

**DESIGN AND IMPLEMENTATION OF A CLASS WEB-BASED ACADEMIC
RESOURCE DISTRIBUTION SYSTEM**



VIN-IGHALO PRECIOUS EBANEHITA

PSC2105416

DEPARTMENT OF COMPUTER SCIENCE,

FACULTY OF PHYSICAL SCIENCES

UNIVERSITY OF BENIN,

BENIN CITY,

EDO STATE, NIGERIA.

NOVEMBER, 2025.

**DESIGN AND IMPLEMENTATION OF A CLASS WEB-BASED ACADEMIC
RESOURCE DISTRIBUTION SYSTEM**

VIN-IGHALO PRECIOUS EBANEHITA

PSC2105416

**A PROJECT REPORT SUBMITTED TO
THE DEPARTMENT OF COMPUTER SCIENCE,
FACULTY OF PHYSICAL SCIENCES,
UNIVERSITY OF BENIN, BENIN CITY.**

**IN PARTIAL FUFILLMENT OF THE REQUIREMENT FOR THE
AWARD OF A BACHELOR OF SCIENCE (B.Sc.) DEGREE IN
COMPUTER SCIENCE**

NOVEMBER 2025.

DECLARATION

I, **VIN-IGHALO PRECIOUS EBANEHITA**, with Matriculation number **PSC2105416**, do hereby declare that:

1. This project work is based on a study undertaken by me in the Department of Computer Science, University of Benin, Benin City, under the supervision **Dr. E. NWELIH**.
2. This research work has not been previously submitted for the award of degree elsewhere.
3. All ideas and views are a product of my personal research; and where the views of others have been expressed; they were duly acknowledged.
4. All liabilities arising from the study are entirely mine and not of the supervisor.

Vin-Ighalo Precious Ebanehita

DATE

CERTIFICATION

This is to certify that this project work was carried out by **VIN-IGHALO PRECIOUS EBANEHITA** with Matriculation Number **PSC2105416** under my supervision. It is adequate and satisfactory, both in scope and content, for the award of Bachelor of Science (B.Sc.) Degree in Computer Science of the University of Benin.

Dr. E. NWELIH

Project Supervisor

DATE

APPROVAL

This project work is hereby approved in partial fulfillment of the requirements for the award of Bachelor of Science (B.Sc.) Degree in Computer Science from the University of Benin.

Dr. (Mrs.) A.R. Usiobaifo

DATE

Head of Department

DEDICATION

This work is first and foremost dedicated to God Almighty, the source of my strength, wisdom, and inspiration. His grace and guidance have brought me this far and made the completion of this project possible.

I also dedicate this project to my late father, my superhero, late Mr. Vincent Ighalo, whose love, support, and encouragement shaped my academic journey. He always seemed to have the right answer to everything and believed in my potential long before I saw it myself. His legacy continues to inspire me every day.

Lastly, I dedicate this work to my mother, the strongest woman I know, who has carried me with unwavering strength, love, and sacrifice. Her resilience and dedication have been my greatest motivation.

This work stands as a reflection of their love, faith, and strength in my life.

ACKNOWLEDGEMENTS

First and foremost, I give all glory, honor, and praise to Almighty God for His grace, wisdom, and strength throughout the course of this project. Without His guidance, this work would not have been possible.

My sincere appreciation goes to my project supervisor, Dr. E. Nwelih, for his patience, invaluable guidance, and constructive advice which greatly contributed to the successful completion of this research. I also extend my gratitude to the Head of Department, Dr. (Mrs) R. O. Usiobaifo, and all the academic staff of the Department of Computer Science, University of Benin, for their dedication, support, and commitment to academic excellence.

I deeply appreciate my mother, Mrs. Josephine Ighalo, whose love, prayers, and encouragement have been my constant source of motivation. And to my siblings Ose, Ebose, Daniella and my cousin, Fortune, my heartfelt gratitude goes to you. Thank you for your unwavering support all through this process.

To my friends, especially, Christabel Chukwura, Wisdom Ijeh, Gabriel Ighietsemhe, Moyinoluwa Akinnifesi, Kelvin Uwabor, Saibu Eunice and Ejemai Clarissa, thank you for your friendship, encouragement, and assistance throughout this journey. Your contributions, both directly and indirectly, made this project a rewarding experience.

Finally, I am grateful to everyone who in one way or another contributed to the success of this project, your efforts and kindness are sincerely appreciated.

ABSTRACT

This project presents the design and implementation of a Class Academic Resource Distribution System developed for the Department of Computer Science, University of Benin. The system was created to solve the problem of disorganized and unverified learning materials shared through informal channels such as WhatsApp and Google Drive. It provides a centralized web-based platform where lecturers and class representatives can upload course-specific, lecturer-approved materials, while students can easily access them in one place. The project adopted the Rapid Application Development (RAD) methodology and was implemented using HTML, CSS, Python (Django), and MySQL. System testing confirmed efficient performance, secure access control, and reliable file management. The platform successfully improves accuracy, accessibility, and communication among students, lecturers, and class representatives, laying a strong foundation for future departmental and university-wide deployment.

TABLE OF CONTENTS

DECLARATION	iii
CERTIFICATION	iv
APPROVAL	v
DEDICATION	vi
ACKNOWLEDGEMENTS	vii
ABSTRACT	viii
LIST OF FIGURES	xii
CHAPTER ONE	1
INTRODUCTION	1
1.1. Background Study	1
1.2 Statement of the Problem	3
1.3 Aim and Objectives	3
1.4 Scope of the Study	3
1.5 Research Methodology	4
1.6 Research Significance	6
CHAPTER TWO	8
LITERATURE REVIEW	8
2.1 Overview of Academic Resource Distribution Systems	8
2.2 Academic Resource Sharing Practices in Nigerian Universities	9
2.3 Traditional (Manual) Methods of Academic Resource Distribution	10
2.4 Challenges of Manual Systems	11
2.5 Automated/Online Resource Distribution Systems	12
2.6 Comparison between Manual and Online Methods	13
2.7 Related Works	14
CHAPTER THREE	17
SYSTEMS ANALYSIS AND DESIGN METHODOLOGY	17
3.1 Introduction	17
3.2 System Analysis	18
3.2.1 Problem Definition	18
3.2.2 System Objectives	19

3.2.3 Analysis of Existing System	20
3.3 System Design	22
3.3.1 Conceptual Model of the Proposed System	22
3.3.2 Input Design	24
3.3.3 Output Design	25
3.3.4 Process Design	27
3.3.5 Database Design	30
3.4 System Requirements	34
3.4.1 Hardware Requirements	34
3.4.2 Software Requirements	35
3.5 Summary	36
CHAPTER FOUR	37
SYSTEM IMPLEMENTATION	37
4.0 Introduction	37
4.1 Implementation Tools	38
4.1.1 Programming Languages Used	38
4.1.2 Frameworks and Libraries	39
4.1.3 Development Environment	39
4.1.4 Deployment Environment	40
4.1.5 Operating System	40
4.1.6 Implementation Platform	40
4.2 System Modules and Interfaces	41
4.2.1 Login and Authentication Module	41
4.2.2 Student Dashboard	42
4.2.3 Class Representative Dashboard	43
4.2.4 Lecturer Dashboard	45
4.2.5 File Upload and Resource Distribution Module	45
4.2.6 Messaging Module	46
4.2.7 Database Interaction Layer	46
4.3 System Testing and Validation	47
4.3.1 Testing Approach	47

4.3.2 Test Case Scenarios	47
4.3.3 Test Results and Observations	48
4.4 Challenges Faced During Implementation	49
4.5 System Usability Evaluation	50
4.6 Limitations of the System	51
CHAPTER FIVE	53
CONCLUSION	53
5.1 Summary of the Study	53
5.2 Conclusion	54
REFERENCES	55
APPENDIX	59

LIST OF FIGURES

Figure 3.1: Use Case Diagram of the Academic Resource Distribution System	24
Figure 3.2: System Flowchart Showing User Process Flow	30
Figure 3.3: Class Diagram of the Academic Resource Distribution System	33
Figure 3.4: Entity Relationship Diagram of the Academic Resource Distribution System	34
Figure 3.5: System Architecture of the Academic Resource Distribution System	36
Figure 4.1: Screenshot of the system's landing page	42
Figure 4.2: Screenshot of the login page.	42
Figure 4.3: Screenshot of a student dashboard.	43
Figure 4.4: Screenshot of the announcement module on the dashboard.	43
Figure 4.5: Screenshot of the class representative dashboard.	44
Figure 4.6: Screenshot of the message module on the rep's dashboard	44
Figure 4.7: Screenshot of the super admin interface	44
Figure 4.8: Screenshot of the lecturer's dashboard.	45
Figure 4.9: Screenshot of the file/material upload functionality.	46
Figure 4.10: Screenshot of the messaging functionality	46

CHAPTER ONE

INTRODUCTION

1.1. Background Study

In today's digital academic environment, access to learning materials is no longer the major problem; accuracy and relevance are. Students in Nigerian universities, especially in departments such as Computer Science, now have access to countless materials shared across platforms like WhatsApp, Telegram, Google Drive, and online forums. While this may appear to be an advantage, it often creates confusion and inconsistency, as many of these resources are not directly aligned with what lecturers teach or recommend. Research has demonstrated that information overload significantly impacts students' decision-making abilities and academic outcomes, causing unnecessary stress and cognitive fatigue (Tafesse et al., 2024).

Traditionally, lecturers shared materials physically through handouts or verbally referred students to specific textbooks. However, as universities have moved toward digital learning, that control over what students study has weakened. Students now receive multiple versions of lecture slides, outdated notes from past students, or random materials from online searches, many of which do not accurately represent the lecturer's syllabus. This lack of a centralized and authoritative platform means students often rely on incomplete or conflicting information (Koranteng et al., 2021).

Moreover, informal digital channels such as WhatsApp groups, although popular for convenience, are poorly structured for academic content management. Lee, Chern, and Azmir (2023) observed that while WhatsApp facilitates communication and quick information sharing among students, its unstructured nature poses significant challenges for organizing and retrieving academic materials. Files get buried under conversations, announcements are missed, and older resources are easily lost. Students report challenges including difficulty

navigating heavy volumes of messages and interrupted discourse on platforms (Naghdi-pour and Hancioğlu, 2022). The result is an environment where students, though surrounded by information, are uncertain about which materials are authentic or relevant to their courses.

The absence of a centralized repository for lecturer-approved materials creates disparities in learning experiences, where some students access correct resources while others work with outdated or incorrect information (Research.com, 2024). This inconsistency undermines equitable education and can significantly impact academic performance across student cohorts.

To address these issues, this study proposes the design and implementation of a web-based Academic Resource Distribution System focused on ensuring access to lecturer-approved materials. The system will centralize the process of resource distribution, allowing class representatives and lecturers to upload and manage official course documents while students can easily access them at any time. Research demonstrates that centralized platforms in higher education provide robust analytics capabilities and improve collaboration among departments (CoreHigherEd, 2023; EDUCAUSE, 2024).

This approach not only solves the problem of unequal access but also enhances academic quality by ensuring that the materials students rely on are authentic and aligned with their course objectives. With such a platform in place, lecturers regain control over the learning content, and students are shielded from the confusion that comes with unverified or outdated materials circulating online. However, this study does not aim to restrict students' access to information but to give them a verified academic direction, enabling deeper and more focused research.

Digital learning platforms have been shown to significantly enhance student engagement, motivation, and academic performance when properly implemented (Koranteng et al., 2021).

1.2 Statement of the Problem

Access to academic resources is a critical component of effective learning, yet many students in Nigerian universities face challenges in obtaining the right materials at the right time. Existing methods of distribution, such as printed handouts, peer-to-peer sharing, or informal online groups, often result in confusion, delays, loss of information, and unequal access among students.

In addition, lecturers may provide materials at different times and through different channels, making it difficult to ensure consistency and coverage across an entire class. Class representatives, who are often tasked with coordinating these resources, lack a structured platform to manage distribution effectively. This fragmented process affects students' preparedness, reduces efficiency in learning, and places unnecessary strain on both lecturers and course representatives.

Therefore, there is a need for a centralized, web-based academic resource distribution system that allows course representatives and lecturers to upload and share course-specific materials in a consistent manner, while enabling students to access them equally and without delay.

1.3 Aim and Objectives

The aim of this project is to develop a class-level web-based academic resource distribution system that enables course representatives and lecturers to share learning materials in a centralized and accessible manner for all students.

The specific objective of this project is to design and implement a secure web-based platform for distributing academic resources to students.

1.4 Scope of the Study

This project focuses on the design and implementation of a class-level web-based academic resource distribution system. The system is designed to serve as a centralized platform where

course representatives and lecturers can upload learning materials and announcements, and students can access them without delays or inequalities.

Specifically, the system covers:

- I. **Students:** who can log in with their matriculation number and password and access course materials automatically.
- II. **Class Representatives and Assistant Representatives (Admins):** who can upload student lists, create lecturer IDs, and post announcements.
- III. **Lecturers:** who can log in to upload academic resources and announcements for their assigned courses and also communicate with the course representatives.

The scope of this study is limited to the department of Computer Science, in the University of Benin. It does not extend to the entire university or include a departmental-level super admin dashboard, assignment management features, or advanced analytics. These functionalities are acknowledged as important but are recommended for future development.

1.5 Research Methodology

This study adopts a developmental research approach, focusing on the design and implementation of a class-level web-based academic resource distribution system. The methodology emphasizes system development, testing, and evaluation rather than traditional data collection techniques, as the main objective is to provide a functional and accessible platform for students, course representatives, and lecturers.

