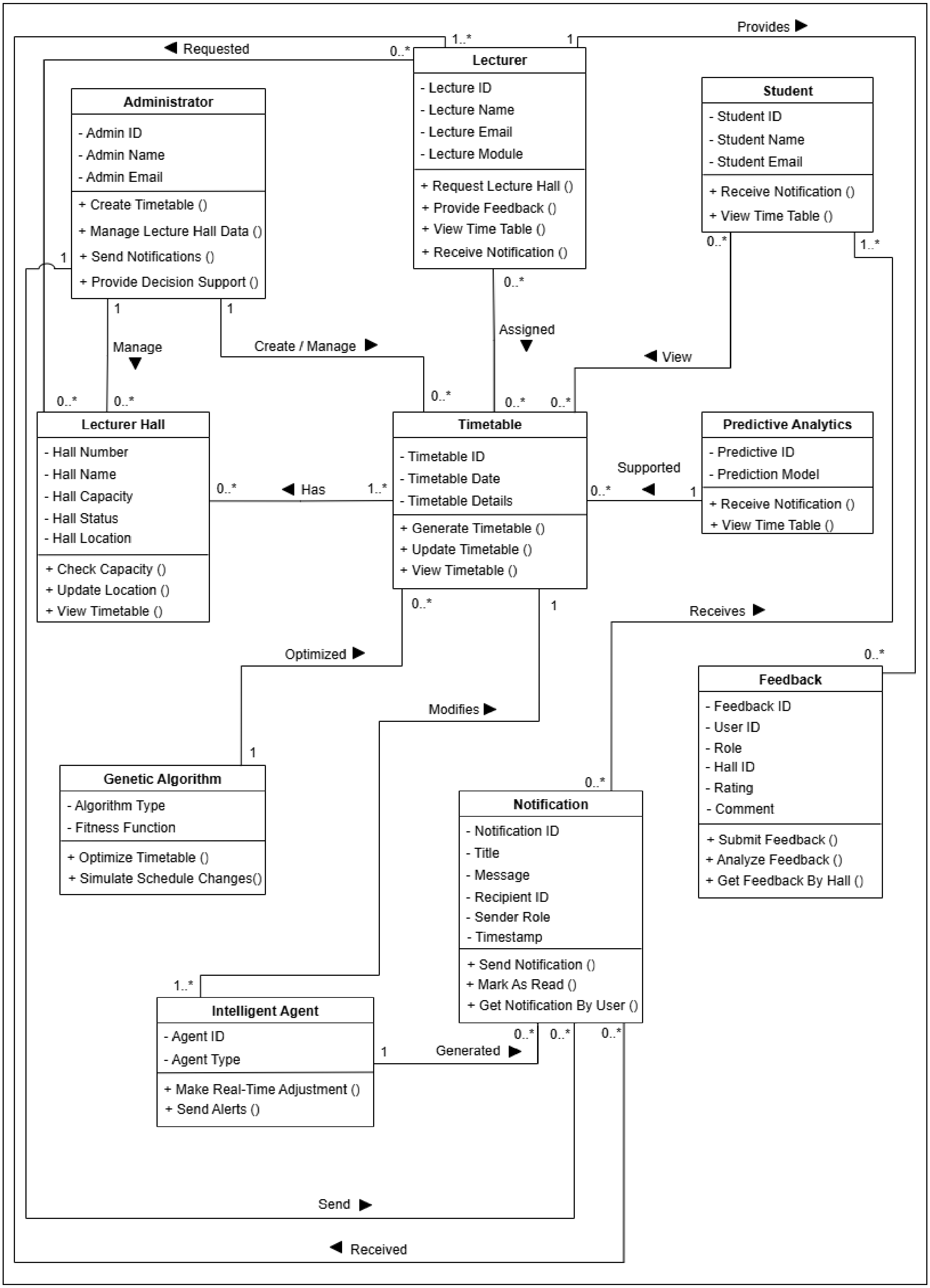


Figure 5: Class Diagram of the Data-Driven and AI-Powered Lecture Hall Management System

4.4.2.1 Class Diagram Explanation

This Class Diagram is used to visualize the structure of the AI-powered Lecture Hall Management System. It describes the major entities (classes) in the system, their attributes (properties), and the relationship (associations) among the entities.

1. **Administrator Class**
   - **Attributes**: Admin ID, Admin Name, Admin Email
   - **Methods**:
     - **Create Timetable**: The admin creates and generates a timetable for lecture halls.
     - **Manage Lecture Hall Data**: The admin manages data related to lecture halls (capacity, location, status).
     - **Send Notifications**: The admin sends notifications to lecturers or students regarding schedule changes or updates.
     - **Provide Decision Support**: Admin uses decision-support tools for managing and resolving scheduling conflicts.
2. **Lecturer Class**
   - **Attributes**: Lecturer ID, Lecturer Name, Lecturer Email, Lecturer Module
   - **Methods**:
     - **Request Lecture Hall**: Lecturers request a specific IT lab for their in-class test.
     - **Provide Feedback**: Lecturers can provide feedback about the halls, timetable, or system.
     - **View Timetable**: Lecturers can view their own timetable via the Outlook.
     - **Receive Notification**: Lecturers receive notifications regarding their schedule or in-class assignments.
3. **Student Class**
   - **Attributes**: Student ID, Student Name, Student Email
   - **Methods**:
     - **Receive Notification**: Students receive notifications about any updates or changes to their lecture schedules.
     - **View Timetable**: Students can view their own timetable, which includes lecture times, lecture halls, and other details.

4. **Lecture Hall Class**
    - **Attributes**: Hall Number, Hall Name, Hall Capacity, Hall Status, Hall Location
    - **Methods**:
        - **Check Capacity**: This method checks the available capacity of the lecture hall.
        - **Update Location**: The location of a hall can be updated as necessary (e.g., if it's moved or renamed).
        - **View Timetable**: View the schedule of the lecture hall to check which classes are assigned to it.
        - **Has**: This relationship indicates that a lecture hall can have multiple schedules assigned to it.
5. **Timetable Class**
    - **Attributes**: Timetable ID, Timetable Date, Timetable Details
    - **Methods**:
        - **Generate Timetable**: This method is responsible for creating and organizing the entire class schedule.
        - **Update Timetable**: The timetable can be updated to accommodate changes (e.g., reallocation of lecture halls).
        - **View Timetable**: Allows the viewing of the scheduled classes for a given date and time.
6. **Predictive Analytics Class**
    - **Attributes**: Predictive ID, Prediction Model
    - **Methods**:
        - **Receive Notification**: The system can notify stakeholders about demand predictions.
        - **View Timetable**: Displays the forecasted schedule or demand for lecture halls.

7. **Genetic Algorithm Class**
   - **Attributes**: Algorithm Type, Fitness Function
   - **Methods**:
     - **Optimize Timetable**: The genetic algorithm is used to optimize the timetable by minimizing conflicts and maximizing efficiency.
     - **Simulate Schedule Changes**: The algorithm simulates changes in the schedule to see how adjustments impact the overall timetable.
8. **Intelligent Agent Class**
   - **Attributes**: Agent ID, Agent Type
   - **Methods**:
     - **Make Real-Time Adjustments**: Intelligent agents help make real-time adjustments when unexpected changes occur (e.g., a lectures cancellation or a last-minute hall reallocation).
     - **Send Alerts**: Sends notifications to affected users (e.g., lecturers, students) about changes.
9. **Notification Class**
   - **Attributes**: Notification ID, Title, Message, Recipient ID, Sender Role, Timestamp
   - **Methods**:
     - **Send Notification**: This method allows the system to send notifications (about schedule changes, room assignments, etc.).
     - **Mark As Read**: Users can mark notifications as read once they have viewed them.
     - **Get Notification by User**: Allows users to view all the notifications they've received.
10. **Feedback Class**
    - **Attributes**: Feedback ID, User ID, Role, Hall ID, Rating, Comment
    - **Methods**:
      1. **Submit Feedback**: Users can submit feedback on lecture halls and the system.
      2. **Analyze Feedback**: Analyzes feedback data to identify any recurring issues or suggestions for improvement.
      3. **Get Feedback by Hall**: Displays feedback specific to a particular lecture hall.

4.4.3 Activity Diagram

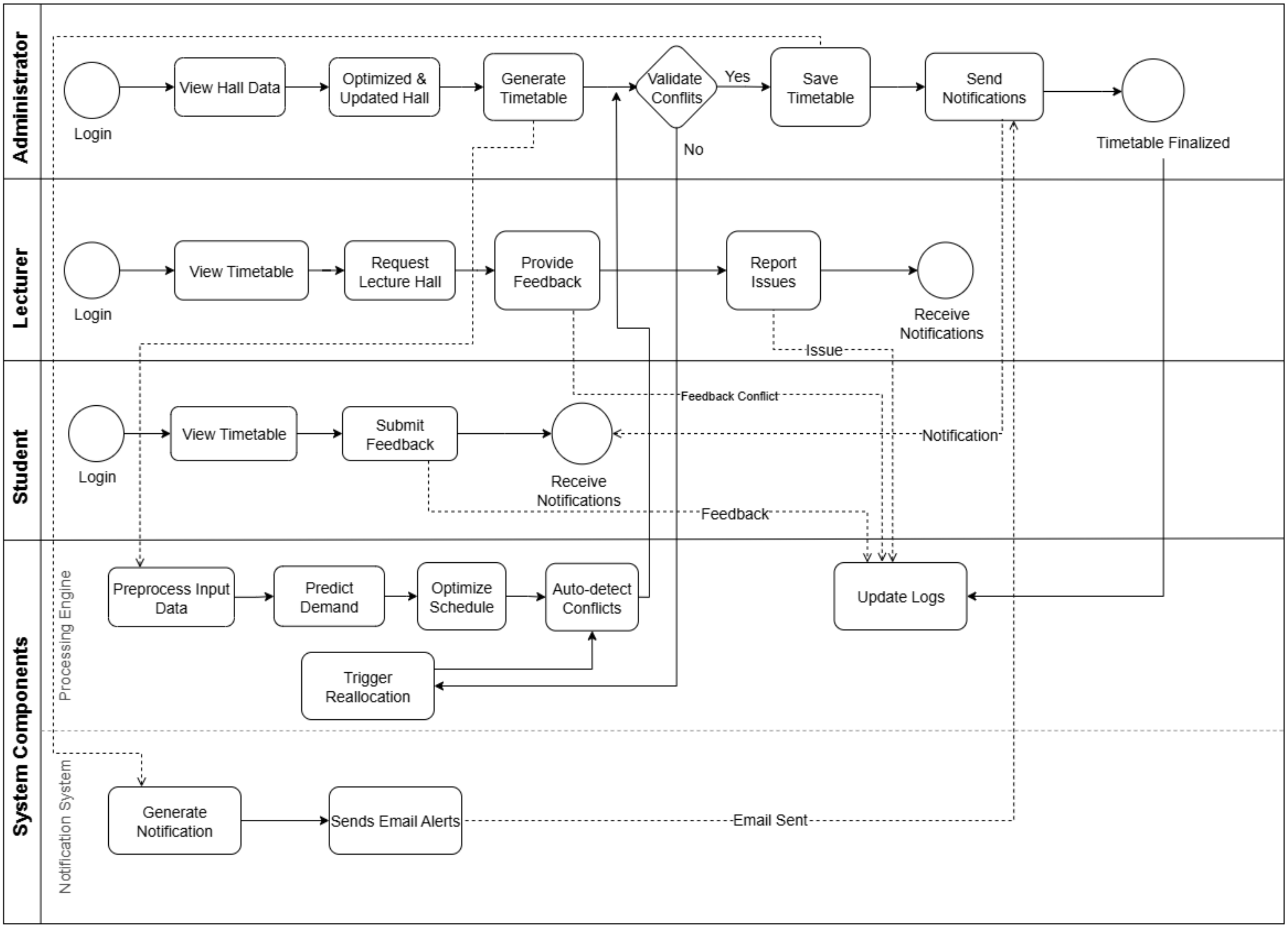


Figure 6: Activity Diagram of the Data-Driven and AI-Powered Lecture Hall Management System

4.4.3.1 Overview of the Activity Diagram

This diagram illustrates how activities and decisions of various users (Administrator, Lecturer and Student) interact with the system. Activities are arranged in columns, with each column corresponding to a user or system component (i.e. the System Processing Engine or Notification System).

Key Elements:

1. **Administrator Activities**:
    - **Login**: The administrator logs into the system.
    - **View Hall Data**: After logging in, the admin views the lecture hall data (e.g., capacity, status, location).
    - **Optimize & Update Hall**: Based on hall data, the administrator optimizes the hall allocation and updates the information as necessary.
    - **Generate Timetable**: After all data is updated, the admin generates the timetable for classes using the system's scheduling tools.
    - **Validate Conflicts**: The system checks for any scheduling conflicts (e.g., double-booking of halls or overlapping schedules). If no conflicts are found, the timetable is saved.
    - **Save Timetable**: Once conflicts are resolved, the admin saves the finalized timetable.
    - **Send Notifications**: After saving the timetable, the admin sends notifications about the finalized schedule to all relevant users (lecturers, students).
    - **Timetable Finalized**: This marks the end of the process for the admin. The timetable is now ready and finalized.
2. **Lecturer Activities**:
    - **Login**: The lecturer logs into the system.
    - **View Timetable**: Once logged in, the lecturer can view their assigned timetable, including the lecture hall and schedule.
    - **Request Lecture Hall**: If necessary, the lecturer can request a specific lecture hall or IT lab.
    - **Provide Feedback**: The lecturer can provide feedback on the lecture halls or the timetable.
    - **Report Issues**: If any problems are found with the hall or schedule, the lecturer can report these issues.
    - **Receive Notifications**: Lecturers will receive notifications regarding updates, changes, or issues with their schedules.

3. **Student Activities**:
    - **Login**: The student logs into the system.
    - **View Timetable**: Once logged in, the student can view their lecture timetable, showing which lectures, they are attending and in which halls.
    - **Submit Feedback**: The student can also submit feedback about the halls or their schedule.
    - **Receive Notifications**: Students will receive notifications regarding changes to their schedule (e.g., hall changes, cancellations).
4. **System Components**:
    - **Preprocess Input Data**: The system processes the data provided by the administrator, lecturer, and students.
    - **Predict Demand**: The system predicts hall demand based on historical data or user input.
    - **Optimize Schedule**: The system uses algorithms to optimize the schedule, making sure halls are efficiently allocated without conflicts.
    - **Auto-detect Conflicts**: The system automatically detects any scheduling conflicts and provides solutions.
    - **Trigger Reallocation**: If conflicts are detected, the system triggers hall reallocation to resolve the conflicts.
    - **Generate Notifications**: The system generates notifications to inform lecturers and students about schedule updates.
    - **Send Email Alerts**: The notifications are sent via email to the relevant users.
    - **Update Logs**: The system logs all actions, such as generating the timetable, user feedback, and sending notifications.

Key Decisions:

- **Validate Conflicts**: The decision that matters the most. In case any conflicts are identified in the timetable, the admin should take measures to resolve them and then save and send notifications.

