ACKNOWLEDGEMENT

We would like to express our deepest gratitude to our project supervisor and department coordinator, **Ms. Susan Sunuwar**, for her invaluable guidance, constant encouragement, and constructive suggestions throughout the development of the "Study Resource Management System". Her dual role provided us with both the technical mentorship required for software development and the administrative support necessary to align our work with the department's standards.

We would also like to thank the entire Department of Computer Application for providing the necessary CASE tools and environment to conduct our project work. The resources provided by the college were essential in helping us bridge the gap between theoretical knowledge and practical application.

Special thanks go to our family and friends for their continuous support and motivation. Finally, we would like to acknowledge the Faculty of Humanities and Social Sciences, Tribhuvan University, for including this Project I course in the curriculum, which has significantly improved our software development, teamwork, and report-writing skills.

ABSTRACT

The "Study Resource Management System" is a web-based application developed as part of the fourth-semester Project I (CACS256) curriculum to streamline the distribution and management of academic materials. In the current academic environment, manual sharing of resources often leads to data inconsistency and lack of verified materials. This project addresses these issues by providing a centralized repository where administrators can manage study resources and students can access them efficiently.

The system is developed following the Waterfall Model of the Software Development Life Cycle (SDLC), ensuring a structured approach from requirement analysis to final testing. The architecture follows a Client-Server model, utilizing a front-end designed with HTML, CSS, and JavaScript, while the back-end is powered by PHP with a MySQL database. Key features include Role-Based Access Control (RBAC), where the Admin has full CRUD (Create, Read, Update, Delete) privileges for managing PDFs and lecture notes, while Students are restricted to viewing and downloading verified resources.

System analysis was conducted using CASE tools to create Data Flow Diagrams (DFD) and Entity-Relationship Diagrams (ERD). The final implementation was verified through unit and system testing to ensure reliability and security. The outcome of this project is a functional, user-friendly platform that enhances the resource-sharing process within an academic institution.

Keywords: *CRUD, Resource Management, RBAC, BCA Project I, Waterfall Model, Web Application.*

Table of Contents

List of Tables

List of Figures

LIST OF ABBREVIATION

SRMS	Study Resource Management System
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheets
CRUD	Create, Read, Update, Delete
SQL	Structured Query Language
XAMPP	Cross-Platform (X), Apache (A), MariaDB (M), PHP (P), and Perl (P)
BCA	Bachelors in Computer Application
UI	User Interface

Chapter 1 : Introduction

1.1 Introduction

Study Resource Management System is a web portal for every student of the T.U BCA aiding them by providing notes, exam papers, important announcement and study materials in a digital way. The main parties involved in the system are the administrators, who acts as the provider and the students, who are the end benefactor of the system.

The best assistant for last minute exam preparation as well as any hardworking student. The primary objective of the system is providing a centralized platform for resources overcoming the traditional approach. Features include administrators uploading and organizing resources, while students are able to search, access, download and provide feedbacks

In the current time frame, study materials are primarily circulated manually through physical notes, printed handouts, emails or other social media platforms. For the elimination of existing system drawbacks, (i.e. time-consuming manual distribution, inconvenience in tracking, chances of misplacement or loss) the proposed system aims to delivers centralized, automated, and secure digital platform.

In a nutshell, the system is a student-centric platform making the access of resources more efficient.

1.2 Problem Statement

- Manual sharing of resourced through the traditional methods such as email, paper notes, social media inducing difficulty in organization,
- Absence of centralized storage system of existing contemporary resource management systems,

1.3 Objectives

- Provide Centralized platform for relevant resources,
- Reduce manual work for collection, management, circulation of academic resources.

1.4 Scope and limitation

1.4.1 Scope

- Providing a centralized platform for students to access notes, questions, and reference materials categorized by semester and subject,
- Enabling authorized staff to perform CRUD (Create, Read, Update, Delete) operations on study materials, including the automatic cleanup of server-side files**,**
- Systematic classification of all uploaded materials based on Semester (1-8) and Subject Name for efficient navigation,
- Includes server-side logic to automatically remove physical files from the directory when a record is deleted, optimizing storage space,
- Implementation of dynamic filtering to allow users to quickly locate specific resources by semester and material type.