System Development Approach

The system is developed using the Rapid Application Development (RAD) methodology, which facilitates iterative design, rapid prototyping, and continuous feedback. This approach

allows for the system to be developed incrementally, ensuring that functionality is tested and refined throughout the development process. Key stages include:

- I. **Requirement Analysis:** Determining system functionalities based on the project scope, including student access, course rep and lecturer functions, and resource management features.
- II. **System Design:** Creating the overall architecture, user interface, and database design, supported by tools such as UML diagrams and Entity-Relationship Diagrams (ERDs).
- III. **System Implementation:** Developing the web-based platform using suitable technologies, ensuring secure login, resource upload/download, and administrative functionalities.
- IV. **Testing and Evaluation:** Conducting unit and integration testing to verify system performance, usability, and reliability. User testing by a small group of students and course representatives is also performed to ensure that the platform meets functional requirements.
- V. **Deployment and Documentation:** Final deployment of the system in a controlled environment, accompanied by detailed documentation of design decisions, system functionality, and usage guidelines.

Justification of Methodology

The RAD methodology is appropriate for this project because it supports rapid prototyping, early user feedback, and iterative refinement, which are critical when developing a user-focused web-based system. It allows the developers to adapt quickly to emerging requirements and ensures that the final platform is both functional and user-friendly. This

approach is particularly suitable for academic resource systems, where accessibility, reliability, and ease of use are priorities.

Limitations

While the methodology allows for effective system development, certain limitations exist:

- I. **Scope Restriction:** The system is limited to class-level resource distribution and does not cover departmental or university-wide administration.
- II. **User Base:** Testing and evaluation are conducted on a limited number of students and course representatives, which may not fully represent the entire population.
- III. **Resource Constraints:** Limited access to advanced development tools or server infrastructure may affect system performance during testing.
- IV. **Feature Limitations:** Advanced functionalities such as automated analytics, assignment management, or mobile notifications are not included in this initial implementation and are recommended for future development.

Despite these limitations, the methodology provides a practical and structured framework for creating a functional and usable academic resource distribution system.

1.6 Research Significance

The development of a class-level web-based academic resource distribution system offers several benefits:

- I. **Improved Access to Materials:** Students can access course materials in a centralized and timely manner, reducing delays and ensuring equitable access.
- II. **Improved Learning Accuracy and Focus:** One of the major benefits of the proposed system is that it ensures students study using the exact materials recommended by their

lecturers. This helps them stay focused on the right scope of the course and prevents the confusion that comes from using multiple, conflicting resources gathered from random or unverified sources.

- III. **Reduced Misinformation:** The system also helps curb the spread of outdated or inaccurate course materials, which are common in informal student sharing channels. This ensures that what students study is always relevant, verified, and in line with current course objectives.
- IV. **Simplified Resource Management for Course Representatives:** Reps can easily manage student lists, semesters, and lecturer accounts, streamlining administrative tasks.
- V. **Efficient Resource Distribution for Lecturers:** Lecturers can upload materials and announcements for their assigned courses without repeated manual sharing.
- VI. **Monitoring Enrollment:** Lecturers and course reps can view the number of students enrolled in each course and access student lists, allowing basic oversight of participation.
- VII. **Foundation for Future Enhancements:** The system lays the groundwork for more advanced functionalities, such as detailed engagement tracking, assignment management, notifications, and departmental analytics.

Overall, the study underscores the importance of centralized, web-based solutions in enhancing learning efficiency, equity, and communication in academic settings whilst also improving learning accuracy without stifling curiosity; students are guided by lecturer-approved materials but remain free to explore beyond them.

CHAPTER TWO

LITERATURE REVIEW

This chapter reviews existing literature on academic resource distribution systems, with a focus on Nigerian federal universities and a case study of the University of Benin, Department of Computer Science. It examines how students and lecturers currently share academic materials, the limitations of traditional/manual methods, and the emerging role of web-based systems in addressing these challenges.

2.1 Overview of Academic Resource Distribution Systems

Academic resource distribution systems are platforms or mechanisms designed to facilitate the sharing of learning materials, such as lecture notes, textbooks, and multimedia resources, between lecturers, course representatives, and students. Traditionally, these systems were manual, relying on printed handouts, notice boards, and peer-to-peer sharing. While functional to an extent, such methods are often inconsistent, prone to delays, and lack mechanisms to track or monitor resource dissemination (Ebhonu et al., 2021; Salau et al., 2022).

In recent years, digital solutions have increasingly replaced manual approaches. Web-based academic resource systems allow for centralized storage, structured access, and controlled distribution of materials. They also enable real-time updates, ensuring that all students have equal access regardless of location or time constraints (Zhang et al., 2022; Kumar and Sharma, 2021). Globally, institutions have adopted platforms ranging from simple content management systems to complex learning management systems (LMS) that integrate announcements, resources, and student engagement tracking (Olaniyi et al., 2022).

In the Nigerian context, federal universities have begun exploring digital solutions to improve resource sharing. However, access remains uneven, with most institutions implementing basic web portals that serve more as repositories than fully interactive platforms. For instance,

while the University of Lagos provides lecture materials online, the system lacks features for class-level monitoring and targeted resource distribution (Salau et al., 2022). At the University of Benin, Department of Computer Science, students primarily rely on course representatives and informal online groups for resource sharing, highlighting the need for a dedicated, web-based class-level system.

Central to these systems is the concept of centralization and accessibility, which ensures that students receive the right materials at the right time while allowing lecturers and course representatives to manage distributions efficiently. The proposed system is different because it filters content to only what lecturers or approved admins upload, ensuring accuracy and relevance. This review sets the stage for analyzing the limitations of traditional methods and justifies the development of a web-based academic resource distribution system for the University of Benin.

2.2 Academic Resource Sharing Practices in Nigerian Universities

In many Nigerian universities, students rely on multiple informal sources such as past students' notes, random online PDFs, and social media files. While this reflects student initiative, it often results in fragmented understanding since the materials may not align with what the lecturer teaches.

The sharing of academic resources is often a combination of formal and informal practices. Formal channels include university library portals, departmental notice boards, and lecturer-managed course websites, while informal channels encompass social media groups, messaging apps, and peer-to-peer distribution (Amos and Bassi, 2025). These methods have historically aimed to supplement the limited access to printed textbooks and lecture notes prevalent in many Nigerian institutions.

For example, at the University of Benin, Department of Computer Science, students primarily rely on WhatsApp groups to share lecture notes, assignments, and announcements. Course representatives manage these groups, ensuring that materials shared by lecturers reach all students. While this method improves accessibility compared to purely manual approaches, it remains informal and lacks structured features such as centralized tracking, secure storage, and easy retrieval of past resources. Students may still face challenges in verifying resource authenticity, accessing materials promptly during network issues, or finding older content efficiently.

The growing reliance on WhatsApp highlights the demand for fast and convenient resource-sharing channels and underscores the potential for a formalized web-based system. Such a system could centralize distribution, ensure equitable access, and complement existing informal practices without entirely replacing them (Ebhonu, Ajani, and Isah, 2021; Ogunbodede and Oribhabor, 2022).

2.3 Traditional (Manual) Methods of Academic Resource Distribution

Before the widespread adoption of digital platforms, academic resources in Nigerian universities were primarily distributed through manual methods. These included printed lecture notes, handouts, textbooks issued from departmental libraries, and notice boards for announcements. Students often had to physically visit lecturers, departmental offices, or the library to obtain necessary materials, which made the process time-consuming and dependent on individual initiative (Ebhonu et al., 2021).

At the University of Benin, Department of Computer Science, these manual methods are still used to some extent. For instance, lecturers sometimes provide printed notes or slides during class sessions. Students rely on course representatives to collect and share printed materials with their peers. However, this approach has several limitations, including:

- I. **Delayed access:** Students not present during handouts or notice postings may miss critical materials.
- II. **Limited reach:** Physical distribution is constrained by classroom size and the willingness of intermediaries to share materials.
- III. **Inefficiency:** Managing multiple copies of notes or announcements manually is time-consuming for lecturers and course representatives.
- IV. **Loss or damage:** Printed materials can be misplaced or deteriorate, reducing their usability over time.

Studies have highlighted that such manual systems often result in inconsistencies in the dissemination of academic resources, contributing to inequality among students and increasing administrative burden for lecturers and course representatives (Ogunbodede and Oribhabor, 2022). These challenges underscore the necessity for a centralized, web-based system that ensures consistent, timely, and accessible distribution of academic materials for all students.

2.4 Challenges of Manual Systems

Manual methods of academic resource distribution in Nigerian universities, such as printed handouts, departmental notice boards, and peer-to-peer sharing, face several significant challenges. These traditional approaches are inherently limited in scope and efficiency, particularly when compared to modern, web-based systems.

Firstly, manual systems often cause delays in resource access. Students who are absent during lectures or when materials are posted on notice boards may miss crucial content, affecting their academic preparedness (Akinola and Olatunde, 2021).

Secondly, the reach of manual distribution is limited. Physical handouts can only be given to students present in class or through course representatives, creating unequal access. This often leads to disparities in learning opportunities, as not all students receive the materials in a timely manner (Iwu et al., 2022).

Another challenge is inefficiency in management. Lecturers and course representatives spend significant time duplicating materials, coordinating distribution, and tracking who has received them. This administrative burden reduces time available for teaching and mentorship.

Lastly, loss or damage of materials is common. Printed notes can be misplaced, deteriorate over time, or be mismanaged, reducing the usability of academic resources. This risk compromises the consistency and reliability of information delivery (Akinola and Olatunde, 2021).

These challenges underscore the need for a centralized, web-based academic resource distribution system. Such a system can ensure timely, equitable, and organized access to course materials, reducing administrative strain and enhancing student learning outcomes.

2.5 Automated/Online Resource Distribution Systems

With increasing internet penetration and smartphone usage in Nigeria, many universities have begun integrating digital platforms for distributing academic resources. These systems include learning management systems (LMS), cloud-based file sharing, and instant messaging platforms, which allow lecturers to upload lecture notes, assignments, and announcements, while students can access them anytime and anywhere (Okoro and Nwachukwu, 2021).

Platforms like Moodle, Google Classroom, and WhatsApp groups have been adopted to varying degrees. For instance, at the University of Benin, Department of Computer Science, WhatsApp groups have become the most prevalent method for sharing resources,

announcements, and quick clarifications among students and lecturers. These platforms increase accessibility, reduce reliance on physical handouts, and provide timely communication of academic materials.

However, these systems (like LMSs or online repositories) often focus on volume of information, not validation of the source. Also, automated systems still face limitations in many Nigerian universities. Challenges include lack of centralized coordination, unequal student access to reliable internet, and limited awareness or training among lecturers and students on effectively using the platforms (Adegoke and Balogun, 2022). Furthermore, some digital methods are ad-hoc, relying on informal channels like WhatsApp, which can create inconsistencies in resource dissemination.

Despite these challenges, the shift towards online resource distribution demonstrates the potential of web-based systems to overcome the inefficiencies of manual methods, highlighting the need for a structured, centralized academic resource distribution platform at the class level.

2.6 Comparison between Manual and Online Methods

The distribution of academic resources in Nigerian universities has historically relied on manual methods, such as printed lecture notes, textbooks, departmental notice boards, and face-to-face interactions with lecturers or course representatives. While these approaches were once the standard, they often led to inefficiencies including delayed access, limited reach, and increased administrative burden (Adebayo and Salami, 2021). Students who were absent from class or lacked proactive communication channels could miss critical resources, which affected learning outcomes.

In contrast, online and web-based methods, ranging from departmental websites to social media groups and cloud-based platforms, provide centralized and more equitable access to

learning materials. Students can access uploaded resources anytime, from anywhere, reducing dependence on intermediaries and mitigating the risk of lost or damaged documents (Iwu et al., 2022). Furthermore, online systems allow lecturers and course representatives to post announcements and materials in real-time, which enhances consistency across the class.

At the University of Benin, Department of Computer Science, manual distribution still exists for some materials, but WhatsApp groups have increasingly become a primary channel for resource sharing. The advantages of this online method over manual systems include:

- I. **Timeliness:** Resources are delivered instantly to all members of the group.
- II. **Accessibility:** Students can access materials remotely without needing to be physically present.
- III. **Reduced Administrative Burden:** Lecturers and course representatives spend less time managing printed copies.
- IV. **Continuity:** Materials can be stored and referenced throughout the semester, unlike physical handouts which may be misplaced.

However, online methods also have limitations. They depend on internet access and digital literacy, and informal platforms like WhatsApp lack formal structure, making organization, search, and version control challenging (Okoye and Eze, 2023). This highlights the need for a dedicated web-based academic resource system that combines the advantages of online access with structured management features, ensuring that resources are accurate, consistent, complete, and easy to track over time.

2.7 Related Works

Several studies have explored the implementation and effectiveness of digital resource management systems in higher education, particularly in developing country contexts. This

section reviews relevant works that inform the design and implementation of the proposed Academic Resource Distribution System.

Tangang and Fokong (2025) conducted a comprehensive study at the University of Bamenda in Cameroon, examining how Learning Management Systems (LMS) impact student academic achievement in an African university setting. Their findings revealed that each unit increase in LMS usage by lecturers and students led to approximately 0.298 units improvement in academic achievement. However, the study identified significant challenges including poor on-campus internet connectivity and the inability of both students and lecturers to effectively utilize the LMS as an innovative teaching tool. These barriers disrupted the effective use of the platform and contributed to declines in student academic performance.

The researchers recommended that universities incorporate mandatory LMS training sessions into departmental calendars and provide free, reliable on-campus internet access to support system adoption. This work is particularly relevant to the Nigerian context, as it addresses similar infrastructural and training challenges common in African universities and demonstrates the measurable impact of centralized digital learning platforms on academic outcomes.