4.5 Proposed System Architecture Diagram

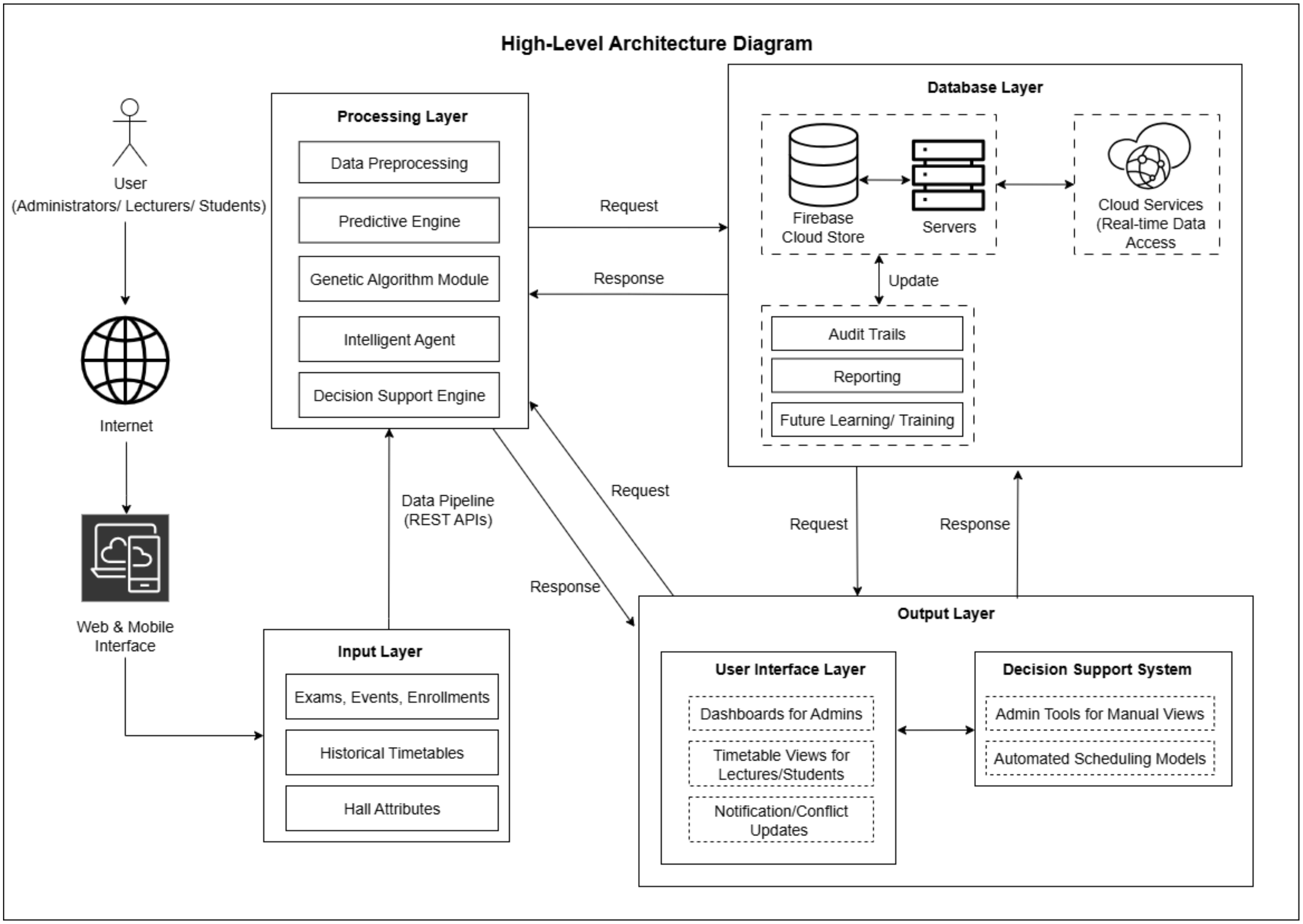


Figure 7: High-Level Architecture Diagram of the Data-Driven and AI-Powered Lecture Hall Management System

4.5.1 High-Level Architecture Description

The proposed Data-Driven and AI-Powered Lecture Hall Management System High-Level Architecture Diagram shows a modular and scalable system architecture, which comprises five main layers: the Input Layer, the Processing Layer, the Database Layer, the Output Layer and the User Interface Components, all connected through a centralized data pipeline and available over the internet.

5. User and Internet Interface

The system is connected to the internet through web and mobile interfaces where administrators, lecturers, and students interact with the system. It is an entry point that allows receiving and sending data and feedback in real-time on several devices.

6. **Input Layer**

This layer gathers structured data from various university operations. Key input components include:

- Exams, Events, and Enrollments
- Historical Timetables
- Lecture Hall Attributes (capacity, facilities, location, etc.)

These inputs flow through a REST API-based data pipeline to the Processing Layer, ensuring secure and consistent data transmission.

3. Processing Layer

This is the system's intelligent core. It includes:

- **Data Preprocessing:** Cleans and formats raw inputs.
- **Predictive Engine:** Applies predictive analytics to forecast future demand for halls.
- **Genetic Algorithm Module:** Optimizes scheduling by solving resource allocation problems.
- **Intelligent Agent:** Enables real-time adaptation and decision-making.
- **Decision Support Engine:** Analyzes scenarios and proposes solutions for admin-level intervention.

The processing layer both requests' data from and saves results to the Database Layer.

4. Database Layer

This layer serves as the data backbone of the system. It includes:

- Servers and Firebase Cloud Store for persistent and scalable data storage.
- External Integration and real-time data access via Cloud Services.
- Audit Trails, Reporting Modules and Machine Learning Training sets facilitate monitoring, feedback, and evolution of the system.

5. Output Layer

The processed data is relayed to two sub-layers:

a. User Interface Layer:

- Dashboards for Admins
- Timetable Views for Lecturers/Students
- Notifications & Conflict Updates

b. Decision Support System:

- Administrative Tools - manual tools
- AI-powered Automated Scheduling Models

This output supports real-time decisions, transparency, and data visualization across stakeholders.

4.6 Functional and Non-functional requirements.

4.6.1 Functional Requirements

Functional requirements outline the primary functions and feature that the AI-powered Lecture Hall Management System must possess. These are directly related to the system's intended functionality as extracted from the stakeholder needs and research objectives.

- **Key Functional Requirements:**

1. **User Authentication and Role Management**
   - The system will provide an admin, lecturer and student login.
   - Users will have a corresponding level of access to each role (view-only, edit, request, feedback)
2. **Lecture Hall Data Management**
   - Admins will be able to add, update and delete hall attributes (capacity, location, status).
   - The system will check the availability and suitability of halls in accordance with the requirements of the event.
3. **Timetable Generation**
   - The system will have the capability for admins to create automated timetables with the use of genetic algorithms.
   - The timetable will reflect availability of halls, module requirements and predicted demand.
4. **Predictive Analytics**
   - The system will use historical information to predict the trends of lecture halls demand.
   - Predict the peak usage periods and allocate them accordingly.

5. **Conflict Detection and Real-Time Adjustments**
   - The system shall automatically detect scheduling conflicts or overallocations.
   - Trigger intelligent agents to suggest reallocation or rescheduling.
6. **Lecture Hall Requests and Issue Reporting**
   - Lecturers shall be able to request halls or report infrastructure issues.
   - Admins receive these requests and resolve or update statuses accordingly.
7. **Notification and Alerts System**
   - The system shall provide automated email or dashboard notification on changes/conflicts/approvals.
8. **Feedback Collection**
   - Students and lecturers shall be able to submit feedback about hall quality, scheduling, or accessibility.
   - Feedback will be stored and analyzed for decision support.
9. **Dashboard and Timetable View**
   - A web UI will be available to administrators that include visual dashboards.
   - Students and Lecturers can see timetables and updates that are personalized to them.

4.6.2 Non-Functional Requirements

Non-functional requirements describe system attributes such as performance, usability, and security that support the system's overall quality and effectiveness.

- **Key Non-Functional Requirements:**

1. **Usability**
   - The interface of the system should be easy and familiar to use for all roles.
   - Must be mobile and desktop browser compatible.
2. **Reliability**
   - The system will be available for 99% of the time in working hours.
   - It will recover from minor faults without losing the data.
3. **Performance**
   - On average input size the timetable generation and conflict detection should happen within 5 seconds.
   - Predictive models must run on historical data within reasonable time frames.
4. **Scalability**
   - The system will support scaling to support more faculties and departments in the future.
5. **Security**
   - All user data and login credentials shall be encrypted.
   - Role-based access control must be implemented to protect sensitive data.
6. **Maintainability**
   - The system should be modular to allow updates and patches without disrupting functionality.

4.7 Conclusion

In this chapter, the system requirements of the AI-based Lecture Hall Management System have been discussed comprehensively for the purpose of meeting the research objectives and requirements of different stakeholders. We talked about functional requirements which concentrate on the basic functions which the system needs to do, like timetabling, predictive analytics and conflict resolution. These capabilities are essential for automating and streamlining lecture hall resource management, delivering a more efficient and data-driven scheduling and resource allocation process.

The non-functional requirements dealt with quality attributes of the system such as performance, scalability, reliability, security, and usability. These assessments ensure that not only are the functional requirements fulfilled but the system also performs optimally in the real-world scenario, while continuing to deliver high standards of service and user satisfaction.

Additionally, the stakeholder analysis identified the users of the system such as administrators, lecturers, and students and their varied needs. The operationalization process was mapped with the goals of the study ensuring that the research objectives are completely reflected in the data collection methods (surveys and historical data analysis).

The system and model analysis section gave an in-depth description of the proposed system including case diagrams, class diagrams, etc. This chapter provides the foundation for the design and development phases by providing clear expectations and guidelines for the system's operation and performance.

In conclusion, this chapter summarizes the requirements needed for the proposed system and sets the direction of the further development and implementation of the Lecture Hall Management System using Artificial Intelligence. These requirements will steer the system design and ensure it fit with the university's operational requirements, setting the stage for future success and adoption.

5 Implementation / Designing

5.1 Introduction

The implementation and design of this study is aimed at the development and creation of the AI-driven Lecture Hall Management System. This chapter defines the process of system development including the algorithm design up to the front-end and back-end implementation. There are also justifications to the technological options selected and a lengthy explanation of the algorithms that are applied in the system.

5.2 Framework Design and Algorithm Development

During the design phase, the main objective was to optimize the lecture hall allocation, foresee the demand, and to automate the process of schedule. The system uses predictive analytics and genetic algorithms to maximize resource assignment and forecast when lecture halls are in the highest demand. The step-by-step framework design is as follows:

1. **System Overview:**
    - The system incorporates the data about the hall capacity, schedules, etc., predictive analytics, genetic algorithms, and real-time changes in the schedules to form an efficient lecture hall management system.
    - The back end is written in Python and Flask, on which all the data processing and decision-making processes will be implemented, as well as data collection, predictive modeling, and optimization of the schedule.
    - The front-end is developed with JavaScript, CSS and HTML. The front-end will be interconnected with the users through displaying timetables, data on the hall allocation, and notifications. It will also be linked to the back end either by using AJAX or fetch calls in updating and showing real-time information.

2. **Planned Algorithm Design:**
    - **Predictive Analytics**: The system will be based on historical data regarding the scheduling results and predict the peak demand of lecture halls. ARIMA or LSTM (Long Short-Term Memory) network-based time-series models will be used to forecast the future demand by using the information about previous usage.
    - **Optimization through Genetic Algorithm**: The genetic algorithm is going to be employed to produce the timetable of the lecture hall which is optimized. The algorithm fitness will reduce the scheduling conflicts including the double booking or capacity overflow. The algorithm will make an approximation of various schedules and will be developed to the solution that will be the most efficient.

- **Steps in Algorithm Design:**

  1. initialization: Produce a starting set of possible schedules.
  2. Selection: Select the most suitable schedules on the basis of their fitness (level of conflict-finesses).
  3. Crossover and Mutation: Combine the most successful schedules to form new ones, adding variation in order to find more successful schedules.
  4. Termination: The algorithm will not terminate until a satisfactory schedule has been discovered, or a specified number of iterations has occurred.

3. **Future Steps**:
   - Once I design the algorithm, I will put the predictive model and genetic algorithm into Python.
   - A back-end system will then relate to the front-end to update and make real time changes.

5.2.1 Genetic Algorithm Pseudocode for Optimizing Lecture Hall Scheduling

```
# Define the Genetic Algorithm for optimizing lecture hall scheduling

# Step 1: Initialize the population with random schedules

def initialize_population(pop_size, num_classes, num_halls):

    population = []

    for _ in range(pop_size):

        schedule = generate_random_schedule(num_classes, num_halls)

        population.append(schedule)

    return population

# Step 2: Evaluate the fitness of a given schedule

def fitness(schedule, conflict_matrix):

    """

    Fitness function that evaluates the quality of a schedule.

    Lower conflict means better fitness.

    """

    conflicts = 0

    for class1 in schedule:

        for class2 in schedule:

            if class1 != class2 and are_conflicting(class1, class2):

                conflicts += 1

    return 1 / (1 + conflicts)  # Inverse of conflicts, higher fitness is better

# Step 3: Selection process (Tournament Selection or Roulette Wheel Selection)

def select_parents(population, fitness_scores):

    # Tournament Selection: Randomly select two individuals and choose the one with the higher fitness

    parents = []

    for _ in range(2):

        tournament = random.sample(population, 3)  # Select 3 individuals randomly

        best_parent = min(tournament, key=lambda s: fitness_scores[tournament.index(s)]) # Select the best parent (lowest conflicts)

        parents.append(best_parent)

    return parents
```

```
# Step 4: Crossover (Recombination)
def crossover(parent1, parent2):
    """
    Perform crossover between two parents to create offspring.
    A single-point crossover is applied here.
    """
    crossover_point = random.randint(1, len(parent1) - 1)
    offspring = parent1[:crossover_point] + parent2[crossover_point:]
    return offspring
# Step 5: Mutation
def mutate(schedule, mutation_rate):
    """
    Perform mutation by randomly changing a small part of the schedule.
    The mutation rate determines how often this occurs.
    """
    for i in range(len(schedule)):
        if random.random() < mutation_rate:
            schedule[i] = mutate_class_schedule(schedule[i])  # Mutate individual class assignments
    return schedule
# Step 6: Main Genetic Algorithm Loop
def genetic_algorithm(pop_size, num_generations, num_classes, num_halls, mutation_rate, conflict_matrix):
    # Initialize population
    population = initialize_population(pop_size, num_classes, num_halls)
    for generation in range(num_generations):
        # Step 6.1: Calculate fitness for each individual in the population
        fitness_scores = [fitness(schedule, conflict_matrix) for schedule in population]
        # Step 6.2: Select parents based on fitness
        new_population = []
        for _ in range(pop_size // 2):
            parent1, parent2 = select_parents(population, fitness_scores)
            offspring1 = crossover(parent1, parent2)
            offspring2 = crossover(parent2, parent1)
```

```
# Step 6.3: Mutate offspring
            offspring1 = mutate(offspring1, mutation_rate)
            offspring2 = mutate(offspring2, mutation_rate)

            new_population.extend([offspring1, offspring2])

        # Step 6.4: Replace old population with new population
        population = new_population

        # Optionally: Print or save the best solution found in each generation
        best_schedule = min(population, key=lambda s: fitness(s, conflict_matrix))
        print(f"Generation {generation + 1}: Best Fitness: {fitness(best_schedule, conflict_matrix)}")

    return best_schedule

# Example of usage
pop_size = 100  # Population size
num_generations = 500  # Number of generations to evolve
num_classes = 10  # Number of classes to schedule
num_halls = 5  # Number of available halls
mutation_rate = 0.05  # Mutation rate
conflict_matrix = []  # A matrix of class conflicts (to be defined)

best_schedule = genetic_algorithm(pop_size, num_generations, num_classes, num_halls, mutation_rate, conflict_matrix)
```

- **Explanation of the Genetic Algorithm Steps:**
  1. **Initialized Population:** A population of random schedules is created. Each individual in the population represents a possible schedule with assigned classes to available halls.
  2. **Fitness Evaluation:**
     - A fitness function is defined to measure how good a schedule is. In this case, the fitness function calculates the number of conflicts (double bookings, overlapping classes) in the timetable and returns the inverse of conflicts as the fitness score. The fewer conflicts, the higher the fitness.
  3. **Selection**:
     - Tournament Selection (or another method like Roulette Wheel) is used to choose parents for the next generation. The individuals with the best fitness (fewer conflicts) are selected for crossover.
  4. **Crossover**:
     - The crossover step mixes the genetic material (schedules) of two parents to create offspring. This typically happens at a random point in the schedule (single-point crossover).
  5. **Mutation**:
     - Mutation introduces small changes in the offspring's schedule to maintain diversity. This can happen by randomly changing class assignments (e.g., changing a class's hall assignment).
  6. **New Generation:**
     - After crossover and mutation, the new generation of schedules is created. The best schedules are retained, and the process repeats for a set number of generations.