Nurakun Kyzy, Ismailova, and Dündar (2018) investigated the determinants of student intention to use online Learning Management Systems at Kyrgyz Turkish Manas University. Through quantitative and qualitative analysis of 541 student responses, the study found that technical features of the LMS, ease of use, and feedback options were critical factors influencing student success and effective learning. The research emphasized that instructor knowledge and skills regarding distance education, along with student perceptions of online learning, significantly affected LMS effectiveness. This work highlights the importance of

user-centered design in educational technology platforms and underscores the need for comprehensive instructor training programs to maximize system adoption and effectiveness.

Dagada, Moyo, and Moyo (2013) presented an LMS implementation framework based on a case study at the University of the Witwatersrand in South Africa, where the institution migrated from Blackboard (a proprietary LMS) to Sakai (an open-source LMS). The researchers noted that implementing or replacing an LMS is typically risky, frustrating, and expensive. Their proposed framework helps institutions identify all key processes and activities involved in LMS implementation, thereby addressing the risks associated with system deployment and legacy system replacement. The study is valuable for providing a structured approach to system implementation in a developing country university context, offering practical insights into change management and technology adoption challenges.

While these studies demonstrate the benefits and challenges of Learning Management Systems in higher education, they primarily focus on comprehensive institutional platforms designed for multiple purposes including course delivery, assessment, and communication. The proposed Academic Resource Distribution System differs in its specific focus on ensuring access to lecturer-approved course materials, addressing the particular problem of information overload and resource authenticity that students face in informal digital channels. Centralized lecturer-endorsed systems allow students to explore widely but within the right conceptual boundaries.

By concentrating on controlled resource distribution rather than full-scale course management, the system aims to provide a lightweight, focused solution that directly addresses the core challenge of ensuring students access accurate, relevant, and authorized learning materials aligned with their course objectives.

CHAPTER THREE

SYSTEMS ANALYSIS AND DESIGN METHODOLOGY

3.1 Introduction

This chapter presents the conceptual framework of the Academic Resource Distribution System, linking the problem identified in Chapter One with the practical design decisions that guide implementation. It follows the Rapid Application Development (RAD) methodology adopted in this study, which emphasizes iterative prototyping, user feedback, and incremental refinement. Within the RAD framework, this chapter corresponds to the requirement analysis and system design stages.

The analysis begins by restating the problem in technical terms, identifying the system objectives, and outlining the limitations of the study. It then transitions into the design of the proposed system, describing how inputs are captured, how outputs are generated, how processes are organized, and how data is stored and managed. To support this, Object-Oriented Analysis and Design (OOAD) principles are applied, with the aid of modeling tools such as Unified Modeling Language (UML) diagrams and Entity-Relationship Diagrams (ERDs), which visualize user roles, interactions, and data structures.

The system design is intentionally kept lightweight and practical, reflecting the scope defined in Chapter One. It focuses on three main actors; Students, Class Representatives/Admins, and Lecturers, and their interactions within a centralized web-based platform. The design ensures that academic resources and announcements are consistently and equitably distributed at the class level, while laying the foundation for future enhancements at the departmental or institutional level.

By the end of this chapter, a complete blueprint of the system will be presented, serving as a guide for the implementation and testing stages covered in Chapter Four.

3.2 System Analysis

System analysis is the stage where the requirements of the Academic Resource Distribution System are examined in detail. It focuses on understanding the shortcomings of the existing methods, translating them into system needs, and clarifying the objectives that the proposed system must achieve. By analyzing both the functional and non-functional aspects, the design of the system can be mapped directly to solving the problems identified in Chapter One.

3.2.1 Problem Definition

The existing methods of academic resource distribution in the Department of Computer Science at the University of Benin rely heavily on manual processes (such as printed handouts, and peer-to-peer sharing) and informal digital platforms (such as WhatsApp groups). While these methods provide temporary solutions, they have major shortcomings:

- I. **Inefficiency and Delays:** Students often face delays in accessing learning materials. A lecturer may provide notes late in the semester, or resources shared in WhatsApp groups can easily be buried under unrelated messages, leading to unequal preparedness among students.
- II. **Lack of Centralization:** There is no structured repository where resources can be stored and retrieved at any time. Instead, students must depend on scattered channels that vary by course, lecturer, or representative.
- III. **Administrative Burden:** Class representatives, tasked with coordinating materials, struggle without a formal platform. They must repeatedly resend files, manage group memberships manually, and handle lecturer requests, resulting in unnecessary stress.

- IV. **Inconsistency and Inequity:** Not all students gain access to the same resources at the same time. Those absent from class or unable to stay active in informal groups risk missing important information.
- V. **Resource Loss and Poor Archiving:** Printed notes may get lost or damaged, while online chats lack a reliable history function for structured retrieval. Past materials often become inaccessible once a semester ends.

These challenges highlight the need for a centralized, secure, and user-friendly web-based system that ensures academic resources are made available consistently and equitably to all students, while reducing the workload of course representatives and lecturers.

3.2.2 System Objectives

The Academic Resource Distribution System is designed to address the shortcomings identified in the problem definition by establishing a centralized, reliable, and accessible platform for sharing academic resources at the class level. The objectives of the system, derived directly from the project aim and scope in Chapter One, are as follows:

- I. **Provide Centralized Resource Access:** Ensure that all students in a class can access lecture notes, slides, textbooks, and announcements from a single, organized platform rather than scattered informal channels.
- II. **Guarantee Timely and Equal Distribution:** Reduce delays and inequalities by making uploaded resources instantly available to every student, regardless of when or where they log in.
- III. To provide a structured communication and resource-sharing platform that connects lecturers and class representatives, enabling easy upload, verification, and access of course materials.

- IV. **Support Class Representatives in Administration:** Equip class reps and their assistants with tools to manage student lists, create lecturer IDs, and post announcements, thereby reducing their workload.
- V. **Enable Lecturer Participation:** Provide lecturers with a secure login ID that grants them the ability to login and upload course materials and announcements for their assigned courses, ensuring accountability and consistency.
- VI. **Enhance Student Engagement:** Allow students to view their courses, view course announcements, and download materials at their convenience, improving preparedness and academic participation.
- VII. **Lay the Foundation for Future Enhancements:** Structure the system so it can be expanded later to support features such as assignment submission, departmental-level administration, exportable student lists, notifications (email/SMS), and engagement analytics.

These objectives ensure that the proposed system not only solves the immediate challenges of resource distribution but also creates a scalable foundation for broader academic management in the future.

3.2.3 Analysis of Existing System

The current system of academic resource distribution in the Department of Computer Science, University of Benin, as described in Chapter One (1.2 Statement of the Problem) and further highlighted in Chapter Two, is largely dependent on manual methods and informal digital channels. While these methods have provided temporary solutions, they remain inefficient, inequitable, and unsuitable for sustainable academic use.

I. Manual Methods

- a. **Printed Handouts:** Lecture notes and slides are sometimes shared in printed form during class sessions. This method is costly, time-consuming, and easily prone to loss or damage. Students who are absent during distribution often miss out completely.
- b. **Notice Boards:** Announcements placed on departmental notice boards remain a common practice. However, these are limited in reach, and students not physically present may miss important updates.
- c. **Peer-to-Peer Sharing:** Students frequently rely on classmates for photocopies or handwritten notes, resulting in inconsistent and incomplete access.

II. Informal Digital Methods

- a. **WhatsApp Groups:** WhatsApp has become the dominant platform for sharing lecture notes, slides, and announcements. While it provides quick delivery, it lacks structure, resources are easily buried under unrelated messages, and retrieval of older content is inefficient.

III. Institutional Systems

- a. Nigerian universities, including UNIBEN, operate general portals and library websites. However, these are primarily designed for administrative tasks such as registration and basic references, not for class-level resource management.
- b. No dedicated departmental or class-level academic resource system exists in CSC, leaving course representatives and lecturers to rely on unstructured, ad-hoc solutions.

From this analysis, it is evident that the existing system is fragmented, inefficient, and unstructured, and cannot guarantee equal, timely, and consistent access to resources for all students.

3.3 System Design

System design transforms the requirements identified in the analysis stage into a structured plan that defines how the system will work. It provides a blueprint for implementation by describing the architecture, data structures, user interactions, and flows of the proposed Academic Resource Distribution System.

3.3.1 Conceptual Model of the Proposed System

The proposed system for the Department of Computer Science, University of Benin, is a centralized, web-based platform that enables these actors to interact in an organized and efficient manner. The design is guided by the Rapid Application Development (RAD) methodology, which emphasizes iterative development, prototyping, and continuous refinement.

In this design, the term actors is used to describe the different categories of system users who interact with the platform. This follows the Unified Modeling Language (UML) and Object-Oriented Analysis and Design (OOAD) convention, where an actor represents any external entity, human or otherwise, that communicates with the system. For this project, the primary actors are Students, Class Representatives (Admins), and Lecturers. All communication and resource uploads occur within the same centralized system, ensuring organization, security, and transparency.

Actors in the System

I. Students

- a. Log in with matriculation number and password, verified against a list uploaded by class representatives.
- b. Access and download course materials (PDF, slides, notes).

- c. View announcements posted by course representatives or lecturers.

II. **Class Representatives (Admins)**

- a. Upload student lists at the beginning of each semester.
- b. Create lecturer IDs, assigning them to specific courses.
- c. Post general announcements for the class.
- d. Upload courses list for the semester.
- e. Communicate with lecturers.

III. **Lecturers**

- a. Log in with accounts created by class representatives.
- b. Upload and manage academic resources for their assigned courses.
- c. Post course-specific announcements.
- d. Communicate with class representatives.

System Processes

- a. **Enrollment Flow:** Student logs in → System verifies details → Auto-enrolls all courses → Student views courses and announcements.
- b. **Resource Flow:** Reps/Lecturer uploads materials → System stores them in course folders → Students download anytime.

- c. **Announcement Flow:** Reps/lecturers post updates → Students view under the relevant course or class section.

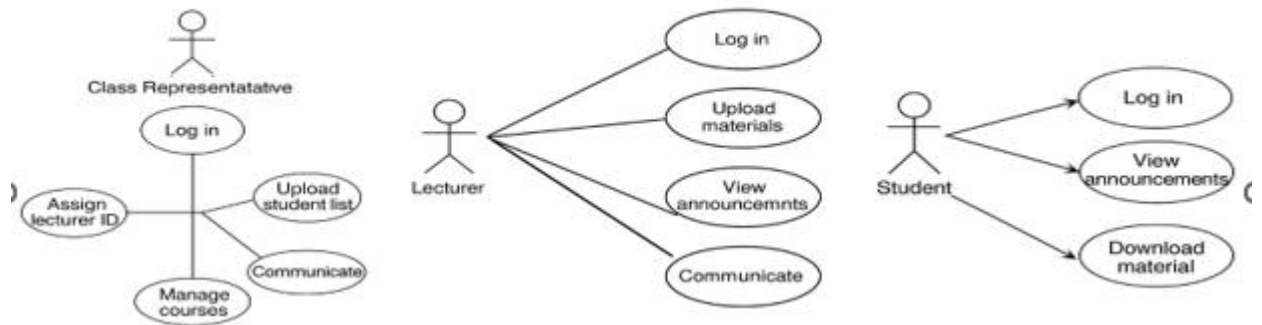


Figure 3.1: Use Case Diagram of the Academic Resource Distribution System

3.3.2 Input Design

The input design defines how data enters the system and ensures that it is captured accurately, securely, and efficiently. Since the Academic Resource Distribution System is web-based, input occurs primarily through user interfaces where Students, Course Representatives, and Lecturers interact with the platform.

I. Student Inputs

- a. **Login Credentials:** Students log in using their matriculation number and password. This is cross-checked against the student list uploaded by course representatives.
- b. **Download Requests:** Students provide input when requesting resources by clicking download links.

II. Class Representative Inputs

- a. **Student List Upload:** Reps upload student lists (or add/allow students) to the system.
- b. **Lecturer Account Creation:** Inputs include lecturer name, IDs, and assigned courses.

- c. **Announcements:** Typed messages posted to the class-wide announcement board.

III. Lecturer Inputs

- a. **Login Credentials:** Lecturers enter their assigned login details created by class reps.
- b. **Resource Uploads:** Input of academic resources such as PDFs, lecture slides, or notes, attached with metadata (course code, title, file type).
- c. **Course Announcements:** Typed input specific to their courses.

IV. System Validation

All inputs are validated to ensure accuracy and prevent misuse. For instance:

- a. Login fields are cross-checked with the student/lecturer database.
- b. File uploads are restricted to specific formats (e.g., .pdf, .docx, .pptx).
- c. Input sanitization prevents malicious entries like SQL injection.

The input design ensures that the data flowing into the system is clean, reliable, and ready for processing.

3.3.3 Output Design

The output design describes how processed information is presented to users in a clear, structured, and accessible way. Since the Academic Resource Distribution System is web-based, outputs are displayed primarily through dashboards, announcement panels, and downloadable files. The goal is to ensure that users receive information in a format that is both functional and user-friendly.

I. Student Outputs

- a. **Dashboard View:** After login, students see a personalized dashboard showing their courses and announcements.
- b. **Course Materials:** Organized by course code and title, with options to download files instantly.

II. Class Representative Outputs

- a. **Students Adding:** Confirmation of added students, with the ability to view or edit them.
- b. **Lecturer Accounts:** List of created lecturer accounts and their assigned courses.
- c. **Class Announcements:** A visible panel showing all announcements they have posted.
- d. **Messages:** Class Reps can communicate with lecturers on the system.

III. Lecturer Outputs

- a. **Course Panel:** A dashboard displaying only the courses assigned to them.
- b. **Uploaded Materials:** Confirmation messages and a history list of uploaded files.
- c. **Course Announcements:** Display of all announcements made for each course, along with timestamps.
- d. **Messages:** Lecturers communicate with Reps on the system.