Parameters:

- Population Size: Number of individuals (schedules) in each generation.
- Generations: Number of iterations the algorithm will run to evolve the best schedule.
- Mutation Rate: Probability that a mutation will occur in a schedule.

5.2.3 Justify the Selection of Technologies

To develop and implement the Lecture Hall Management System successfully, it is important to include the selection of suitable technologies. The table below justifies the selected technologies:

5.2.3.1 Programming Language

Technology	Justification
Python	Python was selected as the main programming language of this system because of its simplicity, readability, and wide data manipulation and machine learning support. Python is a common data science and AI language, which is necessary when it comes to the Predictive Analytics and Genetic Algorithm aspects of the system. The ability to use Python with libraries such as scikit-learn, TensorFlow, and Pandas, makes it the best option when it comes to development of a backend, as well as AI/ML. Also, Python rich ecosystem and a large community facilitate the development and troubleshooting.

Table 6: Programming Language

5.2.3.2 Plan to use Libraries

Library/Tool	Justification
Flask	flask is a light, bendable Python web structure. It was selected because of its simplicity and minimalism which is perfect in developing the undercarriage of the system. With Flask we can easily combine machine learning models, deal with APIs and process data with low overhead. It is especially appropriate in small- to medium-sized applications such as this system, and it is very well-supported in the construction of RESTful APIs, which the front-end can communicate with.
scikit-learn	scikit-learn is a Python library to build machine learning models. It is also necessary to develop the Predictive Analytics element of the system. The library offers effective methods of data mining, predictive modeling, and classification including linear regression, decision trees, and ensemble methods. We shall apply it in order to foresee the demand trends and schedule lecture halls effectively.
Pandas	Pandas is an efficient data manipulation Python Library. It was chosen to process large datasets (use of lecture halls in history and class data). Tasks such as cleaning data, transformation of data and aggregation of the data to be utilized by the predictive models will be completed using panda.
NumPy	NumPy is a numerical library that provides mathematical operations and manipulation of large, multi-dimensional arrays and matrices. Efficient mathematical operations are required, particularly when predictive models and optimization algorithms are being applied. NumPy will be employed together with Pandas to manipulate and process big datasets efficiently in computational terms.
Matplotlib/ Plotly	Data visualization is through matplotlib and Plotly. Static visualizations will be created with Matplotlib, and interactive charts can be created with Plotly. These libraries will assist in delivering outputs of systems, say their demand forecast, schedule optimization output, and hall utilization information, to administrators in a visual format that will be comprehended.

Table 7: Libraries Files

5.2.3.3 Planned Backend and Front-end Frameworks

Technology	Justification
Backend: Flask	The choice of Flask has been driven by its flexibility, minimalism and simplicity of usage. It is best suited to the creation of RESTful APIs which could be used as the communication layer in between the backend (where the data processing and algorithms are executed) and the front-end (where the user interactions can be made). Machine learning models and libraries, including scikit-learn and deep, are also available in Flask and are necessary to generate predictive analytics and optimization in the scheduling system. It allows one to develop fast and offers scalability in future improvements.
Frontend: HTML5/CSS3/JavaScript	The front-end is constructed in HTML5, CSS3 and JavaScript. These technologies have become a choice since they are the basis of modern web applications, and it provides maximum usability of all browsers. The content is structured by HTML5, the modern and responsive design is achieved by CSS3 and the interactivity, form validation and dynamic updates of data are managed by JavaScript without having to refresh the page. These technologies allow the front-end to be lightweight, responsive and not hard to implement.
Frontend Libraries	A responsive and mobile-friendly design is adopted with the help of bootstrap or another CSS framework. They also enable quicker development of UI, since such frameworks include ready-made classes of CSS styles to use (e.g., layouts, grids, and components). This is aimed at making the application simple to navigate and useable by all the stakeholders (admins, lecturers and students).

Table 8: Backend and Front-end Frameworks

5.3 Significant Implementation Attempts

The following section will discuss the essence of the AI-driven Lecture Hall Management System, including the implementation specifics of the key elements and the UI characteristics of them. We are also going to provide examples of code snippets and UI designs that show how these features are implemented.

6.1.1 Predictive Analytics for Hall Demand Forecasting

- **Feature Description**: The system uses **predictive analytics** to forecast the demand for lecture halls, helping administrators plan for high-demand periods. The **predictive model** analyzes past data to generate **future demand predictions**.

- **UI Design**:

Figure 8: Admin Dashboard displaying predicted lecture hall demand

- **Code Snippet**:

```
# ml_service/app.py
from flask import Flask, jsonify, request
from flask_cors import CORS
import pymysql
# --- Flask app ---
app = Flask(__name__)
CORS(app)  # optional, useful during dev
# --- DB helper ---
def get_conn():
    return pymysql.connect(
        host="localhost",
        user="root",
        password="",
        database="nsbm",
        cursorclass=pymysql.cursors.DictCursor,
        autocommit=True,
    )
# --- Utils ---
DAYS = ["Monday","Tuesday","Wednesday","Thursday","Friday"]

def slot_key(slot: str) -> int:
    """Sort time ranges like '10:00-12:00' by start time."""
    try:
        start = slot.split("-")[0]
        h, m = start.split(":")
        return int(h) * 60 + int(m)
    except Exception:
        return 0
```

```
# ---------- Routes ----------
@app.get("/health")
def health():
    return jsonify(ok=True)
@app.post("/train")
def train():
    # placeholder "training" (you can wire real model later)
    # example: count unique (day,time_slot) pairs
    with get_conn() as conn, conn.cursor() as cur:
            cur.execute("SELECT COUNT(*) AS c FROM (SELECT day, time_slot FROM bookings GROUP BY day, time_slot) t;")
        row = cur.fetchone()
    return jsonify(ok=True, keys=row["c"])
@app.get("/forecast")
def forecast():
    """
    Baseline forecast = historical frequency of bookings per (day, time_slot).
    Returns:
      {
        "ok": true,
        "days": ["Monday",...],
        "slots": ["08:00-09:00", ...]  # all distinct slots sorted by start time
            "score":    [{"day":"Tuesday","time":"10:00-12:00","demand":12}, ...],
            "top":      [{"day":"Tuesday","time":"10:00-12:00","demand":12}, ...]  # top 3
      }
```

6.1.2 Admin Dashboard for Timetable Management

- **Feature Description**: Admin Dashboard gives a vision of the lecture hall condition with the presence of a visual schedule and the ability to edit it in case of necessity. This plays a very important role in solving any conflict or dealing with last-minute adjustments.

- **UI Design**:

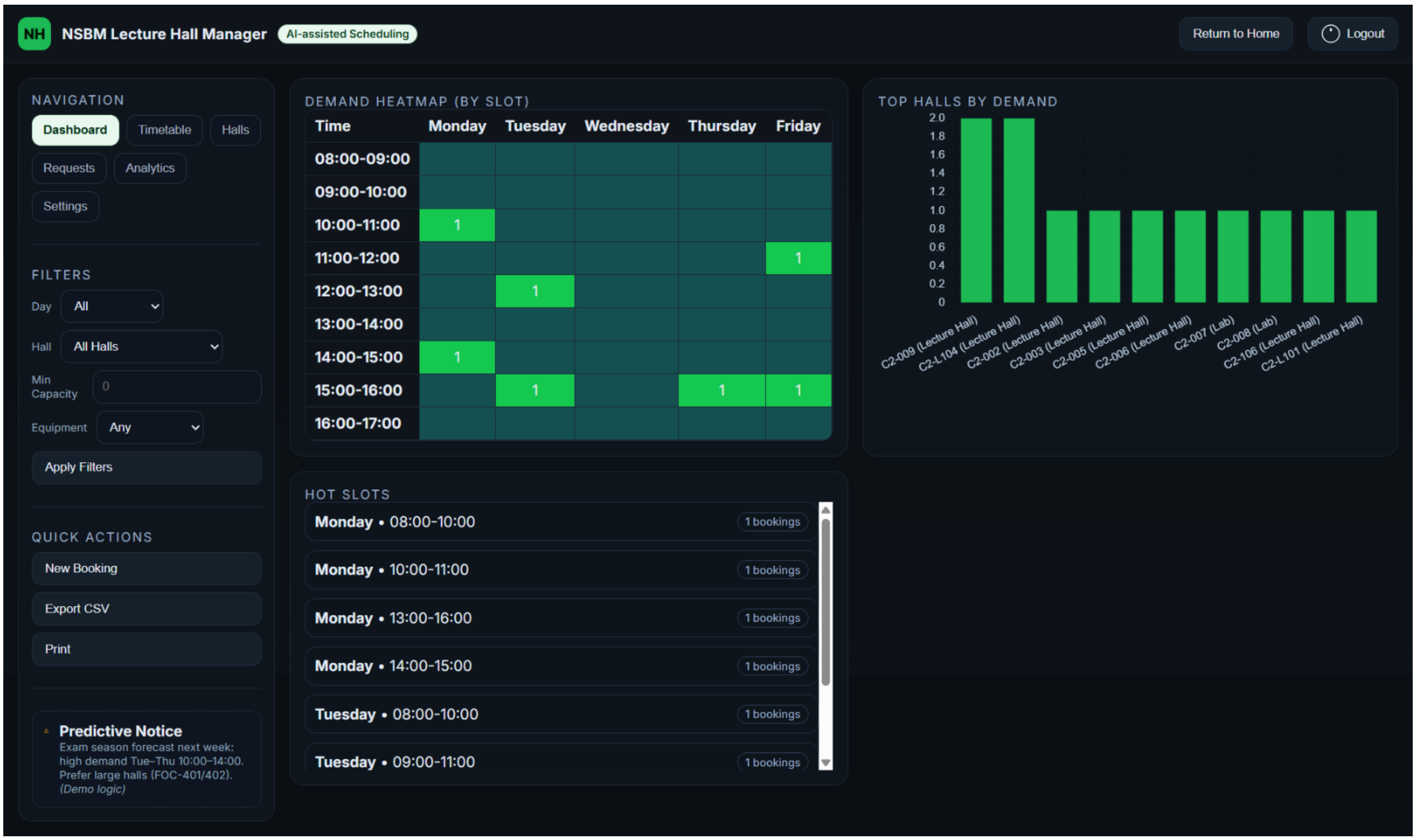


Figure 9: Admin Dashboard for Timetable Management - Predictive Analytics Visualization

5.4 Chapter Summary

The chapter is an Implementation chapter that gives a brief description of the implementation and design process of the AI-powered Lecture Hall Management System. The chapter provides the systematic manner of developing and introducing the main aspects of the system that is aimed at the optimization of the lecture hall scheduling and reduction of conflicts and predicting demand based on sophisticated algorithms.

The chapter starts with a comprehensive description of the structure design because Python and Flask have been selected to be the back-end based on their flexibility, scalability, and the ability to seamlessly integrate with machine learning libraries. The frontend is created with HTML, CSS and JavaScript to have a responsive and friendly interface to achieve interaction with the system. The reasons behind the choice of technologies are presented, focusing on their appropriateness to develop a strong and efficient system.

Subsequently, this chapter proceeds to the design of the algorithm with emphasis on the Predictive Analytics, and the Genetic Algorithm of optimizing the lecture hall schedules and predicting future demand. Although the algorithm design is in the planning stage, the chapter details the planned course of action on the implementation, including the utilization of ARIMA to make a prediction of demand and the utilization of a Genetic Algorithm that will help produce an optimal schedule.

An explanation of the UI design, specifically an Admin Dashboard, showing important data, e.g., predicted hall demand and optimized schedules, are also presented within the chapter. The code snippets and screenshots are given to illustrate the application of the main features, such as the model of predictive control over the demand forecasting and the genetic algorithm of the schedule optimization.

Finally, this chapter is a more extensive analysis of the design and execution of the AI-based Lecture Hall Management System. It addresses the technology options, the algorithms used, the UI design, and the general workflow of the system. In the chapter, it is emphasized that the system allows us to use the lecture halls in the most efficient way, forecast the further demand, and update the user in real-time, which contributes to the increased efficiency of the academic scheduling process overall.

6 Testing and Evaluation

6.1 Chapter Overview

This chapter gives the testing and evaluation procedure of AI-powered Lecture Hall Management System that was created during this research. The purpose is to establish that the system establishes the required functional and nonfunctional requirements and offers a high-quality, efficient and reliable lecture hall scheduling solution. The chapter also entails identifying the test plan, test cases as well as the test strategies that were used to test the performance of the system and the outcome that was obtained after the two forms of testing.

Testing is very important in this project since it assists in accuracy, efficiency, scalability and ease of use of the system. A set of both functional and nonfunctional testing methods have been employed to determine the performance of the system under different conditions and its ability to address the real-life academic scheduling situations. The last part of the chapter is a review of testing strategies employed and their correspondence to the aims of this study.

Test Plan and Test Cases

The test plan will help confirm that the AI powered Lecture Hall Management System does what it is supposed to do, to guarantee its functionality and in support of the stated requirements. The system incorporates predictive analytics, genetic algorithms and real time flexibility in scheduling lecture halls. The major features of the system and test cases to assess the system are given below.

6.1.3 Nonfunctional Testing

Nonfunctional testing focuses on the system's performance, scalability, security, and usability.

Test Case ID	Test Objective	Test Type	Test Description	Expected Outcome
NF01	System Performance (Speed & Latency)	Performance Testing	Test system response time when accessing large datasets and generating schedules.	Response time < 3 seconds for any query.
NF02	Scalability under Load	Load Testing	Simulate a large number of concurrent users (e.g., 1000+ simultaneous users) interacting with the system.	No performance degradation, system should scale seamlessly.
NF03	Usability Testing	Usability Testing	Evaluate the UI's ease of use for administrators and students, including navigation, scheduling, and hall booking.	90% user satisfaction with UI.
NF04	System Security (Data Protection)	Security Testing	Test against SQL injection and XSS vulnerabilities in user input fields.	No security breaches, and the system should reject harmful input.

6.1.4 Functional Testing

Functional testing will ensure the core features of the system operate correctly, such as lecture hall allocation, scheduling, and notification systems.

Test Case ID	Test Objective	Test Type	Test Description	Expected Outcome
F01	Login and User Authentication	Functional Testing	Test the login functionality with valid and invalid credentials for administrators, lecturers, and students.	Successful login for valid credentials; error message for invalid credentials.
F02	Lecture Hall Allocation	Functional Testing	Ensure lecture halls are allocated to classes based on constraints (size, equipment, time).	Correct hall allocation with no conflicts.
F03	Schedule Generation	Functional Testing	Test the Genetic Algorithm for generating an optimized schedule for the upcoming semester.	Conflict-free, optimized timetable generated.
F04	Real-Time Notification System	Functional Testing	Test the notification system by making a change to the lecture schedule and checking notifications received by users.	Notifications sent to the correct users with accurate details.
F05	Feedback System	Functional Testing	Ensure that feedback submitted by users (lecturers/students) is properly recorded and displayed in the admin panel.	Feedback is successfully stored and displayed.

Testing / Evaluation Workflow

1. **Test Preparation**:
   - Define the scope and objectives for both functional and nonfunctional testing.
   - Set up test environments, including simulated user interactions and datasets from the university.
   - Develop test scripts for key features such as hall allocation, schedule generation, and real-time notifications.
2. **Test Execution**:
   - Execute each test case one by one, recording the results. Nonfunctional tests like load and performance tests will be conducted first, followed by functional tests.
   - For each test, compare the actual results with the expected outcomes and identify any discrepancies.
3. **Evaluation**:
   - Analyze the results from each test. Assess the performance and usability of the system based on the testing outcomes.
   - Identify bugs or issues and categorize them by severity. Minor issues may be addressed in future versions, while critical bugs will be fixed immediately.
4. **Reporting**:
   - Summarize the test results, highlighting key findings and the performance of the system.
   - Include any charts or graphs to illustrate system performance, such as response times and load testing outcomes.
   - Provide a clear review of the testing strategy used, showing how it aligns with the system's objectives.

Review of Test Strategies Used

The testing strategies employed in this research were pivotal in ensuring that the **AI-powered Lecture Hall Management System** meets both functional and nonfunctional requirements. The primary goal of testing was to validate that the system operates as expected and delivers the intended outcomes. These strategies were designed to assess the system's performance under real-world conditions, ensuring its robustness, scalability, security, and usability.

Functional testing was the first critical strategy used to ensure the system performs its core tasks correctly. It focused on verifying the primary features of the system, such as **lecture hall allocation**, **schedule generation**, and **real-time notifications**. Functional testing ensures that the system can allocate halls to classes, generate optimized timetables, and notify users about changes. The testing process involved checking that the system performs these functions according to the specified requirements, using real-world data. The expected result was that all core features should be completed without errors, ensuring the system's robustness under normal usage conditions.

In addition to **functional testing**, **nonfunctional testing** was used to evaluate key aspects of the system that are not directly related to functionality but are crucial for performance, scalability, security, and user experience. **Performance testing** assessed the system's ability to handle large datasets and simultaneous user interactions. It ensured that the system could generate schedules and respond to queries quickly, even under heavy load. The goal was to ensure that the system's response time remains within acceptable limits, particularly when handling multiple concurrent users and large volumes of scheduling data.

Scalability testing was another critical aspect of nonfunctional testing. This testing simulated an increasing number of users and data records to verify that the system could scale effectively as the number of users and lecture halls grows. It helped ensure that the system can accommodate the needs of larger institutions in the future without degrading performance. Similarly, **security testing** focused on safeguarding the system against potential vulnerabilities. Security tests, such as SQL injection and cross-site scripting (XSS) attacks, were performed to ensure that the system prevents unauthorized access and protects user data from malicious threats.

Usability testing was also conducted to assess the user interface (UI) of the system. This testing aimed to ensure that the system is easy to navigate and user-friendly for administrators, lecturers, and students. Feedback from users was collected to identify any areas of confusion or frustration. The testing ensured that the system's design and flow meet the expectations of its intended users and that the UI supports the efficient completion of tasks, such as hall allocation and timetable viewing.

Several tools were used to automate and streamline the testing process, improving efficiency and ensuring comprehensive coverage. **Selenium**, an open-source tool, was utilized for **UI testing** to verify that the system's user interface works as expected under different conditions. Selenium allowed the automation of user actions, such as logging in, navigating the dashboard, and interacting with the scheduling features. This helped validate the **functional behavior** of the system's UI and ensured consistent performance across browsers.

JMeter was used for **load testing** and **performance testing**. This tool simulated multiple users accessing the system concurrently to assess how the system performs under stress. JMeter helped identify potential **bottlenecks** in the system, allowing for adjustments to improve its performance when handling large user loads. This tool was particularly useful in testing the **scalability** of the system to ensure it can support future growth.

The combination of **manual testing** and **automated testing tools** allowed for thorough validation of the system's capabilities. These testing strategies were selected to provide a holistic evaluation of the system's functionality, performance, and security. The tools and techniques used were specifically chosen to align with the goals of the **AI-powered Lecture Hall Management System**, ensuring that it can handle large datasets, perform under heavy load, and provide a secure, user-friendly experience.

In conclusion, the **testing strategies** employed were comprehensive and well-suited to the research objectives. They ensured that the **AI-powered Lecture Hall Management System** meets both **functional correctness** and **performance requirements**, allowing for efficient and scalable management of lecture halls. These strategies not only addressed the system's primary functionality but also ensured that the system could operate effectively in a real-world, high-demand academic environment. The results from these tests validate the system's potential for real-world application, offering insights into how it can be improved and scaled for larger institutions in the future.

7 Concluding Remarks

Accomplishment of the Research Objectives

The primary objective of this research was to develop an AI-powered Lecture Hall Management System capable of optimizing lecture hall scheduling using genetic algorithms and predictive analytics. The system was designed to predict demand for lecture halls and allocate resources efficiently, minimizing conflicts and improving resource utilization.

To assess the success of the research, triangulation was used as a strategy, incorporating data from different sources and testing methods. The functional and nonfunctional testing outcomes derived from the previous chapter provided significant evidence that the system meets its objectives. The predictive models were shown to improve the accuracy of hall allocation, and the genetic algorithm successfully generated conflict-free timetables. Additionally, the system was capable of handling large datasets and scaling underload, demonstrating its effectiveness in real-world applications.

In terms of accomplishment, the research achieved its goals by creating a robust and scalable system that can be adapted to various university environments, enhancing the efficiency of scheduling processes.

Problems Encountered

Throughout the development and testing phases of the research, several challenges were encountered:

Data Quality and Availability: One of the primary challenges was obtaining comprehensive, accurate historical scheduling data for training the predictive models. Incomplete or inconsistent data from existing systems created difficulties in ensuring the reliability of the predictive analytics models.

Integration of Machine Learning Models: Integrating machine learning algorithms (e.g., ARIMA for forecasting) with the existing scheduling framework posed some technical hurdles, especially when ensuring that the outputs of the predictive model could be seamlessly incorporated into the scheduling decisions made by the genetic algorithm.

User Interface Challenges: Designing a user-friendly interface for administrators, lecturers, and students was another challenge. Balancing functionality with simplicity required iterative feedback and adjustments during the usability testing phase to ensure that the system would meet the needs of a diverse user base.

System Performance under High Load: While performance testing showed the system could handle moderate loads, simulating extreme scenarios with large datasets and many concurrent users highlighted potential bottlenecks in the real-time processing of scheduling data.

Self-reflection

7.3.1 Your Ideology About Research Carried Out

This research was an opportunity to explore how artificial intelligence and machine learning can improve the efficiency of administrative tasks in educational institutions. My ideology going into this project was centered around the belief that AI-powered solutions can significantly optimize traditional processes, making them more efficient and scalable. Specifically, I was keen to see how genetic algorithms could be used in real-world applications to solve problems like lecture hall scheduling, which, though seemingly mundane, often involves complex constraints that require intelligent solutions.

Through the development of the Lecture Hall Management System, I realized how AI and optimization techniques could bridge the gap between complex scheduling requirements and real-world resource management. This experience reinforced my belief in the power of technology to solve practical problems in education and other sectors.

7.1.1 Benefits Gained

One of the greatest benefits gained from this research was the deep understanding of how machine learning and AI algorithms work in practice. I not only gained theoretical knowledge about predictive analytics and genetic algorithms but also learned how to apply these concepts to real-world problems.

The testing process, in particular, helped me refine my understanding of system performance, scalability, and the importance of user-centered design. I gained practical experience with data preparation, algorithm implementation, and system testing. The project also improved my ability to work with tools like Python, Flask, JMeter, and Selenium in real-world scenarios.

7.1.2 Learning Curves

The research presented several learning curves, especially in the areas of machine learning integration, system architecture, and testing methodologies:

Machine Learning Integration: Initially, integrating machine learning models into the scheduling system was challenging. I had to learn how to preprocess data correctly, select appropriate models, and fine-tune algorithms to generate meaningful results. The process was time-consuming but ultimately helped me develop a better understanding of predictive modeling and data science techniques.

System Architecture: Designing the system architecture was another steep learning curve. Deciding how to structure the system for scalability, performance, and user interaction required a deep dive into system design principles and architectural patterns.

Testing and Evaluation: Finally, the testing and evaluation process was a major learning experience. Testing the system under different conditions, collecting user feedback, and using testing tools like JMeter and Selenium taught me how to ensure that the system is reliable and robust.

Business Insight of Your Idea

7.1.3 Real World Application Possibilities of Your Concept / Idea

The AI-powered Lecture Hall Management System has vast real-world applications beyond academic institutions. The core concepts of this system—AI-driven scheduling, resource optimization, and predictive analytics—can be adapted to a variety of domains. For instance:

1. Corporate Training: Large corporations can use a similar system to schedule and allocate training sessions, workshops, and meetings across multiple rooms or locations.
2. Event Management: The system can be adapted to optimize event space allocation, ensuring that resources such as conference halls or meeting rooms are used effectively and efficiently.
3. Hospital Resource Management: Hospitals could utilize similar AI techniques to optimize the allocation of operating rooms, patient scheduling, and medical staff assignments, ensuring maximum utilization of available resources.
4. Conference Scheduling: Large-scale conferences or symposiums can leverage this system to automatically generate session schedules, assign rooms, and optimize time slots based on participant preferences.

Overall, the AI-powered scheduling system represents a scalable and adaptable solution for managing resources in any environment that requires careful planning, allocation, and optimization of limited resources.

Future Recommendations

As with any system, there are areas for improvement and future enhancements. Based on the challenges and limitations encountered during the research, the following recommendations are proposed for future work:

1. Integration with Real-Time Data: Future versions of the system could integrate with real-time data sources, such as university databases or API endpoints, to automatically update schedules based on changes like room maintenance or lecturer availability.
2. Improved User Interface: While the current system is functional, a more interactive and dynamic UI would enhance the user experience. Adding features like drag-and-drop scheduling and real-time conflict resolution would improve usability.
3. Advanced Predictive Models: Exploring more advanced machine learning techniques, such as deep learning, could further improve the accuracy of demand predictions and scheduling optimization.
4. Cross-Institution Collaboration: For larger scale applications, a network of institutions could collaborate to share data and improve scheduling across universities or campuses, potentially creating a nationwide scheduling optimization system.
5. Mobile Application Development: A mobile app could be developed to allow students, lecturers, and administrators to access schedules, request changes, and receive real-time notifications on their mobile devices.

REFERENCES

[1] M. A. Aslam, F. Murtaza, M. Ehatisham, U. Haq, A. Yasin, and N. Ali, “SAPEx-D : A Comprehensive Dataset for Predictive Analytics in Personalized Education Using Machine Learning,” pp. 1–29, 2025.

[2] V. Maphosa and M. Maphosa, “Artificial intelligence in higher education: a bibliometric analysis and topic modeling approach,” *Appl. Artif. Intell.*, vol. 37, no. 1, 2023, doi: 10.1080/08839514.2023.2261730.

[3] A. Diwakar, S. K. Upadhyay, G. Kumar, and G. Singh, “EFFECT OF COVID-19 ON SCHOOL EDUCATION SYSTEM : CHALLENGES AND EFFECT OF COVID-19 ON SCHOOL EDUCATION SYSTEM : CHALLENGES AND OPPORTUNITIES TO ADOPT ONLINE TEACHING,” no. December, 2020, doi: 10.18510/hssr.2020.862.

[4] “Press Release page-1| MarketsandMarkets.” Accessed: Aug. 31, 2025. [Online]. Available: https://www.marketsandmarkets.com/press-release-2.html

[5] T. H. Gajanayake and P. J. Pemarathne, “Automate Timetable Scheduling with AI: A Review”.

[6] B. Paechter, R. C. Rankin, and A. Cumming, “Improving a lecture timetabling system for university-wide use,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 1408, no. January, pp. 156–165, 1998, doi: 10.1007/bfb0055887.

[7] S. U. Wp, “Academic Freedom Movement in Southeast Asian : Threat , Challenge , and It ’ s Implication in Indonesia,” no. November, 2024.

[8] A. T. Ernst, H. Jiang, M. Krishnamoorthy, and D. Sier, “Staff scheduling and rostering: A review of applications, methods and models,” *Eur. J. Oper. Res.*, vol. 153, no. 1, pp. 3–27, 2004, doi: 10.1016/S0377-2217(03)00095-X.

[9] “AN EFFICIENT DECISION SUPPORT SYSTEM FOR ACADEMIC COURSE SCHEDULING.pdf.”

[10] D. Parchment and S. Sankaranarayanan, “Intelligent agent based student-staff scheduling system,” *Int. J. Comput. Inf. Syst. Ind. Manag. Appl.*, vol. 5, pp. 383–404, 2013.

[11] T. Doleck, D. J. Lemay, R. B. Basnet, and P. Bazelais, “Predictive analytics in education: a comparison of deep learning frameworks,” *Educ. Inf. Technol.*, vol. 25, no. 3, pp. 1951–1963, 2020, doi: 10.1007/s10639-019-10068-4.

[12] R.-W. Bello and J. A. Godwill, “Hybrid Technologies and Genetic Algorithms Applied to School Lecture Timetable Generation,” *SSRN Electron. J.*, 2024, doi: 10.2139/ssrn.4691286.

[13] “Pr rin t n ot pe er re v Pr ep rin er ed”.

[14] P. Dasgupta, “Adaptive Decision Support for Academic Course Scheduling Using Intelligent Software Agents,” *Int. J. Technol. Teach. Learn.*, vol. 1, no. 2, pp. 63–78, 2005.

[15] J. Soyemi, J. Akinode, and S. Oloruntoba, “ISSN : 2249-0868 Foundation of Computer

Science FCS," *Int. J. Appl. Inf. Syst.*, vol. 12, no. 5, pp. 20–27, 2017, [Online]. Available: www.ijais.org

[16] J. Soyemi, J. Akinode, and S. Oloruntoba, "Electronic lecture time-table scheduler using genetic algorithm," *Proc. - 2017 IEEE 15th Int. Conf. Dependable, Auton. Secur. Comput. 2017 IEEE 15th Int. Conf. Pervasive Intell. Comput. 2017 IEEE 3rd Int. Conf. Big Data Intell. Compu*, vol. 2018-Janua, pp. 710–715, 2017, doi: 10.1109/DASC-PICom-DataCom-CyberSciTec.2017.124.