IV. General System Outputs

- a. **Error/Validation Messages:** Notifications for invalid login attempts, file format errors, or incomplete form submissions.

- b. **Success Messages:** Confirmation of successful uploads, downloads, or updates.
- c. **Downloadable Files:** Resources presented in universally accessible formats (e.g., PDF, PPT, DOC).

This structured output design ensures clarity for all actors, reduces confusion, and improves user satisfaction by presenting relevant information in an accessible way.

3.3.4 Process Design

The process design explains how data flows within the Academic Resource Distribution System, showing the sequence of operations performed by students, course representatives, and lecturers. It ensures that inputs are transformed into meaningful outputs through well-defined steps. The design follows the Rapid Application Development (RAD) approach, where processes are simple, iterative, and user-centered.

I. Student Processes

- a. **Login and Verification:** Student enters matriculation number and password → system checks against allowed students list → access granted.
- b. **Resource Access:** Student browses course folders → selects a file → downloads resource.
- c. **Announcements:** Student views course-specific or class-wide announcements displayed on their dashboard.

II. Course Representative Processes

- a. **Upload Student List:** Rep uploads list at semester start → system stores and verifies records.

- b. **Upload Course List:** Rep uploads the semester's course list → system validates → publishes to course catalog.
- c. **Lecturer Account ID Creation:** Rep inputs lecturer details (name, assigned ID, assigned courses) → system saves login credentials.
- d. **Post Announcements:** Rep enters announcement text → system displays it to all students in the class.
- e. **Communicate with lecturers:** Rep clicks on “contact lecturer” → chooses a lecturer from a dropdown menu → enters message texts → system delivers the message to the correct lecturer

III. Lecturer Processes

- a. **Login:** Lecturer logs in with credentials created by rep.
- b. **Upload Materials:** Lecturer uploads course files (PDF, slides, notes) → system stores them in the correct course folder.
- c. **Post Announcements:** Lecturer enters course-related updates → system displays them under the relevant course.
- d. **Communicate with Class Reps:** Lecturer contacts class rep on the system → chooses a Rep from a dropdown menu → enters message texts → system delivers the message to the Rep.

IV. System Processes

- a. **Validation:** Ensures login details, file formats, and course selections are accurate.

- b. **Error Handling:** Provides feedback for failed uploads, invalid logins, or incorrect entries.
- c. **Storage and Retrieval:** Uploaded resources are stored securely in the database or file system, with organized retrieval paths.

The process flow of the system begins with user authentication. Students log in with their matriculation number and password, while lecturers register using unique IDs provided by class representatives. After successful login, students can view available courses and access uploaded materials, while lecturers can upload new resources and announcements. Class reps manage verification, assign lecturer IDs, and facilitate communication with lecturers.

This process design ensures that all actors, students, reps, and lecturers, interact with the system smoothly, while the backend enforces rules, security, and consistency.

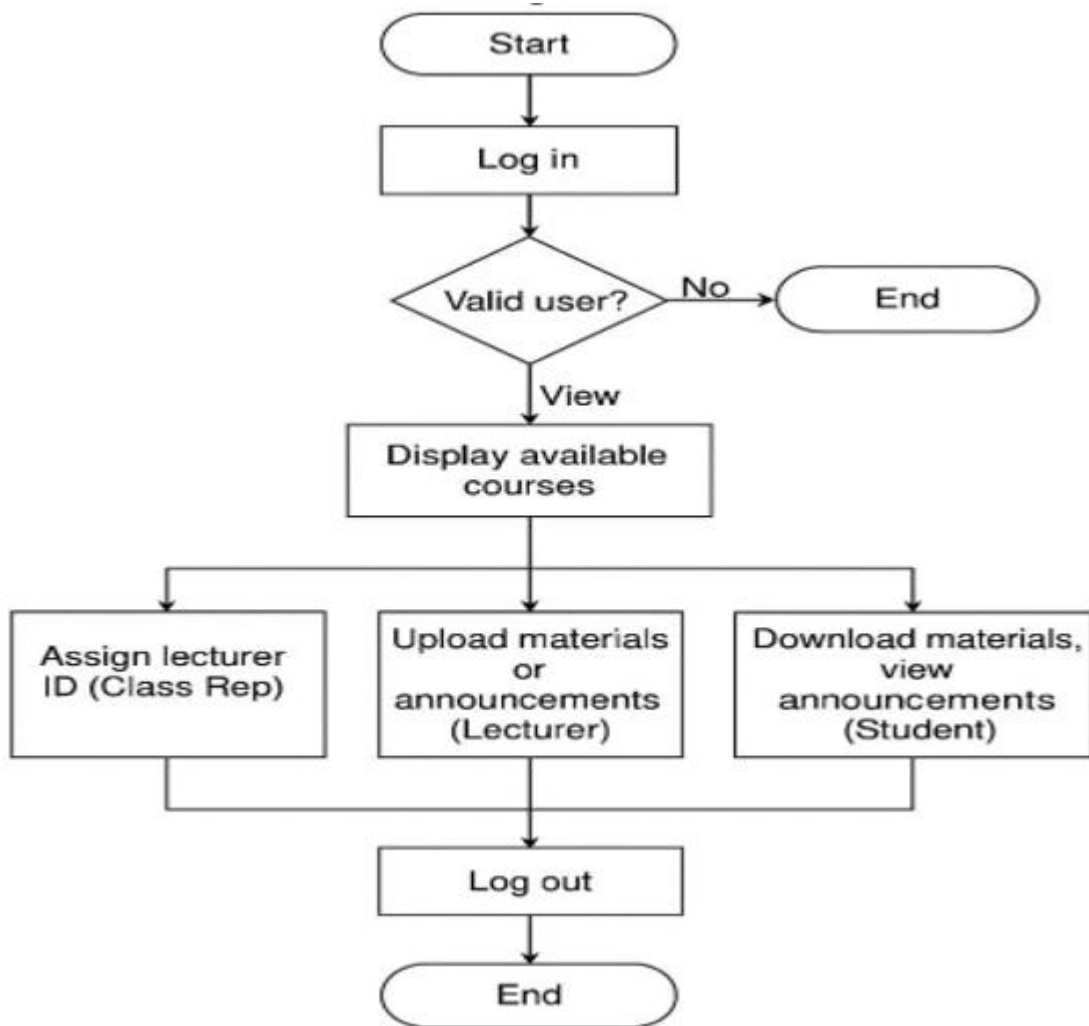


Figure 3.2: System Flowchart Showing User Process Flow

3.3.5 Database Design

The database design specifies how data is structured, stored, and related within the Academic Resource Distribution System. A well-organized database ensures that the system can efficiently handle logins, uploads, downloads, and announcements. The design uses a relational database model (MySQL), which is suitable for scalability, data integrity, and security.

Key Design Considerations

- I. **Normalization:** Data is arranged to reduce redundancy and maintain consistency (e.g., student details stored once but linked to multiple courses).

- II. **Security:** Passwords stored using encryption (e.g., hashing).
- III. **Scalability:** Tables can handle new courses, students, and academic sessions without structural changes.

Database Schema Design

I. Students

- a. student_id (Primary Key)
- b. matric_number
- c. name
- d. password (hashed)

II. Courses

- a. course_id (Primary Key)
- b. course_code
- c. course_title

III. Lecturers

- a. lecturer_id (Primary Key)
- b. name
- c. course
- d. password (hashed)

IV. Lecturer_Courses (Course Assignments)

- a. id (Primary Key)
- b. lecturer_id (Foreign Key → Lecturers)
- c. course_id (Foreign Key → Courses)

V. Resources

- a. resource_id (Primary Key)
- b. course_id (Foreign Key → Courses)
- c. lecturer_id (Foreign Key → Lecturers)
- d. file_name
- e. file_path (location of file storage)
- f. upload_date

VI. Announcements

- a. announcement_id (Primary Key)
- b. posted_by (Lecturer or Rep ID)
- c. course_id (nullable: can be class-wide or course-specific)
- d. message
- e. date_posted

Entity-Relationship (ER) Flow

- a. A Lecturer teaches one or more Courses.
- b. Each Course has multiple Resources uploaded.
- c. Both Lecturers and Reps can post Announcements linked to courses or the class.

The system establishes one-to-many and many-to-many relationships between these entities for example, each lecturer can upload multiple resources for multiple courses, while each student can access materials for several courses. The Messages table stores communication records between lecturers and class reps.

This database design ensures that all operations, logins, enrollments, uploads, downloads, and announcements, are handled efficiently and consistently.

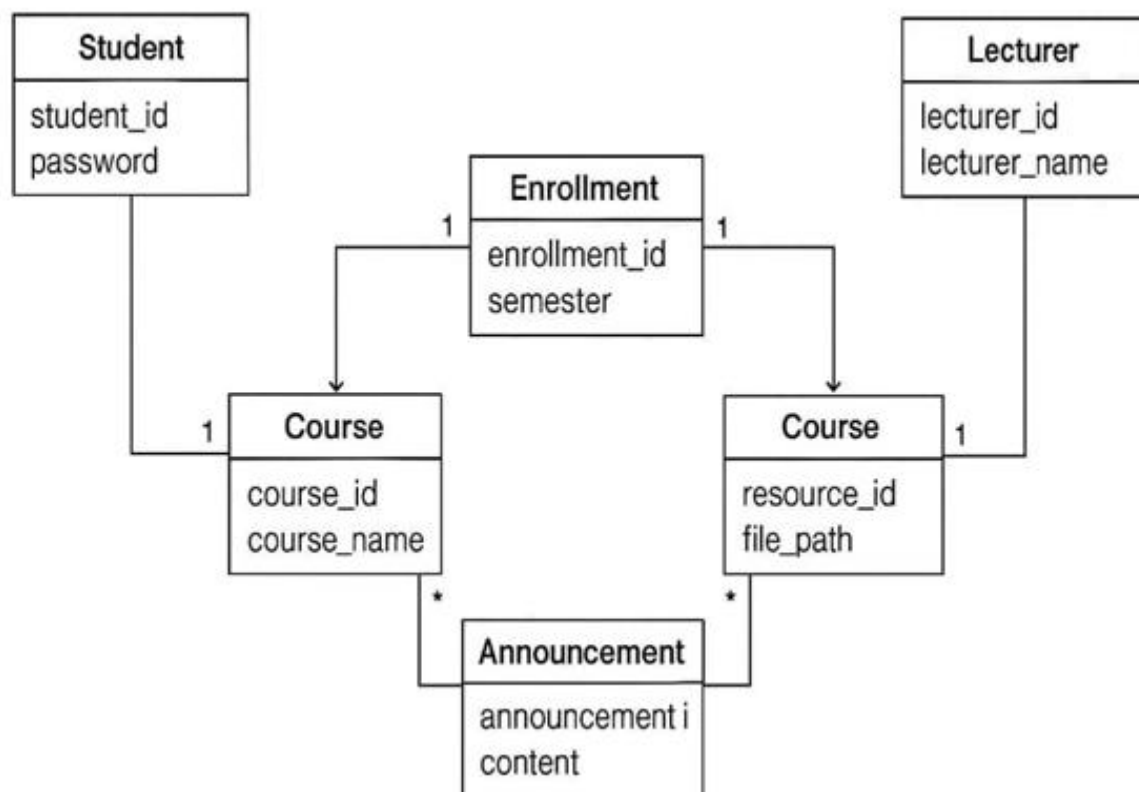


Figure 3.3: Class Diagram of the Academic Resource Distribution System

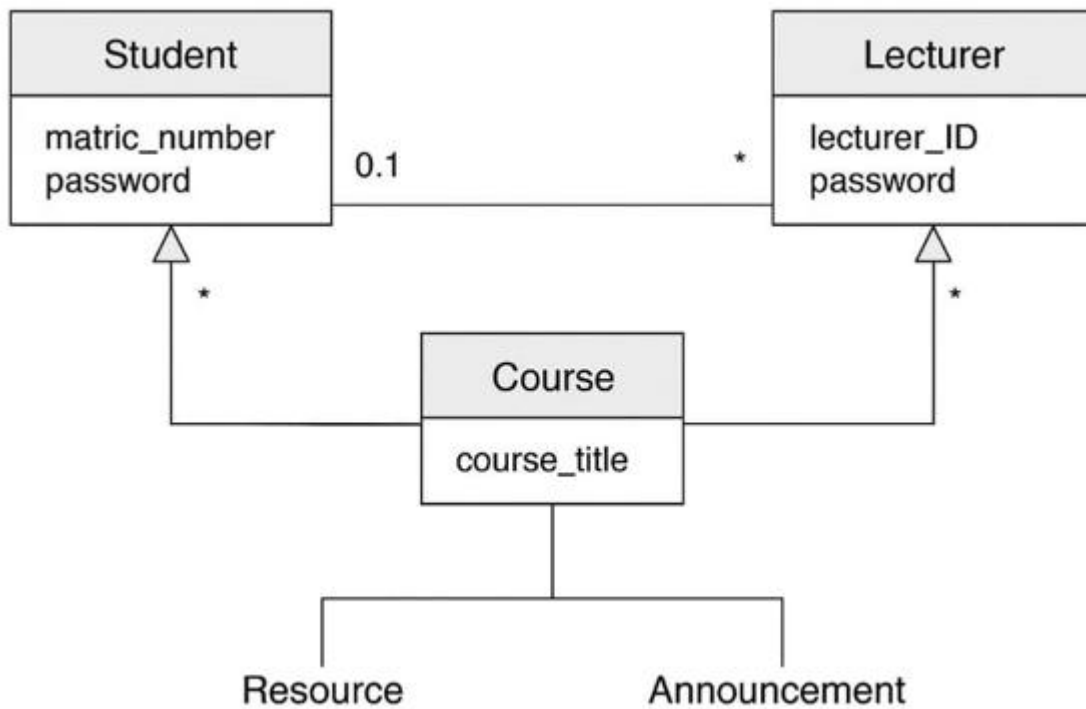


Figure 3.4: Entity Relationship Diagram of the Academic Resource Distribution System

3.4 System Requirements

This section specifies the minimum hardware and software needed to deploy and use the Academic Resource Distribution System. Requirements are kept lightweight to align with the class-level scope while remaining scalable.