APPENDICES

Appendix B: System User Interface Screenshots

Figure 10: Sign Up Interface

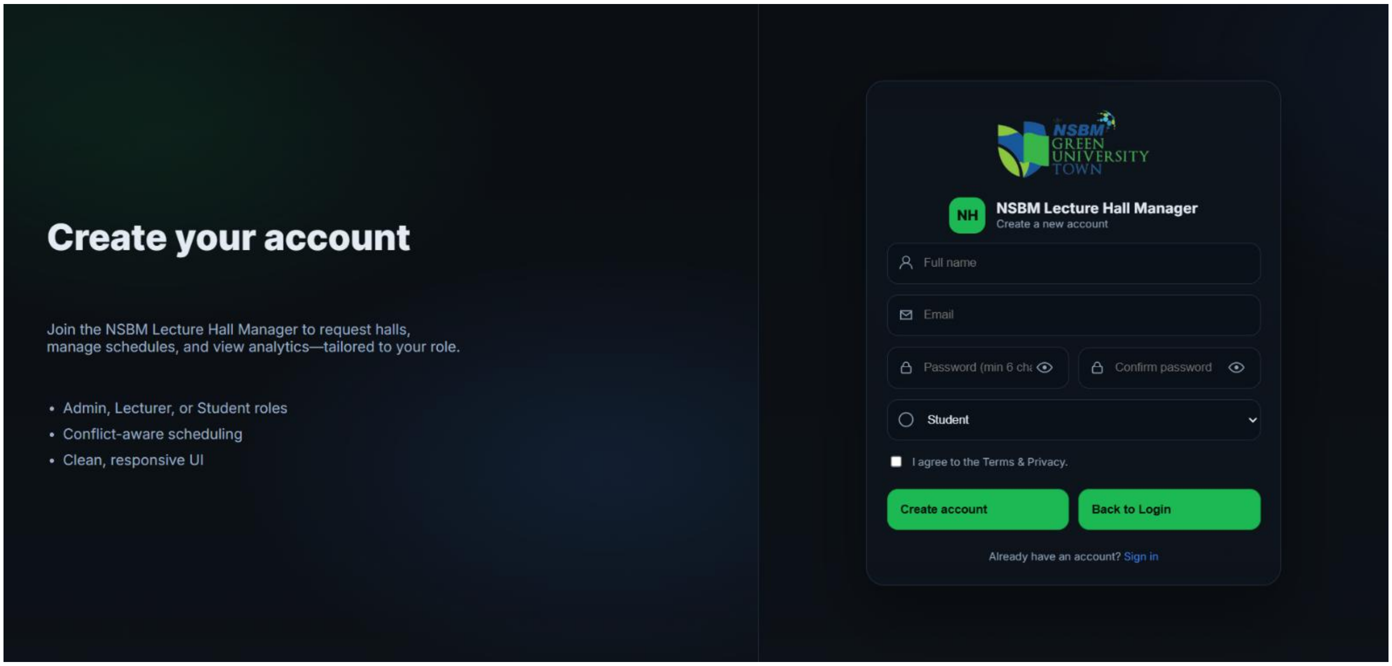


Figure 11: Sign In Interface

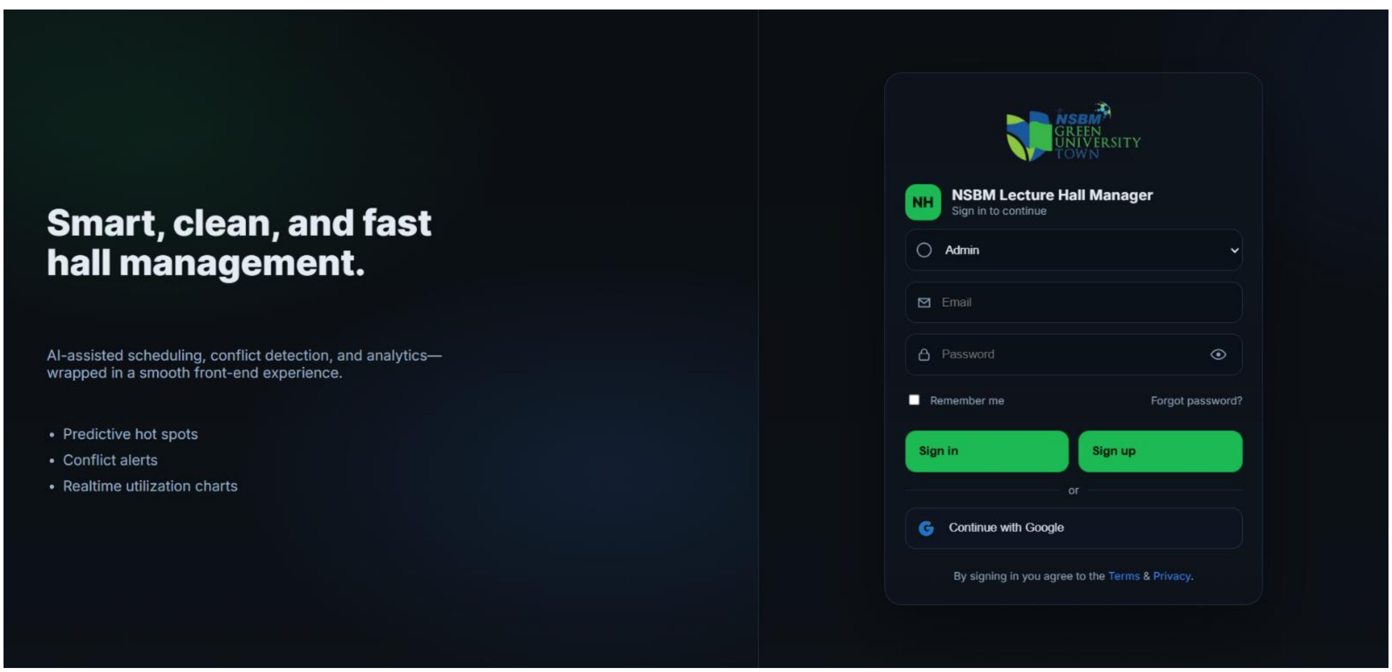


Figure 12: Timetable Generation Interface

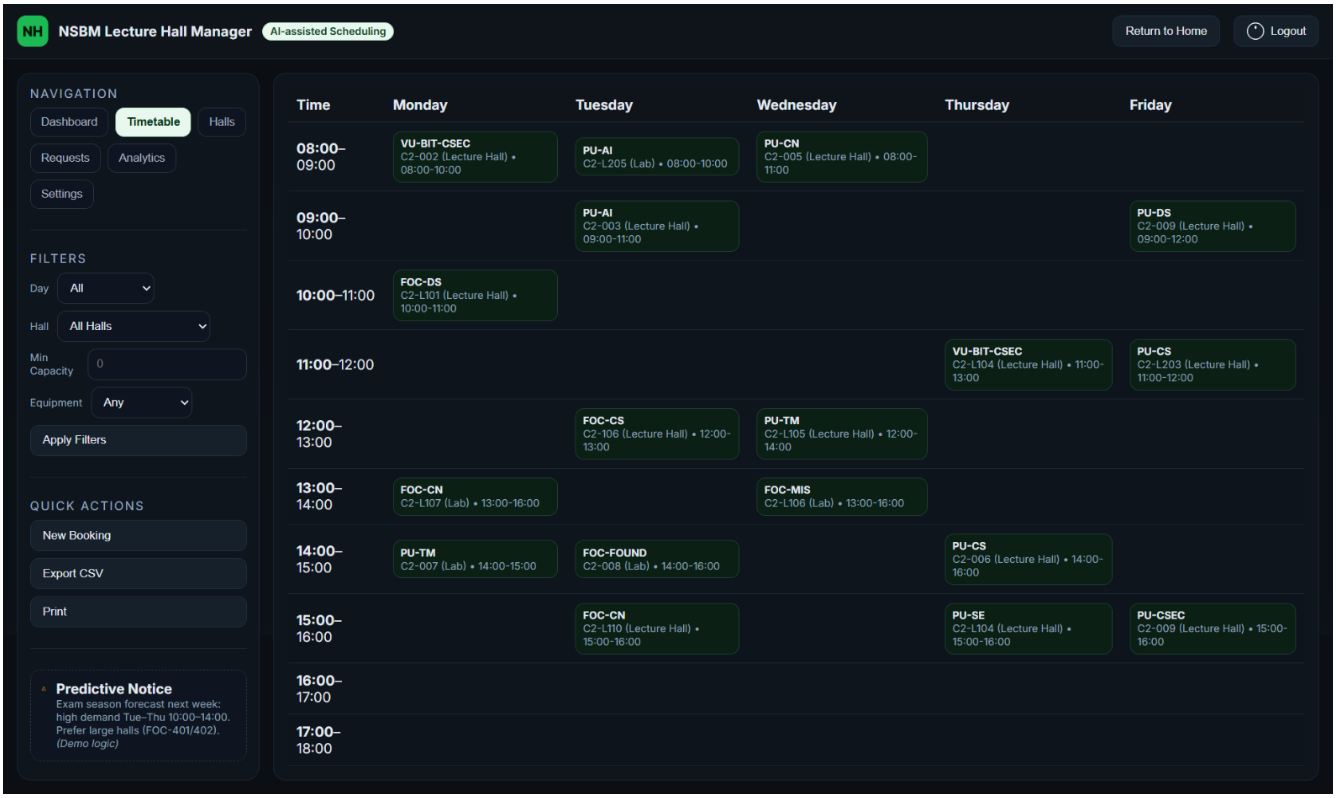


Figure 13: Lecture Hall Allocation Interface (Request)

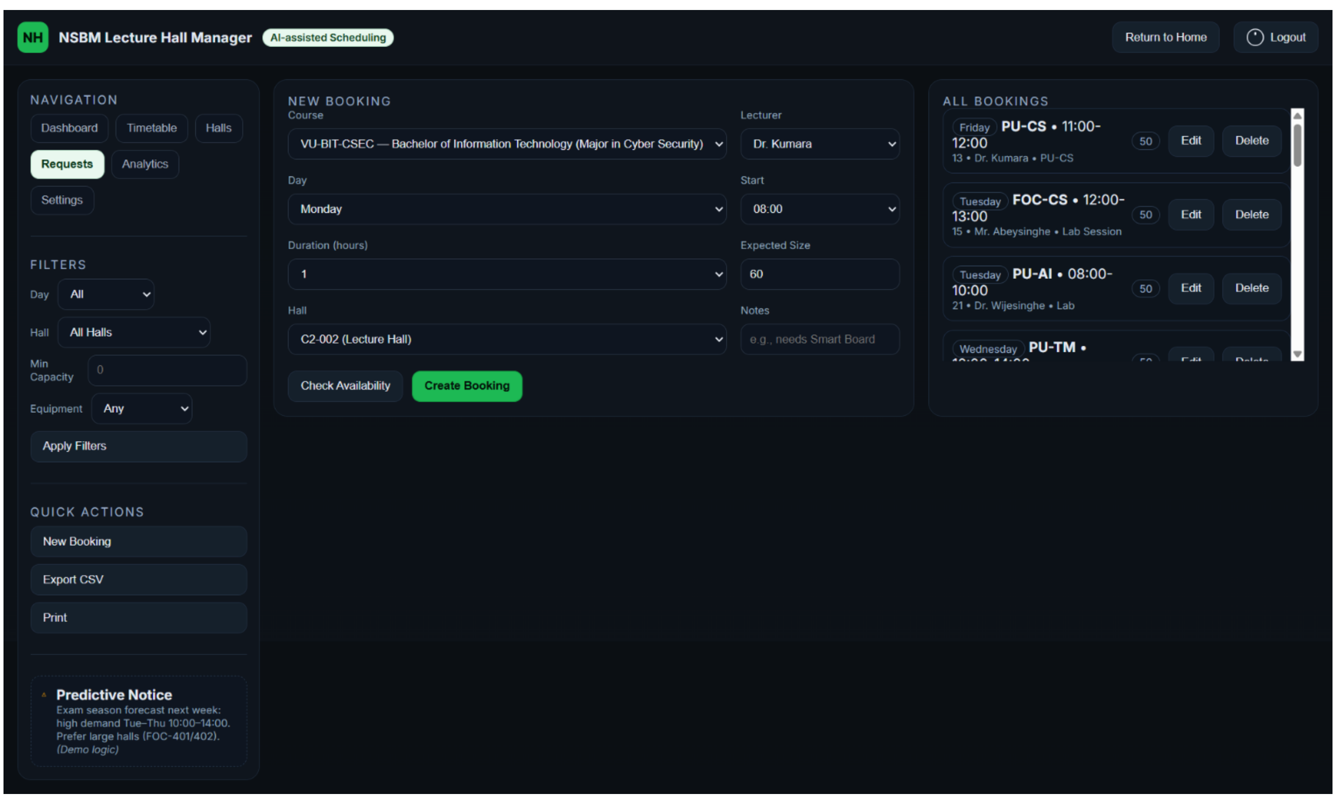


Figure 14: Admin Setting Interface

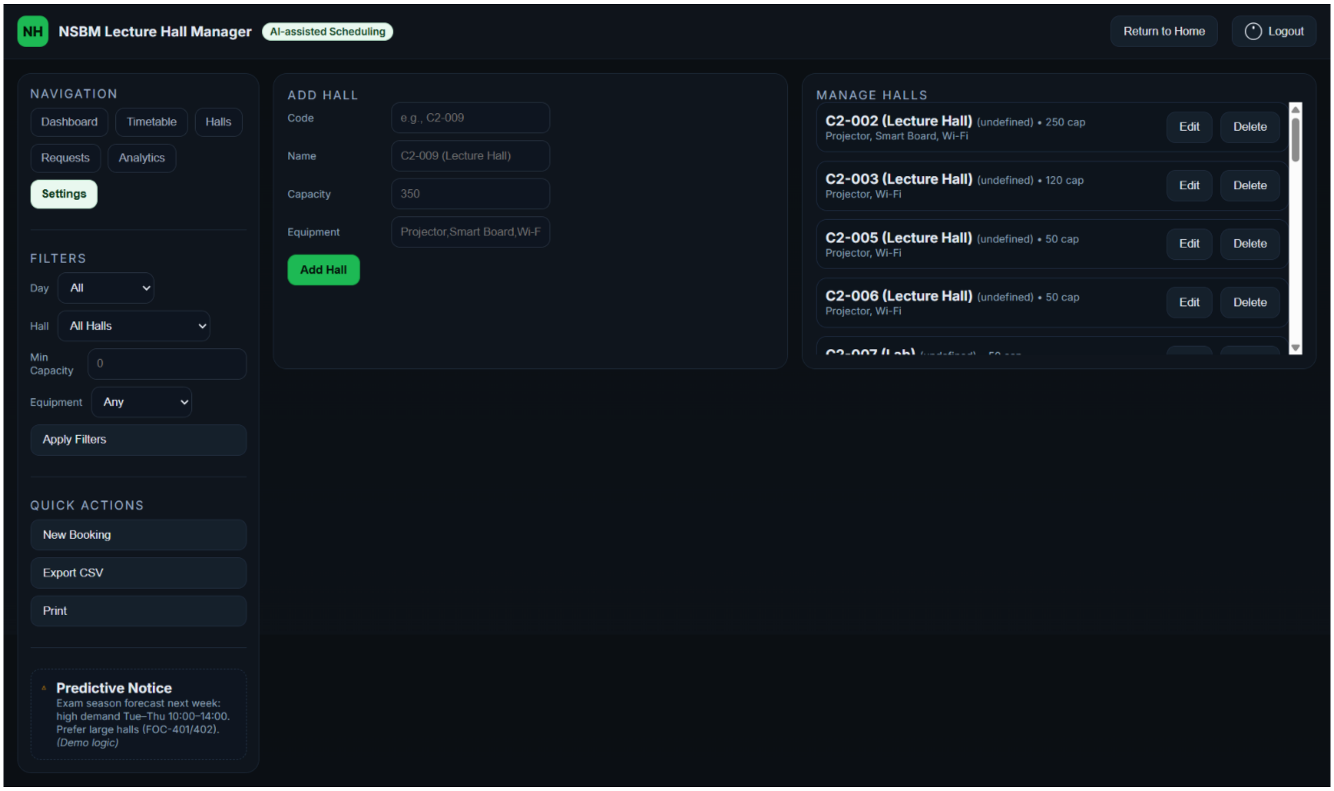


Figure 15: VS Code File Structure

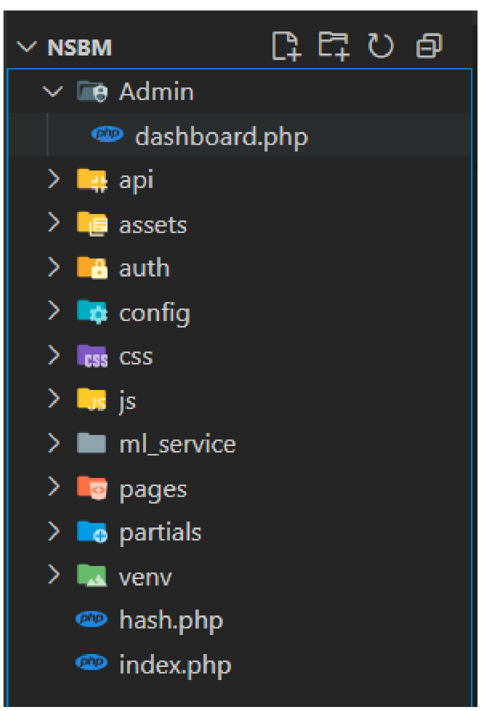


Figure 16: Lecture Hall Details Interface

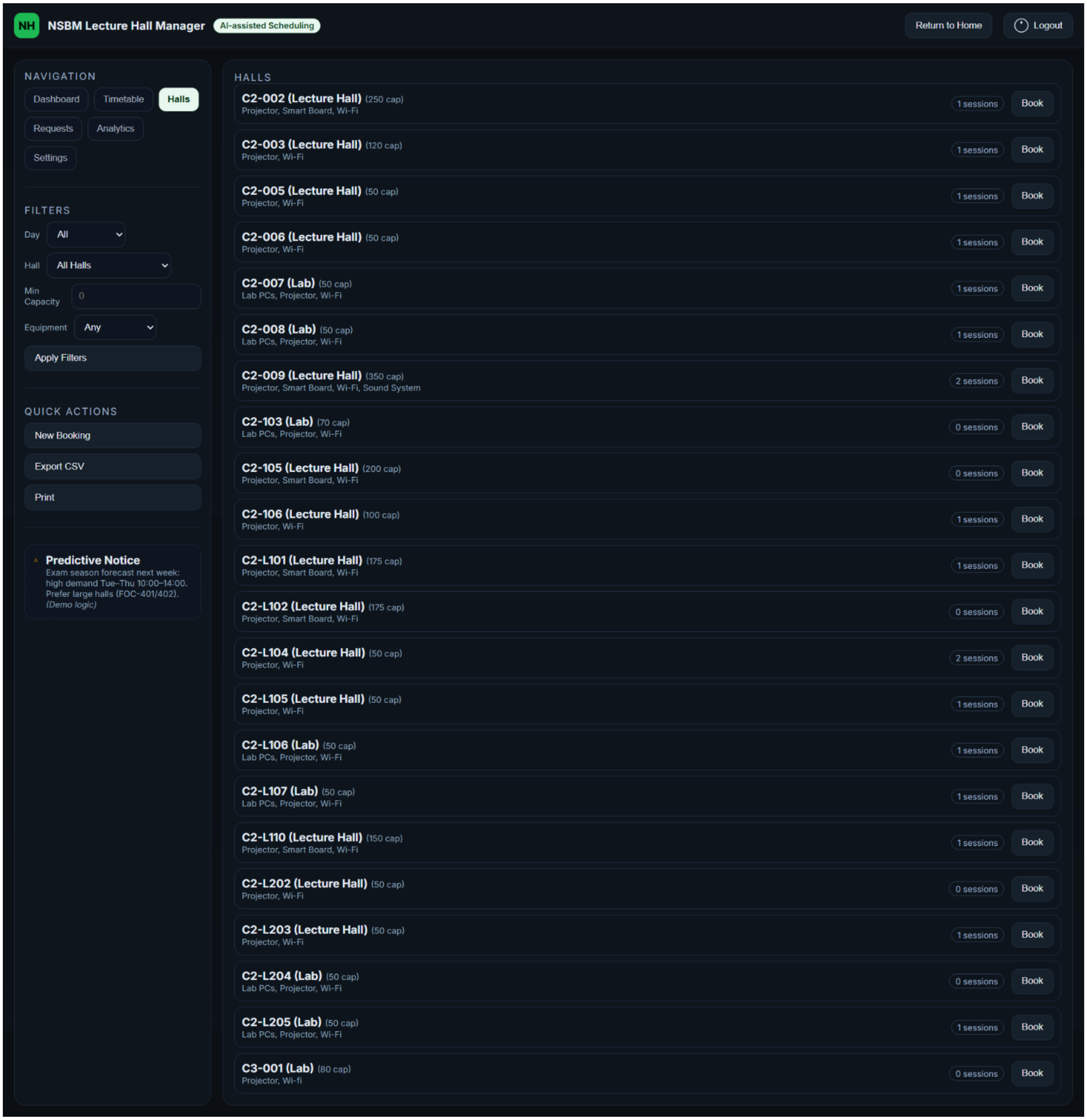


Figure 17: Lecturer & Student Dashboard

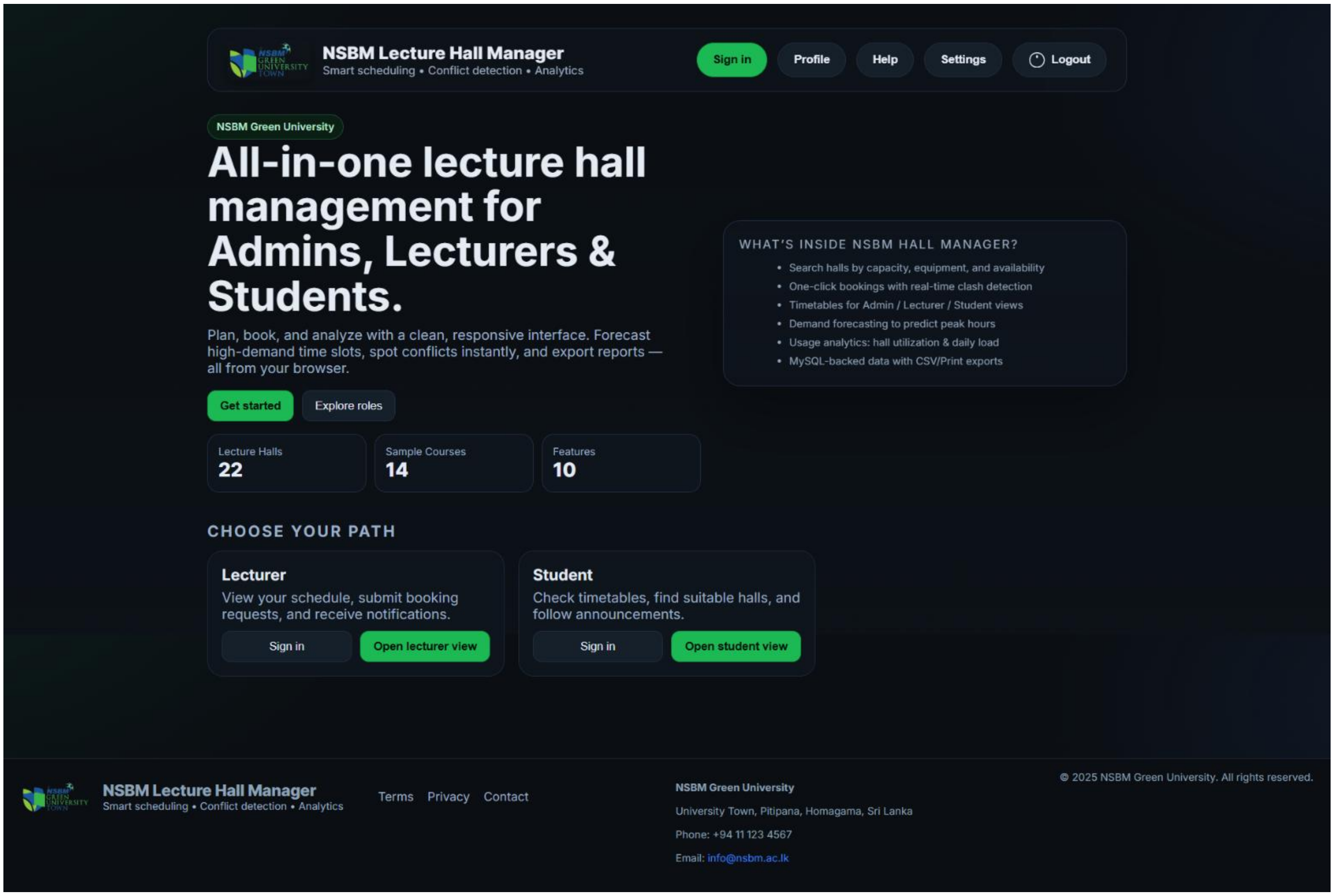


Figure 18: Help Page Interface

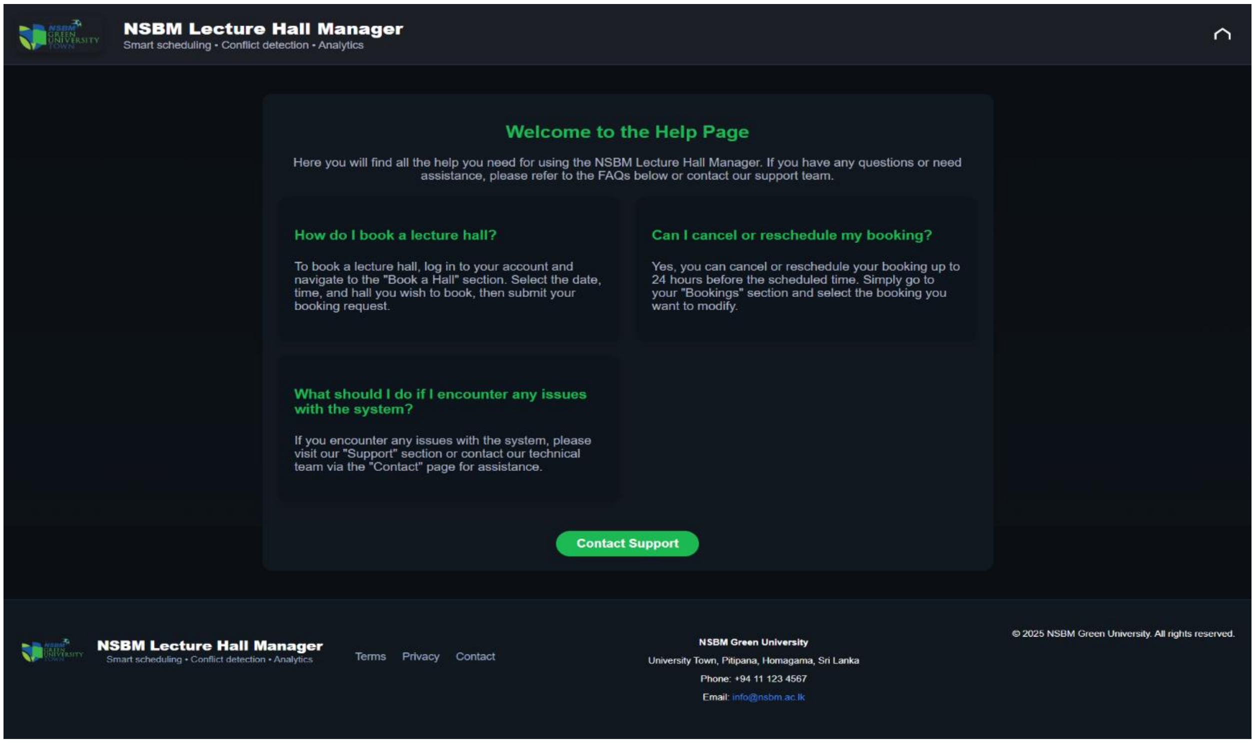


Figure 19: Feedback Submission Interface

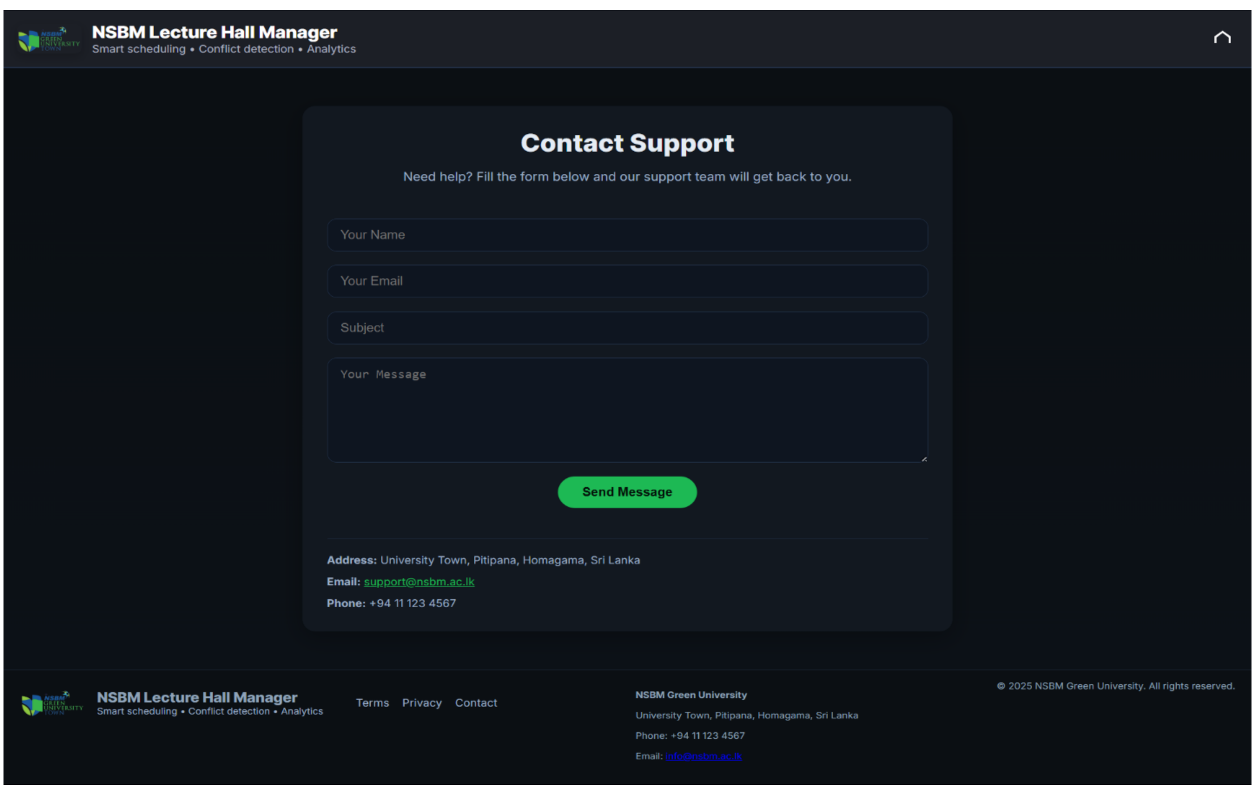


Figure 20: Lecturer Dashboard Interface

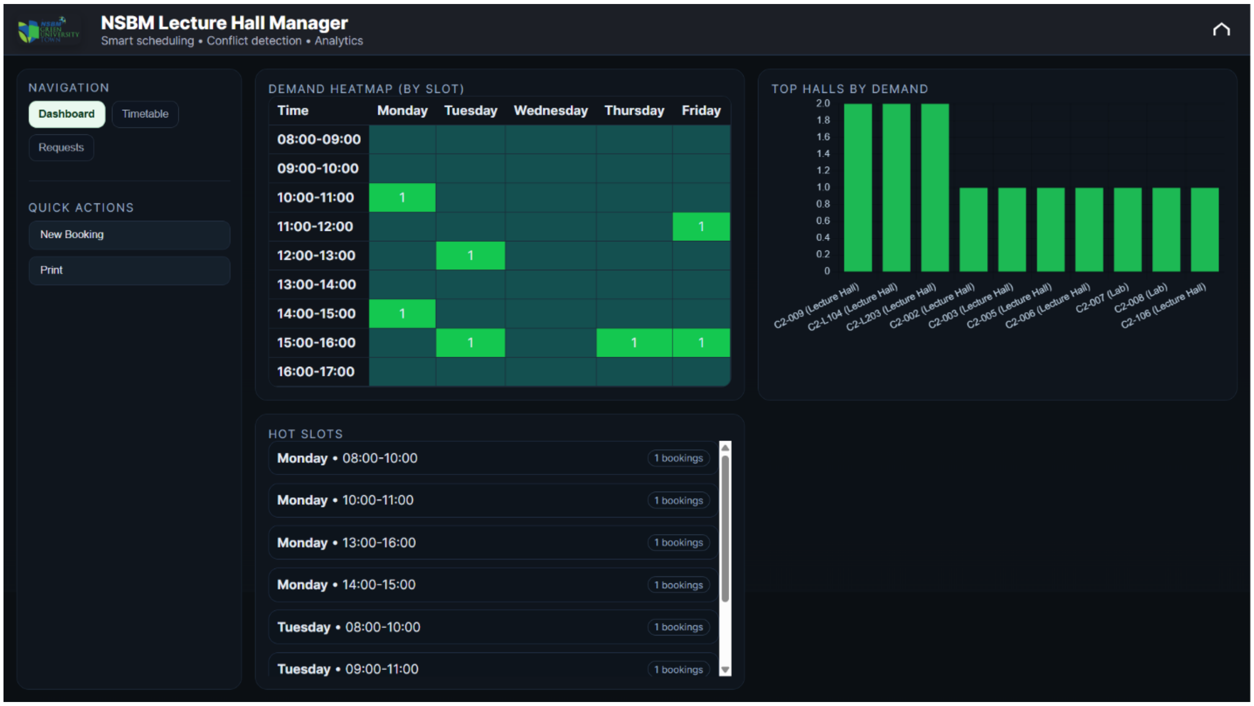


Figure 21: Student Dashboard Interface

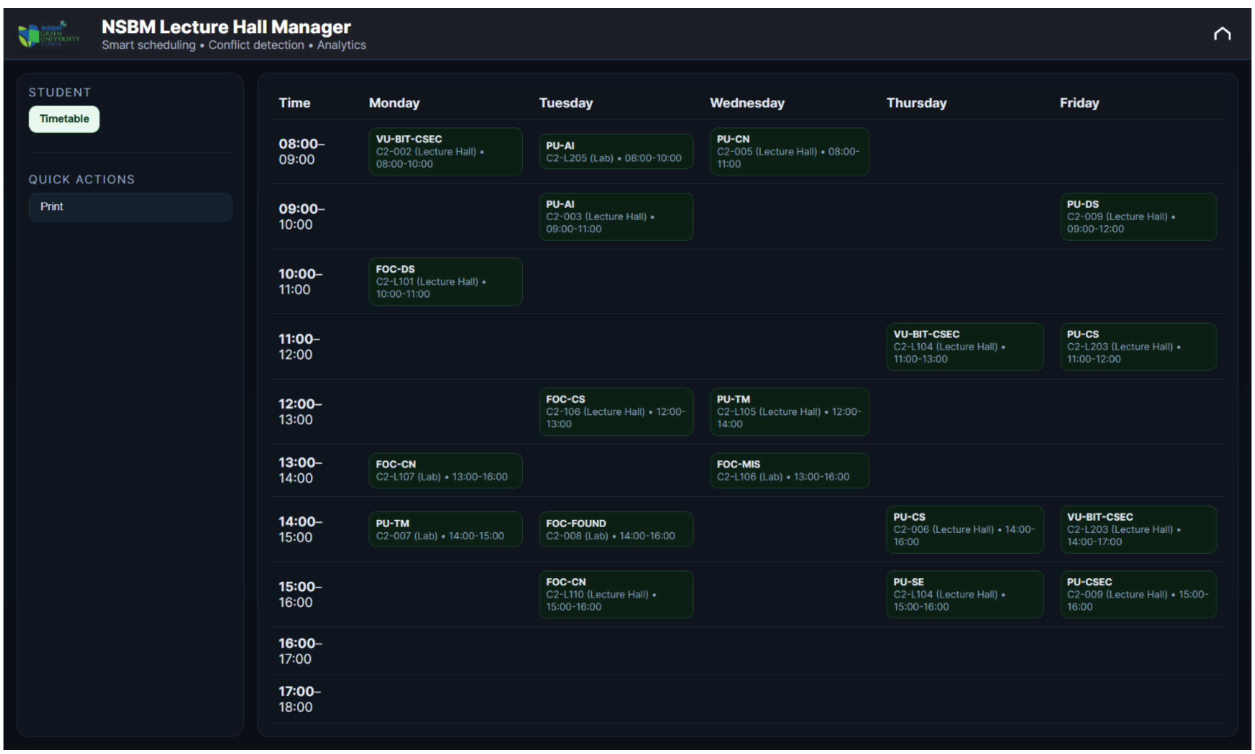


Figure 22: Profile Information Interface