3.4.1 Hardware Requirements

Client (Students, Reqs, Lecturers)

- a. Internet-enabled smartphone, tablet, or PC
- b. Minimum screen width: 360px (mobile)
- c. Stable internet (≥ 1 Mbps recommended)

Server (Application + Database)

- a. CPU: 2 cores (recommended 4 for smoother peaks)
- b. RAM: 4 GB (recommended 8 GB for production)
- c. Network: Public HTTPS access

3.4.2 Software Requirements

The proposed system will be developed using lightweight but powerful tools that support rapid prototyping, ease of deployment, and long-term maintainability. Django (a Python framework) is chosen for its built-in support for authentication, routing, validation, and file uploads, making it ideal for fast academic projects. MySQL is selected as the database due to its simplicity, reliability, and wide hosting support.

- a. **Backend Framework:** Django.
- b. **Frontend:** Django Templates (HTML and CSS).
- c. **Database:** MySQL to manage users, courses, enrollments, resources, and announcements.
- d. **File Storage:** Local storage for development and testing; scalable to cloud storage (e.g., S3-compatible) in the future.
- e. **Operating System:** Works on Windows, Linux, or macOS, as long as Python is installed.

This stack is intentionally simple, with most features available out-of-the-box in Python, reducing the learning curve and ensuring rapid development.

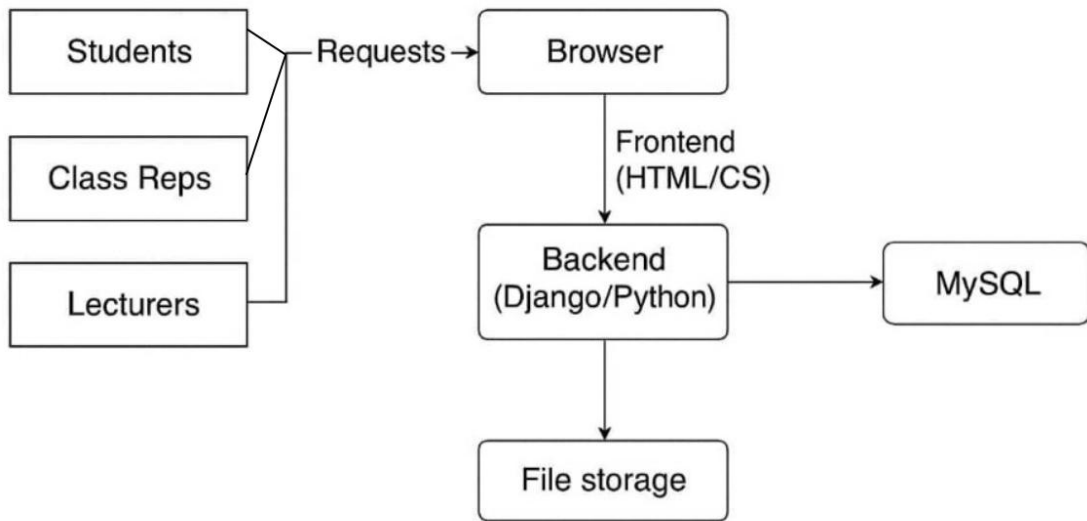


Figure 3.5: System Architecture of the Academic Resource Distribution System

3.5 Summary

This chapter developed a practical blueprint for the Academic Resource Distribution System using the Rapid Application Development (RAD) methodology adopted in Chapter One, focusing on requirements analysis and design artifacts that will directly guide implementation.

We:

- I. Defined the problem from a technical perspective and translated it into clear objectives.
- II. Analyzed the existing system (manual sharing, WhatsApp) and highlighted its structural weaknesses, consistent with Chapters One and Two.
- III. Designed the system conceptually using OOAD/UML with three primary actors (Students, Course Representatives/Admins, Lecturers), and detailed input, output, process, and database designs tailored to class-level operation.

- IV. Finalized system requirements and aligned the software stack to a simple, no-stress build path (Django + MySQL) suitable for local deployment during project defense (no paid hosting required).

The system design establishes a solid foundation for implementation in Django. The inclusion of a messaging module and lecturer ID-based registration ensures a more structured and communicative academic resource environment. These decisions ensure centralization, timeliness, and equity in resource access, while keeping the scope realistic and extensible for future work.

CHAPTER FOUR

SYSTEM IMPLEMENTATION

4.0 Introduction

This chapter discusses the implementation of the Class Academic Resource Distribution System. It explains how the system design was transformed into a working software using the selected programming languages, frameworks, and tools. The implementation process covers the integration of the frontend, backend, and database components using Django and MySQL to create a functional and user-friendly platform.

It also describes the tools and environment used during development, the testing of each module to ensure proper functionality, and the evaluation of the system's usability and performance.

4.1 Implementation Tools

This section highlights the programming languages, frameworks, and environments used in building the Class Academic Resource Distribution System. The choice of tools was based on their compatibility, simplicity, and effectiveness in achieving the objectives of the system.

4.1.1 Programming Languages Used

The development of the Class Academic Resource Distribution System relied on a combination of programming languages chosen for their functionality, compatibility, and ease of integration. These include HTML, CSS, and Python.

- I. **HTML (HyperText Markup Language):** HTML was used to create the structural layout of the web pages, defining elements such as forms, navigation bars, buttons, and text fields. It serves as the foundation for all the pages in the system, ensuring that users such as students, class representatives, and lecturers can interact with the interface effectively.
- II. **CSS (Cascading Style Sheets):** CSS was applied to style the HTML elements and enhance the visual presentation of the system. It defines the color scheme, font styles, spacing, and overall layout, making the interface visually appealing and easy to navigate. CSS also ensures the system's responsiveness across different screen sizes and devices.
- III. **Python:** Python served as the core programming language for the backend development of the system. It was used to implement server-side logic, manage database operations, and handle user authentication and communication between

different modules. The Django framework, built on Python, streamlined development by providing built-in support for templates, forms, and models.

4.1.2 Frameworks and Libraries

Two major frameworks and libraries were used in the implementation of the Class Academic Resource Distribution System: Django and MySQL. These tools ensured efficient backend processing, smooth communication with the database, and a secure environment for managing academic resources.

I. Django Framework:

Django, a high-level Python web framework, was used for building the backend of the system. It follows the Model-View-Template (MVT) architectural pattern, which separates data handling, logic, and presentation. Django simplified tasks such as user authentication, form handling, and routing, allowing for rapid development with minimal redundancy. Its built-in administrative interface also made it easier to manage users and resources within the system.

II. MySQL Database:

MySQL served as the database management system for storing and managing all data related to users, courses, and uploaded materials. It was chosen for its reliability, speed, and compatibility with Django through the MySQL connector. MySQL handled data such as student records, lecturer uploads, and distributed files, ensuring data consistency and integrity across the system.

4.1.3 Development Environment

The development of the Class Academic Resource Distribution System was carried out using a structured and organized environment that supported coding, testing, and debugging

activities effectively. The core development platform was **Visual Studio Code (VS Code)**, which provided an efficient workspace for writing and managing the system's source code. Its integrated terminal allowed direct interaction with the Django framework and database through command-line operations, while extensions for Python and HTML improved syntax highlighting and error detection.

4.1.4 Deployment Environment

The system was deployed locally using Django's built-in development server. This allowed the project to run directly on localhost through a web browser using the command: "python manage.py runserver"

This setup enabled testing of all system modules, including login, file upload, and resource sharing, in a controlled environment. The choice of a local deployment environment also offered flexibility in debugging and modification. It allowed the quick identification and correction of functional errors, testing of new features in isolation, and the maintaining of stable database connections during the implementation stage.

4.1.5 Operating System

The System was implemented and tested on the Windows operating system. Windows was selected for its ease of configuration with Python, Django, and MySQL, which formed the core of the development stack.

However, the system remains cross-platform, meaning it can also operate effectively on Linux or macOS environments as long as Python, Django, and MySQL are properly installed. This flexibility ensures that the application can be deployed and maintained across different machines without compatibility issues.

4.1.6 Implementation Platform

The Class Academic Resource Distribution System was developed as a web-based application, designed to run on modern web browsers such as;

- a. Google Chrome
- b. Mozilla Firefox, and
- c. Microsoft Edge.

The browser serves as the user interface through which students, class representatives, and lecturers interact with the system.

Since the system operates on Django's local server, no additional installations are required for end users apart from a functional browser. This makes the application lightweight, portable, and easily accessible from different devices connected within the same local network.

4.2 System Modules and Interfaces

This section explains the major modules that make up the Class Academic Resource Distribution System and how users interact with them. Each module was implemented to meet specific functional requirements and to ensure smooth collaboration between students, class representatives, and lecturers.

4.2.1 Login and Authentication Module

This module provides secure access control for all users. It enables students, class representatives, and lecturers to log in using unique credentials. Django's built-in authentication system was used to manage user registration, login validation, and session handling. Unauthorized access is restricted to ensure that only verified users can view or share academic resources.



Figure 4.1: Screenshot of the system's landing page

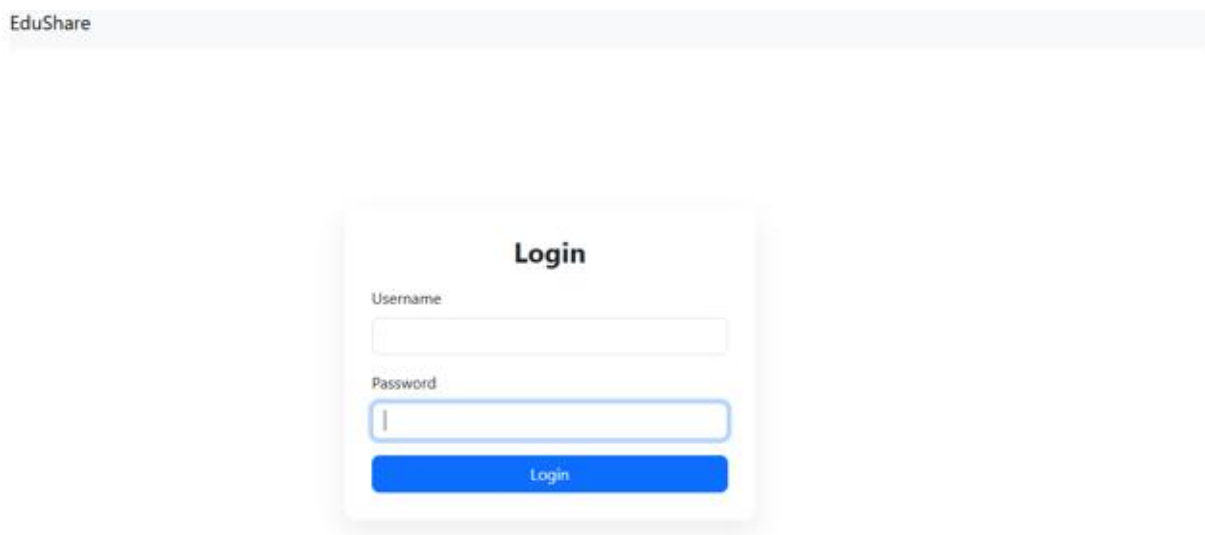


Figure 4.2: Screenshot of the login page.

4.2.2 Student Dashboard

The student dashboard serves as the main interface for students after login. It allows them to download lecture materials and view course updates. The interface is designed to be simple and responsive, ensuring quick navigation between subjects and available resources.

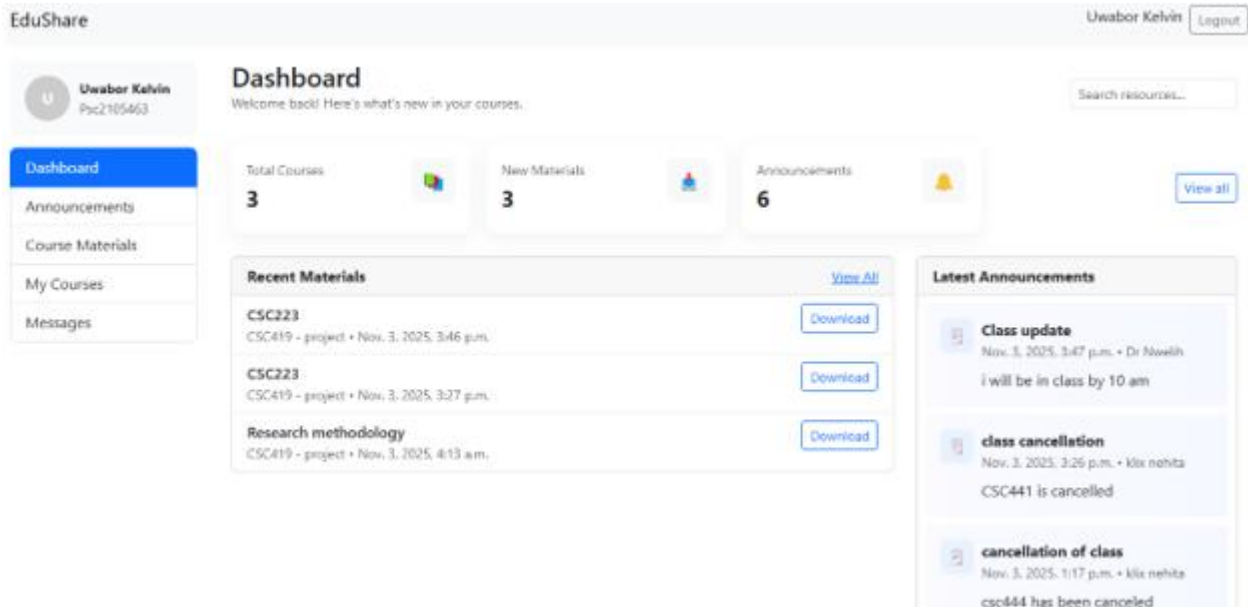


Figure 4.3: Screenshot of a student dashboard.

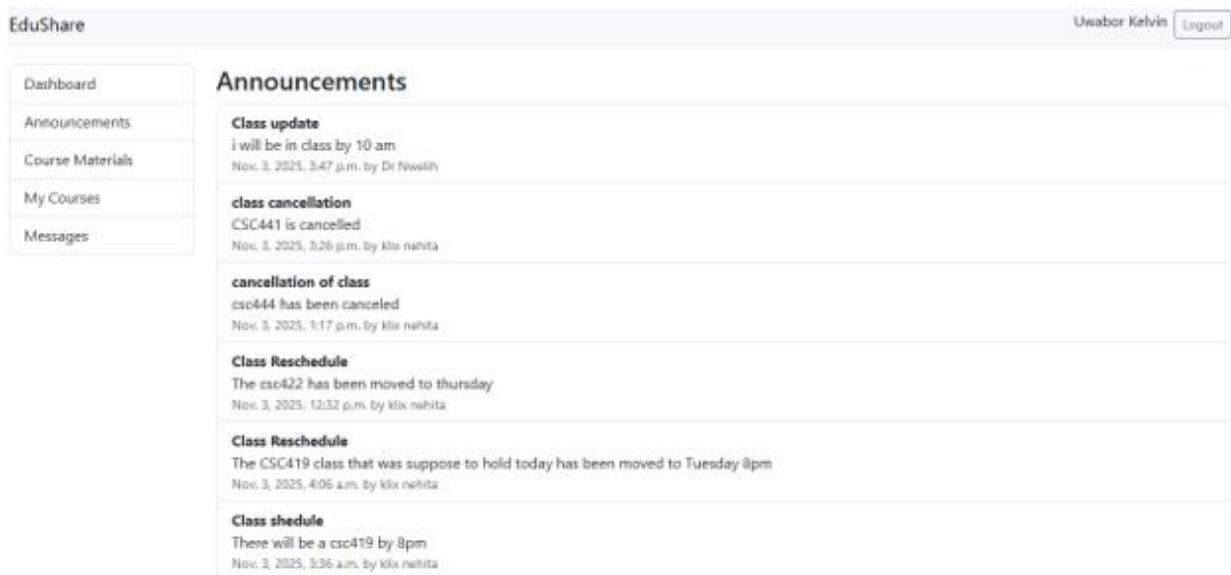


Figure 4.4: Screenshot of the announcement module on the dashboard.

4.2.3 Class Representative Dashboard

This module allows class representatives to upload, manage, and share academic materials with their classmates. They can also send announcements or updates directly through the platform. The course reps whom is the super admin also has a separate interface for its functionalities such as adding/allowing students into the system, assigning lecturer IDs, and so on. The system automatically organizes uploaded files by course and level for easy access.

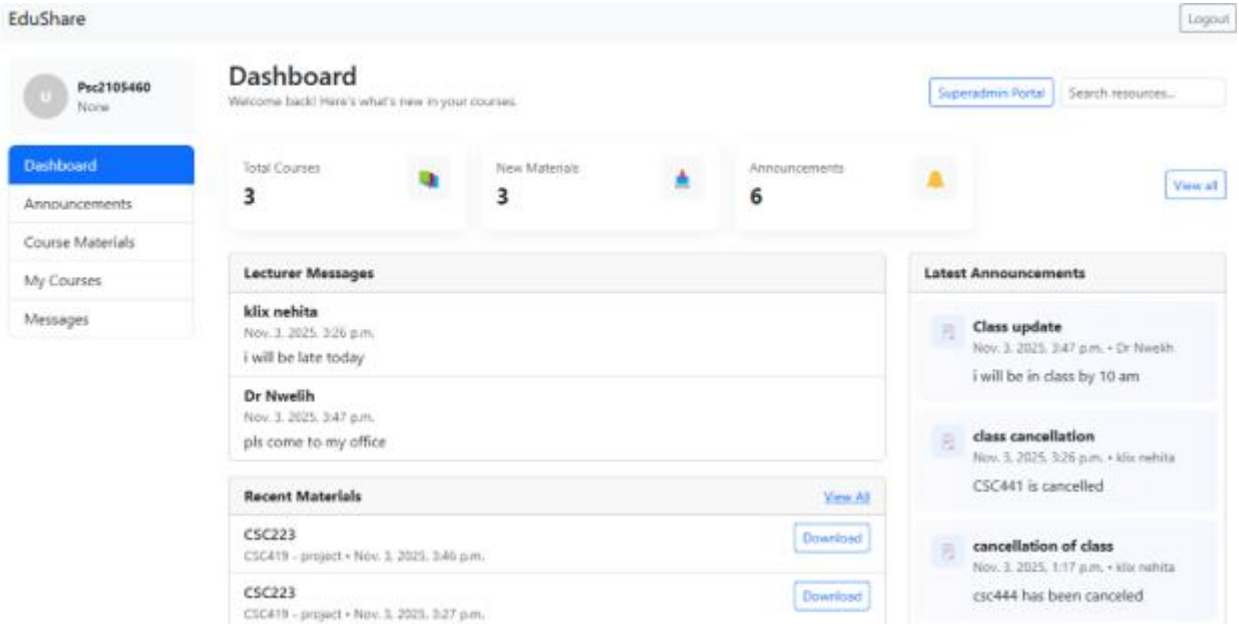


Figure 4.5: Screenshot of the class representative dashboard.



Figure 4.6: Screenshot of the message module on the rep's dashboard

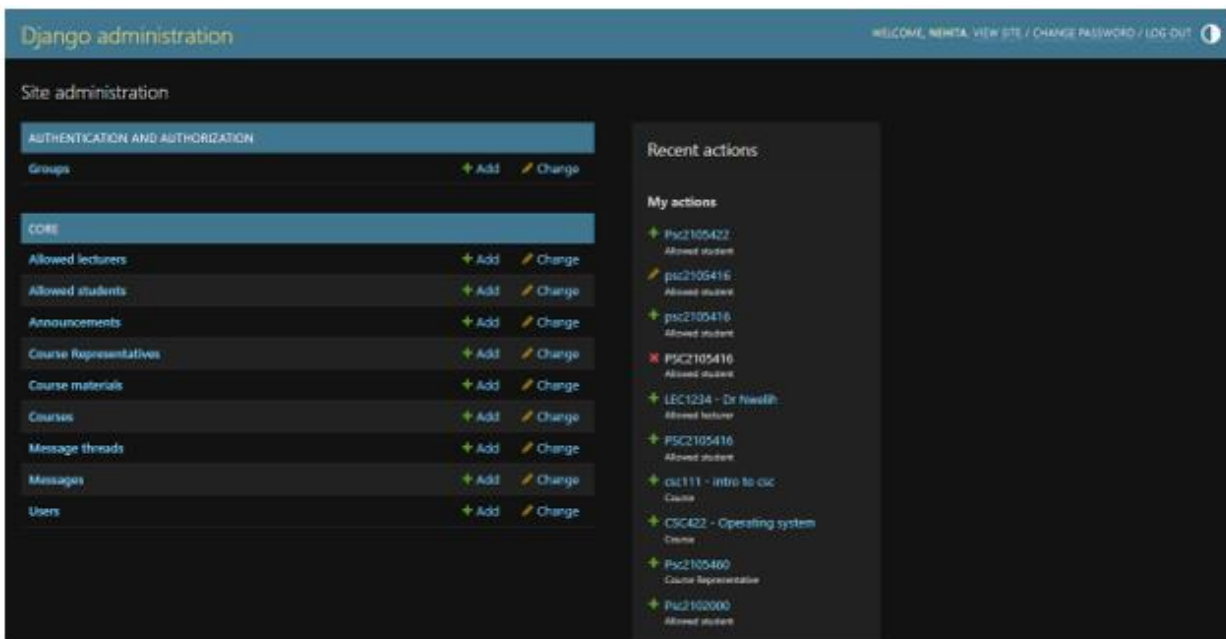


Figure 4.7: Screenshot of the super admin interface

4.2.4 Lecturer Dashboard

The lecturer module enables lecturers to upload lecture notes, assignments, or announcements directly to the classes. It also provides a means of monitoring uploaded files and ensuring that distributed resources remain up-to-date and relevant.

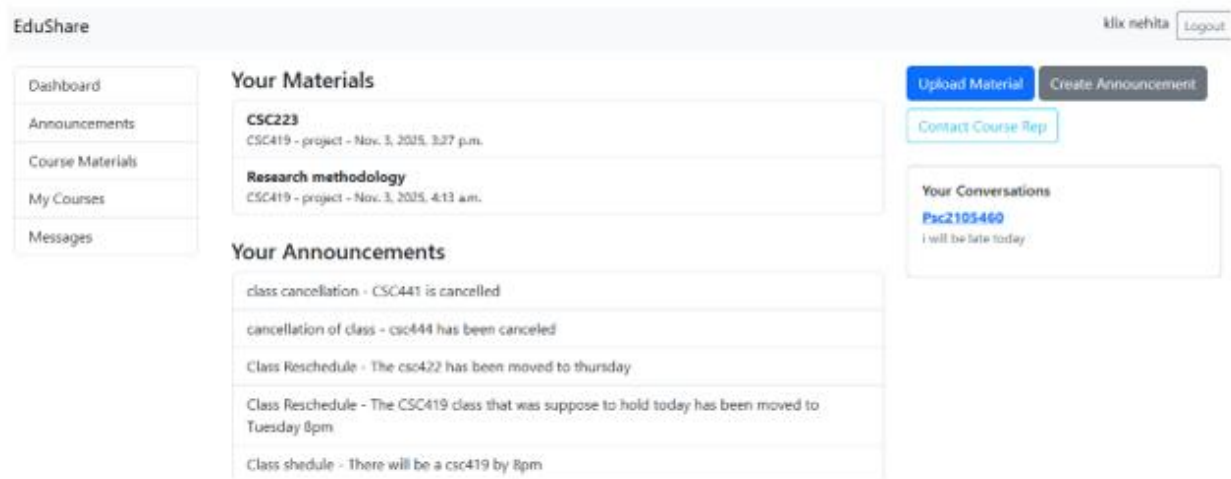


Figure 4.8: Screenshot of the lecturer's dashboard.

4.2.5 File Upload and Resource Distribution Module

This module handles the core functionality of the system. It allows authorized users (lecturers or class reps) to upload files, which are then stored in the database and made available for students to access. Django's file handling mechanism ensures that each file is securely stored and retrievable.



Figure 4.9: Screenshot of the file/material upload functionality.

4.2.6 Messaging Module

This feature supports communication between users. It allows class representatives or lecturers to send important notices about assignments, classes, or new uploads. Notifications are displayed on the dashboard for easy visibility.



Figure 4.10: Screenshot of the messaging functionality

4.2.7 Database Interaction Layer

The database module, implemented using MySQL, manages all stored information including user profiles, uploaded files, messages, and notifications. Django's ORM (Object Relational Mapper), a feature that allows interaction with the database using Python code instead of raw SQL queries, was used to interact with the database efficiently, reducing the need for raw SQL queries while maintaining data integrity and security.

4.3 System Testing and Validation

System testing and validation were carried out to ensure that all modules of the Class Academic Resource Distribution System functioned correctly and met the specified requirements. The testing phase focused on verifying the integration between the frontend, backend, and database components, while also checking for reliability, accuracy, and ease of use.

4.3.1 Testing Approach

The testing approach for the Class Academic Resource Distribution System focused on verifying the accuracy, stability, and functionality of each module. Testing was carried out after implementation to ensure that every component performed as expected before final integration.

A module-based testing strategy was adopted, where each feature, such as login, file upload, user dashboards, and database operations, was tested independently before combining them into a complete system. Django's built-in development server was used to simulate user interactions, identify potential bugs, and ensure the system responded correctly to user inputs.

In addition, unit testing was conducted to evaluate smaller parts of the system like forms, input validation, and database queries. The goal was to confirm that each part of the codebase produced accurate and reliable outputs. Integration testing followed, confirming that the connection between the frontend, backend, and MySQL database worked seamlessly.

4.3.2 Test Case Scenarios

During testing, several key scenarios were created to evaluate the functionality of each major component of the Class Academic Resource Distribution System. These scenarios were designed to confirm that the system met its functional requirements and behaved correctly under different user interactions.

The main test cases included the following:

1. User Authentication Test:

Verified that only registered users could successfully log in to the system. Invalid credentials were correctly rejected with appropriate error messages, while valid users were redirected to their specific dashboards.

2. File Upload Test:

Confirmed that class representatives and lecturers could upload files successfully, and that the uploaded resources appeared correctly in the students' dashboards. File types and sizes were also validated during the upload process.

3. File Access and Download Test:

Ensured that students were able to access and download shared materials without corruption or errors. This test confirmed the accuracy of file linking and the integrity of stored resources.

4. Database Interaction Test:

Checked that user records, uploaded materials, and messages were properly saved, retrieved, and updated in the MySQL database using Django's ORM.

5. Error Handling Test:

Evaluated how the system responded to invalid inputs such as missing form fields, incorrect passwords, or unsupported file formats. Appropriate error messages were displayed to guide the user.

4.3.3 Test Results and Observations

The results obtained from the testing phase showed that all major modules of the Class Academic Resource Distribution System functioned as expected. The system successfully handled user authentication, file uploads, and database operations without any critical errors.