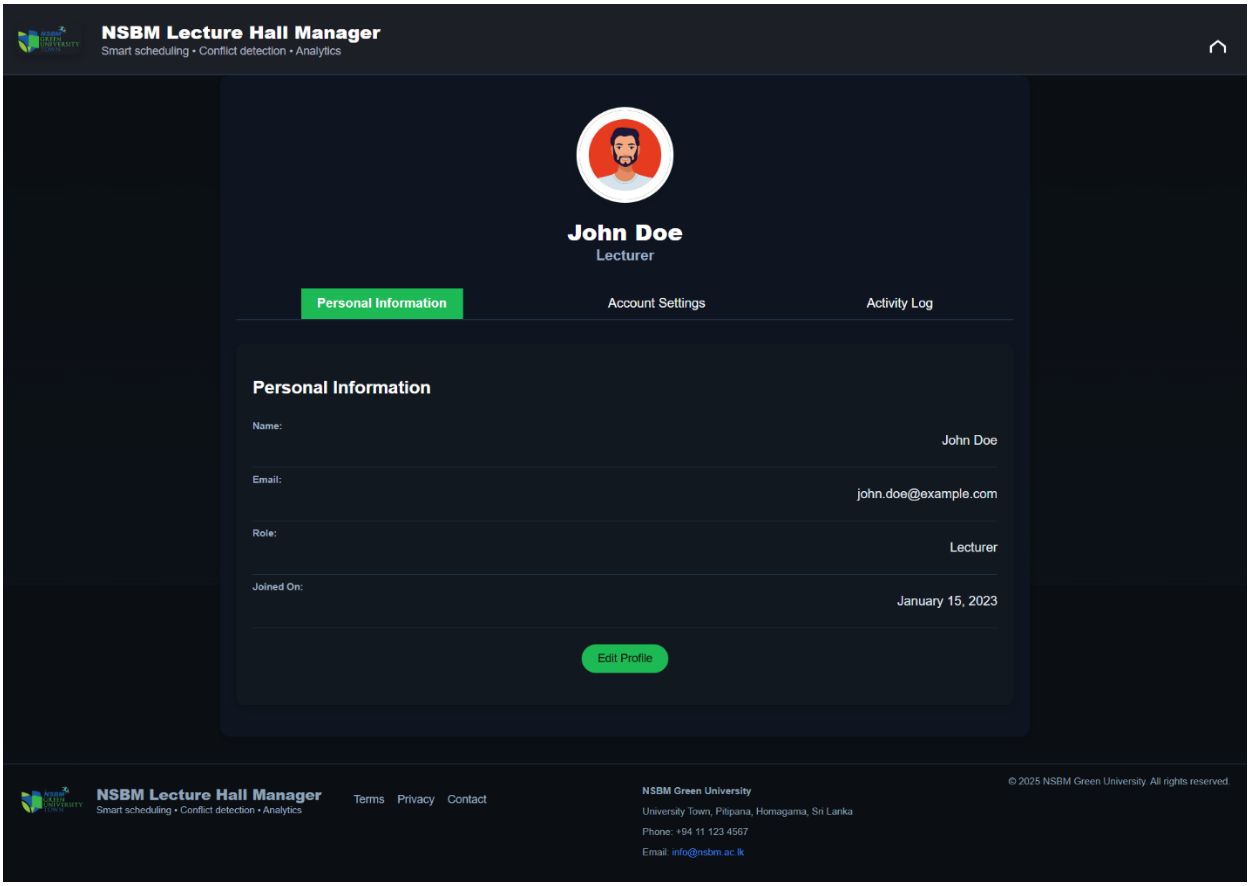


Figure 23: Database Table Structure Screenshot

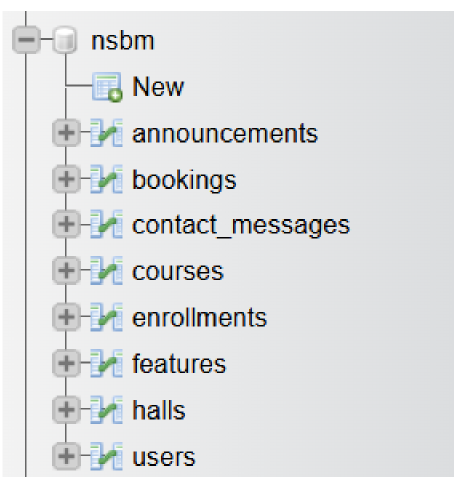
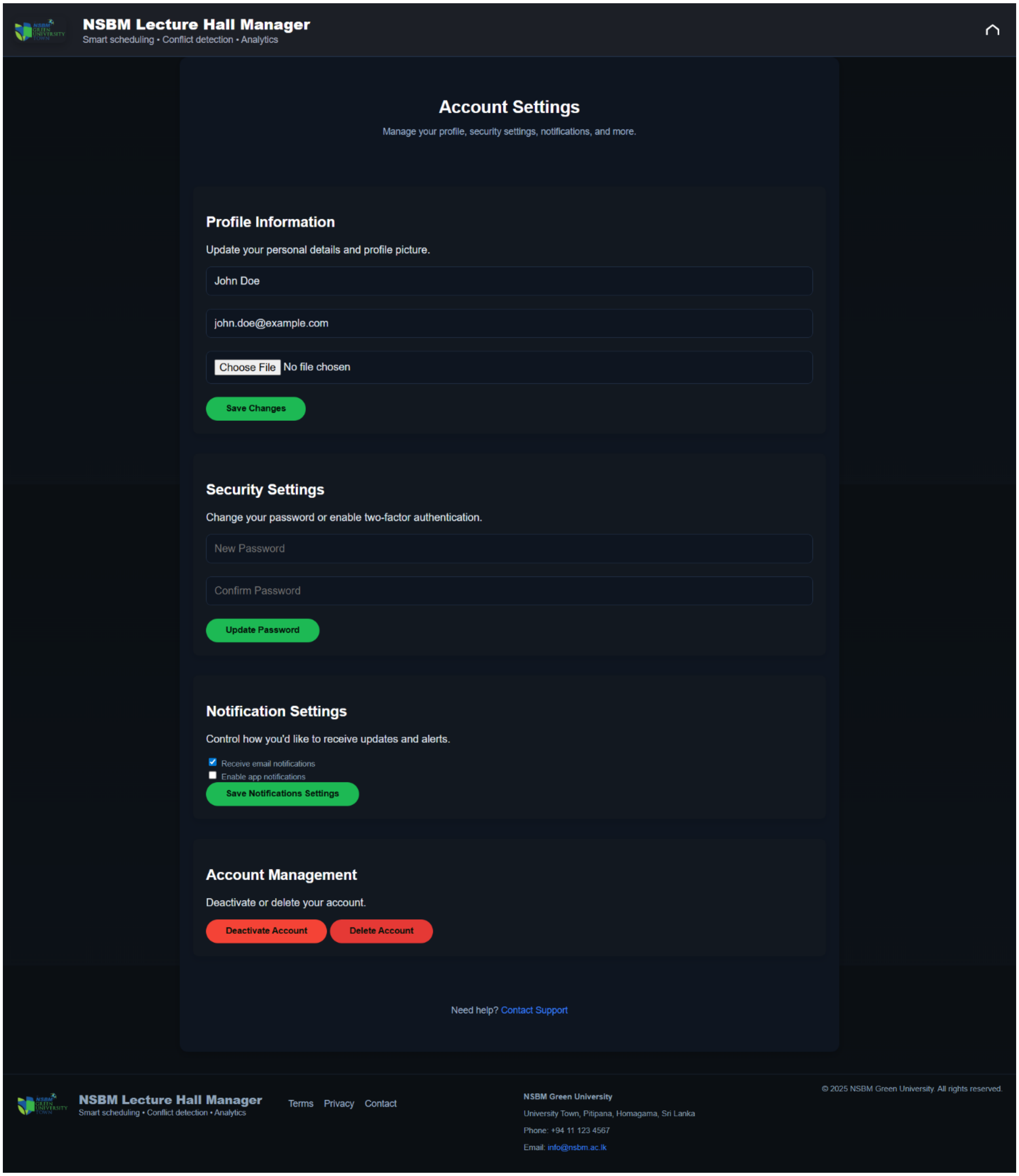


Figure 24: Account Setting Interface



Appendix C: Meeting Record Forms

- Research Project - Student Progression Report:

Research Project - Individual
Student Progression Report
[Supervisor Copy]

01. Student Name
E.G. Tharusha Dilhara Egodage

02. Index Number
22717

03. Degree Program
BSc in Management Information Systems (Special)

04. Supervisor Name
Mr. Gayan Perera

05. Project Title
Data-driven and AI-powered lecture hall management system for optimal scheduling, resource allocation and future prediction in university operations.