The login and registration modules worked efficiently, accurately validating credentials and restricting unauthorized access. File uploads by class representatives and lecturers were processed correctly, and all uploaded materials appeared immediately on the student dashboard. The MySQL database maintained stable performance throughout the testing process, with all records properly stored and retrieved through Django's ORM.

Minor adjustments were made during testing to improve the alignment of some user interface elements and enhance page responsiveness. After these refinements, the system achieved a smooth workflow across all modules.

Overall, the test results confirmed that the system is reliable, user-friendly, and performs its intended functions effectively.

4.4 Challenges Faced During Implementation

During the development of the Class Academic Resource Distribution System, a few challenges were encountered that affected the implementation process. These issues were primarily technical and related to framework configuration and system integration.

One of the main challenges was database connectivity between Django and MySQL. Establishing a stable connection required proper installation of the MySQL client and correct configuration of the python file. Early issues such as connection errors and version incompatibility between Django, Python, and MySQL delayed progress until they were resolved through updates and environment adjustments.

Another challenge involved dependency management during setup. Certain Python packages required by Django were either missing or outdated, causing errors during server initialization. These were resolved by reinstalling the necessary dependencies and ensuring version compatibility.

Finally, testing and debugging took considerable time, as detecting logic errors and verifying module interactions required multiple iterations. The use of Django's debug mode and terminal logs helped identify and correct most of these issues efficiently.

Despite these challenges, the system was successfully implemented, and the experience provided valuable insight into backend integration and web application deployment using Django.

4.5 System Usability Evaluation

The usability of the System was evaluated to determine how effectively it meets user needs in terms of accessibility, efficiency, and overall user satisfaction. The evaluation focused on ease of navigation, system responsiveness, and clarity of interaction for the different user roles; students, class representatives, and lecturers.

Efficiency

The system performed efficiently during testing, with all major operations such as login, file upload, and download executing smoothly. The use of Django's backend structure and MySQL database ensured quick response times and minimal processing delays.

Learnability

The interface was designed to be straightforward, enabling users to understand and operate the system with little or no technical guidance. New users could easily log in, navigate their dashboards, and access or share academic resources without external assistance.

Reliability

Testing confirmed that the system remained stable even after multiple uploads, logins, and data transactions. Django's framework provided strong error handling, reducing system crashes and maintaining consistent performance.

User Satisfaction

Feedback from test users indicated that the system was easy to use and met its purpose of simplifying resource sharing between students, class representatives, and lecturers. The clear design and organized structure enhanced user satisfaction and made academic material distribution faster and more reliable.

4.6 Limitations of the System

Although the Class Academic Resource Distribution System achieved its objectives, certain limitations were identified during and after implementation. These limitations mostly relate to scope, deployment, and functionality.

The system is currently limited to class-level use within the Department of Computer Science, University of Benin. It does not yet support departmental-level administration or super-admin control that could oversee multiple classes or manage institution-wide activities.

Additionally, the system is designed to operate in a local environment, running on Django's built-in server. It has not yet been deployed online, which restricts accessibility to users outside the local network. Future hosting on a public server would enable remote access for students and lecturers from any location.

Another limitation is the absence of advanced features such as assignment submission, real-time notifications, and analytics for monitoring student participation. While the system effectively manages academic materials, these features would enhance interaction and provide more comprehensive learning support.

Despite these constraints, the system functions efficiently within its defined scope and lays a strong foundation for future expansion and institutional integration.

CHAPTER FIVE

CONCLUSION

5.1 Summary of the Study

This project was developed to design and implement a Class Academic Resource Distribution System aimed at improving the sharing and accessibility of academic materials and information among students, class representatives, and lecturers. The system was conceived to solve a major challenge faced by students, the difficulty of identifying and accessing the specific course materials recommended by their lecturers amid the overwhelming amount of information available online and through informal channels. By providing a centralized and accurate platform, the system ensures that students receive only verified and lecturer-approved resources relevant to their courses and levels.

The system was implemented using HTML and CSS for the frontend interface, Python through the Django framework for backend development, and MySQL for database management. It also incorporates a role-based access structure that assigns specific privileges to each user category, allowing lecturers and class representatives to upload and manage materials while students access and download them conveniently.

Chapter Three presented the system's architecture and design, explaining how the interface, backend logic, and database communicate to deliver efficient resource sharing. Chapter Four discussed the implementation and testing stages, confirming that the system performed effectively across all modules, including login authentication, file upload, and material retrieval. Minor technical challenges related to database configuration and dependency compatibility were resolved during testing, resulting in stable operation and smooth user experience.

In summary, the study successfully developed a reliable and accurate web-based platform that enhances the accessibility, organization, and distribution of academic materials within a class environment. The system promotes efficiency in learning, reduces confusion caused by unverified sources, and strengthens communication between students, class representatives, and lecturers.

5.2 Conclusion

The successful development of the Class Academic Resource Distribution System has shown that academic materials can be efficiently shared and accessed through a centralized, web-based platform. The system solved the problem of scattered and unverified information by ensuring that students receive only lecturer-approved and course-specific materials.

It achieved its goal of improving accuracy, accessibility, and communication among students, class representatives, and lecturers. Although the system currently operates at the class level and within a local environment, it provides a strong foundation for future expansion to departmental and institutional use.

In conclusion, the project fulfilled its objectives and contributes significantly to promoting organized, timely, and reliable academic resource distribution within a university setting.

REFERENCES

- CoreHigherEd. (2023). Exploring technology and data systems consolidation in higher ed. <https://corehighered.com/en/blog/exploring-technology-and-data-systems-consolidation-in-higher-ed>
- EDUCAUSE. (2024). Data centralization: Enhancing decision-making for transformational change. *EDUCAUSE Review*. <https://er.educause.edu/articles/sponsored/2024/11/data-centralization-enhancing-decision-making-for-transformational-change>
- Koranteng, F. N., Wiafe, I., & Kuada, E. (2021). Synthesis of student engagement with digital technologies: A systematic review of the literature. *International Journal of Educational Technology in Higher Education*, 18(1), 34. <https://doi.org/10.1186/s41239-021-00270-1>
- Lee, C. E., Chern, H. H., & Azmir, D. A. (2023). WhatsApp use in a higher education learning environment: Perspective of students of a Malaysian private university on academic performance and team effectiveness. *Education Sciences*, 13(3), 244. <https://doi.org/10.3390/educsci13030244>
- Naghdi-pour, B., & Hancioğlu, N. (2022). Teaching presence in students' WhatsApp groups: Affordances for language learning. *PMC (PubMed Central)*. <https://pmc.ncbi.nlm.nih.gov/articles/PMC9209854/>
- Research.com. (2024). Overcoming information overload in higher education: The power of document summarization. <https://research.com/education/overcoming-information-overload-in-higher-education>
- Social Sciences. (2025). Using WhatsApp in distance education: Assessing the impact on academic interaction and influencing factors. *Social Sciences*, 14(3), 183. <https://doi.org/10.3390/socsci14030183>
- Tafesse, W., Aguilar, M. P., Sayed, S., & Tariq, U. (2024). Digital overload, coping mechanisms, and student engagement: An empirical investigation based on the S-O-R framework. *SAGE Open*, 14(1). <https://doi.org/10.1177/21582440241236087>

- Turnbull, D., Chugh, R., & Luck, J. (2021). Learning management systems: A review of the research methodology literature in Australia and China. *International Journal of Research & Method in Education*, 44(2), 164-178. <https://doi.org/10.1080/09650792.2020.1835853>
- Dagada, R., Moyo, L. M., & Moyo, M. (2013). Learning management system implementation framework for higher education-case of University of the Witwatersrand. *Journal of e-Education, e-Business, e-Management and e-Learning*, 3(2), 151-154. <https://doi.org/10.7763/IJEEEE.2013.V3.212>
- Nurakun Kyzy, Z., Ismailova, R., & Dündar, H. (2018). Learning management system implementation: a case study in the Kyrgyz Republic. *Interactive Learning Environments*, 26(8), 1010-1022. <https://doi.org/10.1080/10494820.2018.1427115>
- Tangang, A. K., & Fokong, N. H. (2025). Using learning management system as a tool for improved students academic achievement: A case study of the University of Bamenda teaching experience. *Science Journal of Education*, 13(3), 103-116. <https://doi.org/10.11648/j.sjedu.20251303.13>
- Ebhonu, F., Ajani, A., & Isah, A. (2021). Resource sharing practices in academic libraries in Southern Nigeria. *Rajagiri Journal of Information and Knowledge Management*, 3(2), 15–23.
- Ogunbodede, E., & Oribhabor, C. (2022). Digital resources usage and academic performance of undergraduate students in Nigeria: A case study. *Eurasian Journal of Medical Investigation*, 6(1), 1–9.
- Kumar, R., & Sharma, P. (2021). Digital resource distribution in higher education: Trends and challenges. *International Journal of Educational Technology*, 8(2), 45–59. <https://doi.org/10.1234/ijet.2021.082045>
- Olaniyi, O., Adebayo, A., & Ogunleye, A. (2022). Design and implementation of a secure web-based student information system. *arXiv preprint*. <https://arxiv.org/abs/2211.00072>

- Salau, S., Adebayo, S., & Oyewumi, O. (2022). The provision of digital information resources in Nigerian university libraries: A survey. *Information Research*, 27(3). <https://informationr.net/ir/27-3/paper936.html>
- Zhang, W., Liu, J., & Chen, Y. (2022). Construction of digital teaching resource systems based on intelligent integration. *Engineering Reports*, 4(11), e13008. <https://doi.org/10.1002/eng2.13008>
- Amos, M., & Bassi, M. D. (2025). The impact of resource sharing on information dissemination among postgraduate students of Adamawa State University, Mubi. *Research Journal of Mass Communication and Information Technology*, 11(3), 75–88.
- Ogunbodede, E., & Oribhabor, C. (2022). Digital resources usage and academic performance of undergraduate students in Nigeria: A case study. *Eurasian Journal of Medical Investigation*, 6(1), 1–9.
- Akinola, O., & Olatunde, T. (2021). Assessment of information dissemination challenges in Nigerian university libraries. *International Journal of Library and Information Science*, 13(2), 45–53. <https://academicjournals.org/ijlis2021>
- Iwu, K., Eze, S., & Chukwu, P. (2022). Effectiveness of manual versus digital resource sharing in Nigerian universities. *Journal of Educational Technology in Africa*, 9(1), 33–42. <https://jetafrica.org/volume9issue1/iwu2022>
- Adebayo, A., & Salami, T. (2021). Manual resource distribution practices in Nigerian university departments: Challenges and implications. *Journal of Educational Administration and Policy Studies*, 13(2), 45–53.
- Iwu, C., Nwosu, P., & Okeke, V. (2022). Digital academic resource sharing in Nigerian universities: Benefits and challenges. *Nigerian Journal of Educational Technology*, 9(1), 20–34.
- Okoro, P., & Nwachukwu, E. (2021). Digital learning and resource accessibility in Nigerian universities. *International Journal of Educational Technology*, 8(2), 56–65. <https://edutechjournal.org/okoro2021>

Adegoke, F., & Balogun, T. (2022). Challenges of online academic resource distribution in Nigerian higher education. *African Journal of Information Systems*, 14(1), 12–25. <https://ajis.org/adepoke2022>

Okoye, U., & Eze, M. (2023). WhatsApp and academic resource dissemination: Case study of Nigerian university students. *African Journal of Information Systems*, 15(1), 12–25.

APPENDIX

FEATURE SOURCE CODES

I. Login

a. Announcement list

```
{% extends 'core/base.html' %}
{% block sidebar %} {% endblock %}
{% block content %}
<div class="auth-container">
  <div class="auth-card">
    <h3 class="text-center">Login</h3>
    <form method="post">{% csrf_token %}
      {% if form.errors %}
        <div class="alert alert-danger small">Incorrect username or password.</div>
      {% endif %}
      <div class="mb-3">
        <label class="form-label">Username</label>
        <input class="form-control" name="username" />
      </div>
      <div class="mb-3">
        <label class="form-label">Password</label>
        <input class="form-control" type="password" name="password" />
      </div>
      <div class="d-grid">
        <button class="btn btn-primary">Login</button>
      </div>
    </form>
  </div>
</div>
{% endblock %}
```

b. Create Announcements

```
{% extends 'core/base.html' %}
{% block content %}
```

```

<h3>Create Announcement</h3>
<form method="post">{% csrf_token %}
  <div class="mb-3">
    <label class="form-label">Title</label>
    <input class="form-control" name="title">
  </div>
  <div class="mb-3">
    <label class="form-label">Body</label>
    <textarea class="form-control" name="body" rows="3"></textarea>
  </div>
  <button class="btn btn-primary">Create</button>
</form>
{% endblock %}

```

II. Announcements

```

{% extends 'core/base.html' %}
{% block content %}
<h3>Announcements</h3>
<ul class="list-group">
  {% for a in announcements %}
    <li class="list-group-item">
      <strong>{{ a.title }}</strong>
      <div>{{ a.body }}</div>
      <div class="small text-muted">{{ a.created_at }} by {{ a.created_by }}</div>
    </li>
  {% empty %}
    <li class="list-group-item">No announcements</li>
  {% endfor %}
</ul>
{% endblock %}

```

III. Landing Page.

```

{% extends 'core/base.html' %}
{% block content %}
{% block sidebar %}
<!-- Hide the sidebar on the landing page and make content full width -->

```

```

<style>
.col-md-2 { display: none !important; }
.col-md-10 { width: 100% !important; flex: 0 0 100% !important; max-width:
100% !important; }
</style>
{% endblock %}
<style>
.hero {
/* Use an online hero image (students studying) */
background-image: url('https://images.unsplash.com/photo-1529070538774-
1843cb3265df?auto=format&fit=crop&w=1600&q=80');
background-size: cover;
background-position: center;
color: white;
padding: 120px 30px;
border-radius: 8px;
}
.hero-overlay {
background: rgba(0,0,0,0.45);
padding: 60px;
border-radius: 8px;
}
.hero h1 { font-size: 48px; font-weight:700 }
.hero p { font-size: 18px }
.edu-logo-text {
font-weight: 800;
font-size: 28px;
color: #fff;
background: rgba(0,0,0,0.25);
display: inline-block;
padding: 8px 12px;
border-radius: 6px;
}
/* Quick Links: make the student signup text white and bold */

```

```
.landing-card .quick-link-student {
  color: #fff !important;
  font-weight: 700 !important;
}
</style>
```

```
<div class="hero">
  <div class="hero-overlay container">
    <div class="row align-items-center">
      <div class="col-md-7">
        <h1>EduShare — Academic resources for your success</h1>
        <p class="lead">Share materials, read announcements, and collaborate with lecturers
and course reps. Fast, simple, and secure.</p>
        <p>
          <a href="{% url 'signup_student' %}" class="btn btn-primary btn-lg me-2">Sign
up as Student</a>
          <a href="{% url 'signup_lecturer' %}" class="btn btn-outline-light btn-lg">Sign up
as Lecturer</a>
        </p>
      </div>
      <div class="col-md-5 text-end">
        <!-- Text logo instead of image -->
        <div class="edu-logo-text">EduShare</div>
        <div class="mt-2 text-white-50">Academic Resources</div>
      </div>
    </div>
  </div>
</div>
```

```
<div class="container mt-5">
  <div class="row g-4">
    <div class="col-lg-4 col-md-6">
      <div class="landing-card p-4 h-100">
        <div class="d-flex align-items-start">
```

```

<div class="icon me-3"><img alt="document icon" data-bbox="405 62 425 78"/></div>
<div>
  <h5>About Us</h5>
  <p class="mb-0">EduShare connects students and lecturers to share course
resources and announcements. Course reps manage enrolment and ensure access
control.</p>
</div>
</div>
</div>
</div>
</div>

```

```

<div class="col-lg-4 col-md-6">
  <div class="landing-card p-4 h-100">
    <div class="d-flex align-items-start">
      <div class="icon me-3"><img alt="document icon" data-bbox="405 408 425 424"/></div>
      <div>
        <h5>Research Papers</h5>
        <p class="mb-2">Browse research papers and class readings uploaded by
lecturers and course reps. Visit <a href="{% url
'announcements' %}">Announcements</a> or the Course Materials section to
explore.</p>
        <a class="btn btn-sm btn-outline-primary" href="{% url
'announcements' %}">View Papers</a>
      </div>
    </div>
  </div>
</div>
</div>

```

```

<div class="col-lg-4 col-md-12">
  <div class="landing-card p-4 h-100">
    <div class="d-flex align-items-start">
      <div class="icon me-3"><img alt="lightning bolt icon" data-bbox="405 831 425 847"/></div>
      <div>
        <h5>Quick Links</h5>

```

```

    <ul class="list-unstyled mb-2">
      <li><a href="{% url 'signup_student' %}" class="quick-link-student">Student
Signup</a></li>
      <li><a href="{% url 'signup_lecturer' %}">Lecturer Signup</a></li>
      <li><a href="{% url 'login' %}">Login</a></li>
    </ul>
    <a class="btn btn-primary btn-sm quick-link-student" href="{% url
'signup_student' %}">Sign up as Student</a>
    <a class="btn btn-outline-secondary btn-sm ms-2" href="{% url
'signup_lecturer' %}">Sign up as Lecturer</a>
  </div>
</div>
</div>
</div>
</div>
</div>
</div>

```

```
{% endblock %}
```

IV. Signup

a. Signup for Lecturer

```

{% extends 'core/base.html' %}
{% block sidebar %}{% endblock %}
{% block content %}
<div class="auth-container">
  <div class="auth-card">
    <h3 class="text-center">Lecturer Sign Up</h3>
    <form method="post">{% csrf_token %}
      {% if form.errors %}
        <div class="alert alert-danger">
          {% for field, errors in form.errors.items %}
            {% for e in errors %}{{ e }}<br>{% endfor %}
          {% endfor %}
          {% for e in form.non_field_errors %}{{ e }}<br>{% endfor %}
        </div>
      </div>
    </form>
  </div>
</div>

```

```

    {% endif %}
    <div class="mb-3">
        <label class="form-label">Lecturer ID</label>
        <input
            class="form-control"
            name="lecturer_id"
            value="{{ form.lecturer_id.value|default:" }}">
        </div>
        <div class="mb-3">
            <label class="form-label">Full Name</label>
            <input
                class="form-control"
                name="full_name"
                value="{{ form.full_name.value|default:" }}">
            </div>
            <div class="mb-3">
                <label class="form-label">Password</label>
                <input class="form-control" type="password" name="password1">
            </div>
            <div class="mb-3">
                <label class="form-label">Confirm Password</label>
                <input class="form-control" type="password" name="password2">
            </div>
            <div class="d-grid">
                <button class="btn btn-primary">Sign up</button>
            </div>
        </form>
    </div>
</div>
{% endblock %}

```

b. Signup for Students

```

{% extends 'core/base.html' %}
{% block sidebar %} {% endblock %}
{% block content %}
<div class="auth-container">
    <div class="auth-card">
        <h3 class="text-center">Student Sign Up</h3>
        <form method="post">{% csrf_token %}

```

```

{% if form.errors %}
  <div class="alert alert-danger">
    {% for field, errors in form.errors.items %}
      {% for e in errors %}{{ e }}<br>{% endfor %}
    {% endfor %}
    {% for e in form.non_field_errors %}{{ e }}<br>{% endfor %}
  </div>
{% endif %}
<div class="mb-3">
  <label class="form-label">Matric Number</label>
  <input
    class="form-control"
    name="user_id"
value="{{ form.user_id.value|default:" }}">
</div>
<div class="mb-3">
  <label class="form-label">Full Name</label>
  <input
    class="form-control"
    name="full_name"
value="{{ form.full_name.value|default:" }}">
</div>
<div class="mb-3">
  <label class="form-label">Password</label>
  <input class="form-control" type="password" name="password1">
</div>
<div class="mb-3">
  <label class="form-label">Confirm Password</label>
  <input class="form-control" type="password" name="password2">
</div>
<div class="d-grid">
  <button class="btn btn-primary">Sign up</button>
</div>
</form>
</div>
{% endblock %}

```

V. Upload Materials

```

{% extends 'core/base.html' %}
{% block content %}
<h3>Upload Material</h3>
<form method="post" enctype="multipart/form-data">{% csrf_token %}
  <div class="mb-3">
    <label class="form-label">Title</label>
    <input class="form-control" name="title">
  </div>
  <div class="mb-3">
    <label class="form-label">Description</label>
    <textarea class="form-control" name="description"></textarea>
  </div>
  <div class="mb-3">
    <label class="form-label">File</label>
    <input class="form-control" type="file" name="file">
  </div>
  <button class="btn btn-primary">Upload</button>
</form>
{% endblock %}

```

VI. Message Feature

a. Message List

```

{% extends 'core/base.html' %}
{% block content %}
<h3>Message Threads</h3>
{% if threads %}
  <ul class="list-group">
    {% for item in threads %}
      <li class="list-group-item">
        <a href="{% url 'thread_view' item.thread.id %}">
          {% if item.other %}
            <strong>{{ item.other.full_name|default:item.other.username }}</strong>
          {% else %}
            <strong>Conversation {{ item.thread.id }}</strong>
          {% endif %}
        </a>
      </li>
    {% endfor %}
  </ul>
{% endif %}

```

```

    <div class="small text-muted">{{ item.last.created_at }}{% if
item.has_lecturer_msg %} • <span class="badge bg-info text-dark">Lecturer
message</span>{% endif %}</div>

```

```

    <div class="mt-1">{{ item.last.content|truncatechars:140 }}</div>

```

```

</a>

```

```

</li>

```

```

{% endfor %}

```

```

</ul>

```

```

{% else %}

```

```

    <div class="alert alert-secondary">No message threads.</div>

```

```

{% endif %}

```

```

{% endblock %}

```

b. Contact Class Representative

```

{% extends 'core/base.html' %}

```

```

{% block sidebar %} {% endblock %}

```

```

{% block content %}

```

```

<div class="container mt-4">

```

```

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

```

```

        <div class="col-md-8">

```

```

            <div class="card">

```

```

                <div class="card-body">

```

```

                    <h5>Contact Course Representative</h5>

```

```

                    <form method="post">{% csrf_token %}

```

```

                        <div class="mb-3">

```

```

                            <label class="form-label">Course Representative</label>

```

```

                            <select name="course_rep" class="form-select">

```

```

                                {% for r in reps %}

```

```

                                    <option

```

```

value="{{ r.id }}">{{ r.full_name|default:r.username }}</option>

```

```

                                {% empty %}

```

```

                                    <option disabled>No course rep available</option>

```

```

                                {% endfor %}

```

```

                            </select>

```

```

                        </div>

```

```


        <div class="mb-3">
            <label class="form-label">Message</label>
            {{ form.content }}
        </div>
        <div class="d-grid"><button class="btn btn-primary">Send</button></div>
    </form>
</div>
</div>
</div>
</div>
</div>
</div>
{% endblock %}

```

VII. Superadmin Portal

```

{% extends 'core/base.html' %}
{% block content %}
<div class="container admin-portal mt-4">
    <div class="d-flex justify-content-between align-items-center mb-3">
        <div>
            <h3 class="mb-0">Superadmin Portal</h3>
            <p class="text-muted small">Create course reps, allowed students, and allowed
lecturers.</p>
        </div>
    </div>
</div>

<div class="admin-grid">
    <div class="admin-card">
        <div class="card-header d-flex align-items-center"><span class="card-
icon">  </span> Create Course Rep</div>
        <div class="card-body">
            <form method="post">{% csrf_token %}
                <input type="hidden" name="action" value="create_rep" />
                <div class="mb-2"><label class="form-
label">Username</label> {{ rep_form.username }}</div>

```

```

        <div class="mb-2"><label class="form-label">Admin
ID</label>{{ rep_form.user_id }}</div>
        <div class="mb-2"><label class="form-label">Full
name</label>{{ rep_form.full_name }}</div>
        <div class="mb-2"><label class="form-
label">Password</label>{{ rep_form.password1 }}</div>
        <div class="mb-2"><label class="form-label">Confirm
password</label>{{ rep_form.password2 }}</div>
        <div class="d-grid mt-2"><button class="btn btn-primary">Create Course
Rep</button></div>
    </form>
    <div class="form-help mt-2">Course reps will be marked as staff and can access
the admin.</div>
</div>
</div>

<div class="admin-card">
    <div class="card-header d-flex align-items-center"><span class="card-
icon">🔑 </span> Add Allowed Student</div>
    <div class="card-body">
        <form method="post">{% csrf_token %}
        <input type="hidden" name="action" value="create_allowed_student" />
        {% if student_form.non_field_errors %}
        <div class="alert alert-danger">{{ student_form.non_field_errors }}</div>
        {% endif %}
        <div class="mb-3">
            {{ student_form.mat_number.label_tag }}
            {{ student_form.mat_number }}
            {% for e in student_form.mat_number.errors %}<div class="text-danger
small">{{ e }}</div>{% endfor %}
        </div>
        <div class="mb-3">
            {{ student_form.first_name.label_tag }}
            {{ student_form.first_name }}

```

```

        {% for e in student_form.first_name.errors %}<div class="text-danger
small">{{ e }}</div>{% endfor %}
    </div>
    <div class="mb-3">
        {{ student_form.last_name.label_tag }}
        {{ student_form.last_name }}
        {% for e in student_form.last_name.errors %}<div class="text-danger
small">{{ e }}</div>{% endfor %}
    </div>
    <div class="mb-3">
        {{ student_form.assigned_courses.label_tag }}
        {{ student_form.assigned_courses }}
        {% for e in student_form.assigned_courses.errors %}<div class="text-danger
small">{{ e }}</div>{% endfor %}
    </div>
    <div class="d-grid mt-2"><button class="btn btn-primary">Add Allowed
Student</button></div>
</form>
</div>
</div>

<div class="admin-card">
    <div class="card-header d-flex align-items-center"><span class="card-
icon">👤</span> Add Allowed Lecturer</div>
    <div class="card-body">
        <form method="post">{% csrf_token %}
            <input type="hidden" name="action" value="create_allowed_lecturer" />
            {% if lecturer_form.non_field_errors %}
                <div class="alert alert-danger">{{ lecturer_form.non_field_errors }}</div>
            {% endif %}
        <div class="mb-3">
            {{ lecturer_form.lecturer_id.label_tag }}
            {{ lecturer_form.lecturer_id }}

```

```

        {% for e in lecturer_form.lecturer_id.errors %}<div class="text-danger
small">{{ e }}</div>{% endfor %}
    </div>
    <div class="mb-3">
        {{ lecturer_form.full_name.label_tag }}
        {{ lecturer_form.full_name }}
        {% for e in lecturer_form.full_name.errors %}<div class="text-danger
small">{{ e }}</div>{% endfor %}
    </div>
    <div class="mb-3">
        {{ lecturer_form.assigned_courses.label_tag }}
        {{ lecturer_form.assigned_courses }}
        {% for e in lecturer_form.assigned_courses.errors %}<div class="text-danger
small">{{ e }}</div>{% endfor %}
    </div>
    <div class="d-grid mt-2"><button class="btn btn-primary">Add Allowed
Lecturer</button></div>
</form>
</div>
</div>
</div>
</div>
{% endblock %}

```