Meeting Number	Meeting 01	Meeting 02	Meeting 03	Meeting 04	Meeting 05	Meeting 06	Meeting 07
Date	27/9/24	14/10/24	13/11/24	5/05/2025	18/08/2025	27/08/2025	08/09/2025
Student Signature							
Supervisor Signature							

Meeting Number	Meeting 08	Meeting 09	Meeting 10	Meeting 11	Meeting 12	Meeting 13	Meeting 14
Date							
Student Signature							

Figure 25: Student Progression Report I

- Research Project - Student Progression Report:

NSBM GREEN UNIVERSITY TOWN

Supervisor Signature							

Documentations	Proposal 10%	Interim 01 10%	Interim 02 10%	Final Submission (Report + Viva) 70%
Date				
Approved (Yes / No)	Yes			
Student Signature				
Supervisor Signature				

Other Comments (Supervisor Use Only)

Figure 26: Student Progression Report II

- Meeting 1 Record:

NSBM GREEN UNIVERSITY TOWN

Final Year Project – Supervisory meeting minutes

Meeting No: 01

Date : 27/09/2024

Project Title : Data-driven and AI-Powered lecture hall management system for optimal scheduling, resource allocation and future prediction in university operations.

Name of the Student : E.G.T.D.Egodage

Student ID : 22717

Name of the Supervisor : Mr. Gayan perera

Items discussed:

* Discussed about the project title and areas to consider.
* My initial research idea was rejected and discussed new ideas related to information systems for the research.

Items to be completed before the next supervisory meeting:

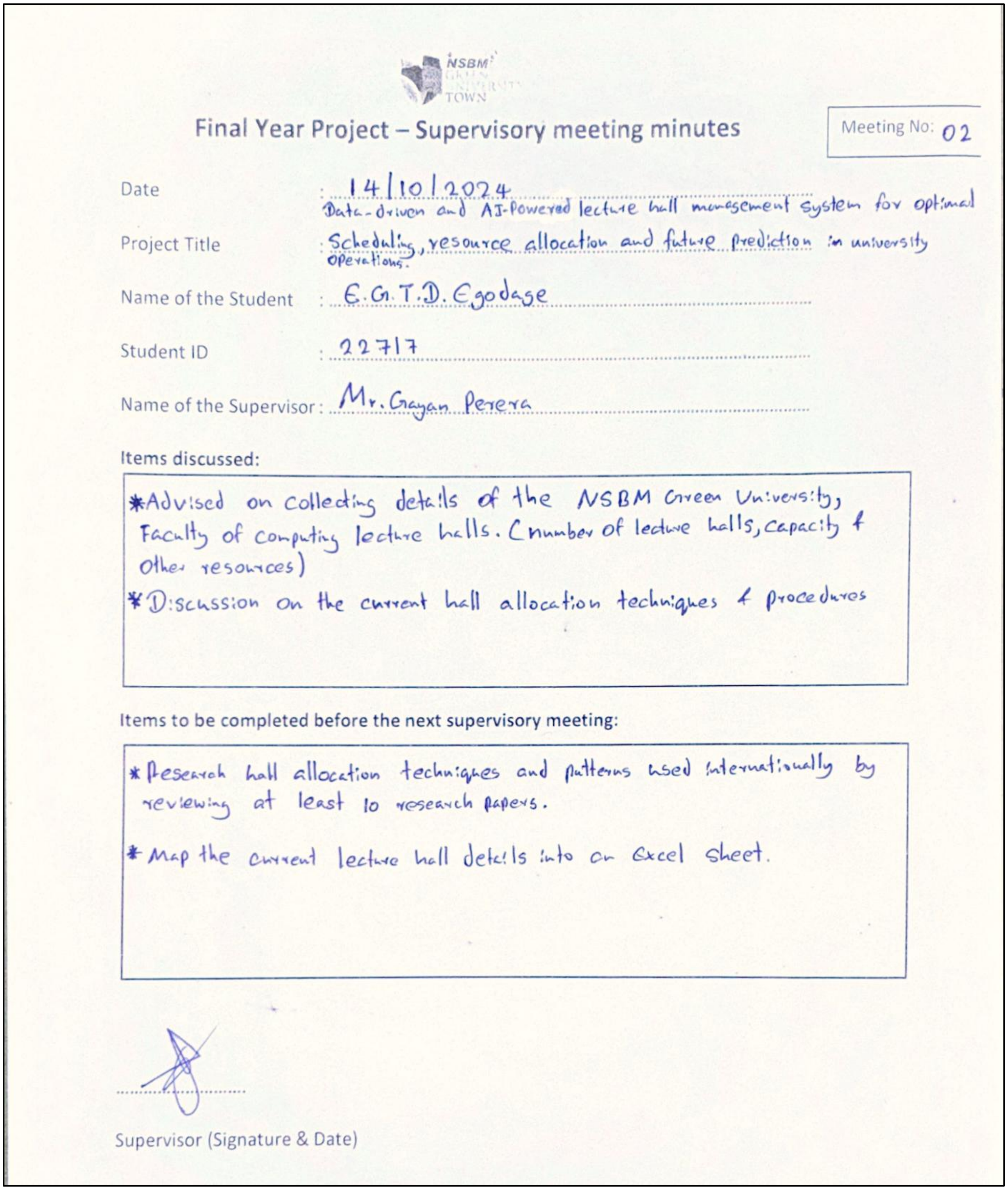
* Find the new three research ideas

....

Supervisor (Signature & Date)

Figure 27: Supervisor Meeting 01

- Meeting 2 Record:



NSBM GREEN UNIVERSITY TOWN

Final Year Project – Supervisory meeting minutes

Meeting No: 02

Date : 14/10/2024

Project Title : Data-driven and AI-Powered lecture hall management system for optimal Scheduling, resource allocation and future prediction in university operations.

Name of the Student : E.G.T.D. Egodage

Student ID : 22717

Name of the Supervisor : Mr. Gayan Perera

Items discussed:

* Advised on collecting details of the NSBM Green University, Faculty of computing lecture halls. (number of lecture halls, capacity & other resources)
* Discussion on the current hall allocation techniques & procedures

Items to be completed before the next supervisory meeting:

* Research hall allocation techniques and patterns used internationally by reviewing at least 10 research papers.
* Map the current lecture hall details into an Excel sheet.

Supervisor (Signature & Date)

Figure 28: Supervisor Meeting 02

- Meeting 3 Record:

NSBM GREEN UNIVERSITY TOWN

Final Year Project – Supervisory meeting minutes

Meeting No: 03

Date : 13/11/2024

Project Title : Data-driven and AI-powered lecture hall management system for optimal scheduling, resource allocation and future prediction in university operations.

Name of the Student : E.G.T.D. Egodage

Student ID : 22717

Name of the Supervisor : Mr. Gayan Perera

Items discussed:

* Discussed Particle Swarm Optimization (PSO) and its application in the project scenario.
* Reviewed the concept of Simulated Annealing (SA) and its working principles
* Talked about how to perform hall allocation by considering lecture requirements along with qualitative and quantitative input parameters.

Items to be completed before the next supervisory meeting:

* Study genetic algorithms and intelligent agents, including their capabilities, significance, setup and operational mechanisms.
* Conduct research on the topic. "what can genetic algorithms and intelligent agents do?"

Supervisor (Signature & Date)

Figure 29: Supervisor Meeting 03

- Meeting 4 Record:

NSBM GREEN UNIVERSITY TOWN

Final Year Project – Supervisory meeting minutes

Meeting No: 04

Date : 05/05/2025

Project Title : Data-Driven and AI-Powered lecture hall management System for optimal Scheduling, resource allocation, and future Prediction in University operations.

Name of the Student : E.G.T.D.Egodage

Student ID : 22717

Name of the Supervisor : Mr. Gayan Perera

Items discussed:

- Discussed a full overview of the current system, including it's components, System inputs, outputs and identified gaps.
- Received guidance on how to review related research and how to incorporate relevent studies into the literature review section of the Project.

Items to be completed before the next supervisory meeting:

- Design a proposed new system to address the identified gaps.
- Describe the high-level architecture of the proposed system.
- Develop a high-level architecture diagram of the new system, detailling all System Components, involed stakeholders, system inputs, and expected outputs.

....

Supervisor (Signature & Date)

Figure 30: Supervisor Meeting 04

- Meeting 5 Record:

NSBM GREEN UNIVERSITY TOWN

Final Year Project – Supervisory meeting minutes

Meeting No: 05

Date : 18/08/2025

Project Title : Data-Driven and AI-powered lecture hall management system for optimal scheduling, resource allocation and future prediction in University operations.

Name of the Student : E.G.T.D. Egodage

Student ID : 22717

Name of the Supervisor : Mr. Gayan Perera

Items discussed:

*Reviewed the high-level architecture diagram of the proposed system and discussed necessary changes to improve clarity and completeness

*Discussed the usercase diagram of the system and focusing on which components and relationships need to included.

Items to be completed before the next supervisory meeting:

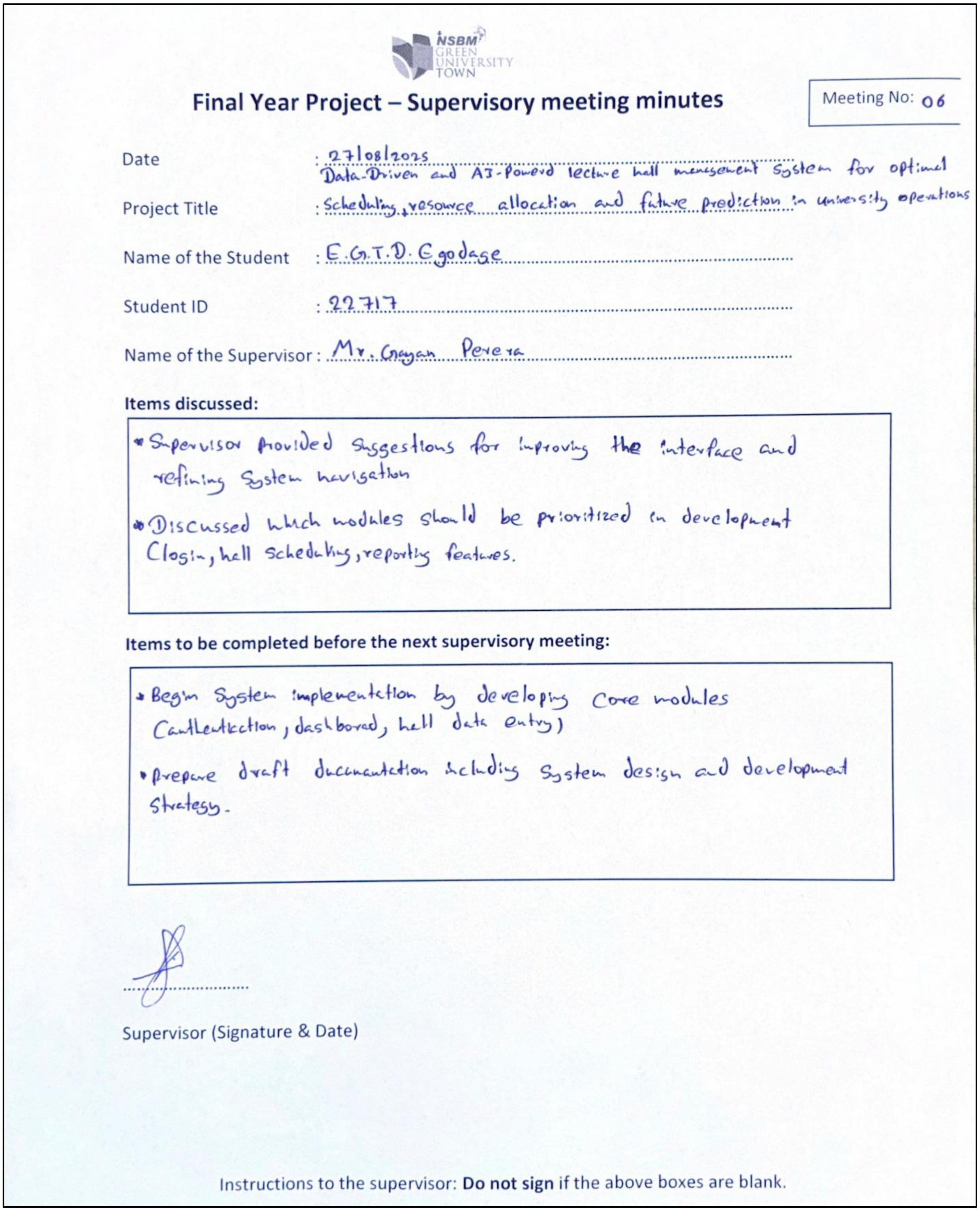
* Design the user interfaces and develop a prototype of the new system.
* Draft a detailed implementation plan, including Gantt chart outlining project milestones and timelines.

Supervisor (Signature & Date)

Instructions to the supervisor: **Do not sign** if the above boxes are blank.

Figure 31: Supervisor Meeting 05

- Meeting 6 Record:



NSBM GREEN UNIVERSITY TOWN

Final Year Project – Supervisory meeting minutes

Meeting No: 06

Date : 27/08/2025

Project Title : Data-Driven and AI-Powered lecture hall management system for optimal scheduling, resource allocation and future prediction in university operations

Name of the Student : E.G.T.D. Egodage

Student ID : 22717

Name of the Supervisor : Mr. Gayan Perera

Items discussed:

* Supervisor provided suggestions for improving the interface and refining system navigation
* Discussed which modules should be prioritized in development (login, hall scheduling, reporting features.

Items to be completed before the next supervisory meeting:

* Begin system implementation by developing core modules (authentication, dashboard, hall data entry)
* Prepare draft documentation including system design and development strategy.

Supervisor (Signature & Date)

Instructions to the supervisor: **Do not sign** if the above boxes are blank.

Figure 32: Supervisor Meeting 06

- Meeting 7 Record:

NSBM GREEN UNIVERSITY TOWN

Final Year Project – Supervisory meeting minutes

Meeting No: 07

Date : 08/09/2025

Project Title : Data-Driven and AI-Powerd lecture hall management system for optimal Scheduling, resource allocation and future prediction in university operation

Name of the Student : E.G.T.D. Egodage

Student ID : 22717

Name of the Supervisor : Mr. Gayan Perera

Items discussed:

* Discussed the key components and sections that must be included in the final thesis submission.
* Revived the current progress of the system development and identified pending tasks.
* Supervisor explained how to prepare an effective test plan, including the structure and essential elements to be covered during testing.

Items to be completed before the next supervisory meeting:

* Complete the front-end development of the system.
* Begin work on the back-end development components.

Supervisor (Signature & Date)

Instructions to the supervisor: **Do not sign** if the above boxes are blank.

Figure 33: Supervisor Meeting 07