1.4.2 Limitation

- The system resources are applicable specifically for students of T.U BCA,
- Mandatory internet connectivity to access of un-downloaded resources

1.5 Report Organization

This report on the Study Resources Management System is organized into five main chapters to provide a clear, systematic, and comprehensive overview of the project development process. Each chapter focuses on a specific aspect of the project to ensure clarity and understanding for the reader.

Chapter 1 Introduction:
This chapter introduces the Study Resources Management System, including the background, objectives, and significance of the project. It explains the problems faced by students and educators in managing study materials and the need for a digital platform to store, organize, and access resources efficiently. The chapter also outlines the scope, limitations, and expected outcomes of the project.

Chapter 2 System Analysis and Design:
This chapter covers the analysis of system requirements, both functional and non-functional. It includes techniques used to gather requirements, such as team discussions and surveys. System modeling tools such as Data Flow Diagrams (DFDs), use case diagrams, and interface structure diagrams are presented to provide a visual representation of the system’s workflow and user interactions.

Chapter 3 System Design:
This chapter describes the detailed design of the Study Resources Management System. It explains the database design, including tables for users, resources, and categories. The design of modules, interfaces, and user interaction flows is also detailed. Screenshots and interface diagrams illustrate how the user and admin interact with the system.

Chapter 4 Implementation and Testing:
This chapter explains the actual coding and development process. It describes the tools, programming languages, and platforms used, such as PHP, MySQL, HTML, CSS, Bootstrap, XAMPP, and Git/GitHub for version control. It also presents unit testing and system testing of each module to verify that the system functions as expected and meets the specified requirements.

Chapter 5 Conclusion and Recommendations:
The final chapter summarizes the lessons learned from the project, the outcomes achieved, and the significance of the Study Resources Management System. It provides conclusions on the effectiveness of the system and gives recommendations for future enhancements and improvements.

Chapter 2 : Background Study and Literature Review

2.1 Background Study

The foundation of the **Study Resource Management System** lies in the integration of Information Systems (IS) with web-based content delivery. An Information System in an educational context is defined as an integrated set of components for collecting, storing, and processing data to provide information and digital products to students and faculty. These systems function as a specialized Content Management System (CMS), distinct from broad Learning Management Systems (LMS). While an LMS typically manages entire courses, quizzes, and grades, a Resource Management System focuses specifically on the efficient storage, organization, and retrieval of academic files such as PDFs, lecture notes and past questions. This distinction is crucial for this project, which aims to provide a lightweight, resource-centric alternative to heavy academic software.

Security and data integrity in such web applications are maintained through Role-Based Access Control (RBAC). RBAC is a security mechanism that restricts system access to authorized users based on their specific roles within an organization. In the context of this project, the system implements a hierarchical permission structure where the 'Admin' role possesses full Create, Read, Update, and Delete (CRUD) privileges to ensure content validity, while the 'Student' role is limited to 'Read' and 'Download' permissions. This separation of concerns ensures that the integrity of the study materials is preserved against unauthorized modifications or accidental deletions.

2.2 Literature Review

Several existing systems and studies have addressed the challenge of digital academic resource distribution. A recent survey on digital library systems highlighted that while general file-sharing platforms (like Google Drive) are widely available, they often lack the structured, syllabus-oriented categorization required for specific academic programs like BCA. The study emphasized that students frequently struggle with unstructured "peer-to-peer" sharing, where the authenticity of notes cannot be verified. To address this, researchers have proposed centralized "Online Note Sharing Systems" that allow verified faculties to upload materials sorted by subject and semester. These systems have proven that centralized control significantly reduces the distribution of incorrect or outdated syllabus materials.

Furthermore, comparative studies between Learning Management Systems (LMS) and Content Management Systems (CMS) have revealed significant trade-offs in implementation. Platforms like Moodle and Blackboard offer comprehensive features but are often criticized for being resource-heavy, expensive to host, and complex to navigate for simple file-retrieval tasks. A 2025 study on the "EduShare" platform demonstrated that lightweight web technologies (such as PHP/MySQL or modern JS frameworks) could create scalable, responsive resource platforms that perform better than traditional LMS for the specific purpose of material sharing.

Despite these advancements, a gap remains in the availability of cost-effective, role-secured systems specifically tailored for the local BCA curriculum. Most existing simple solutions lack the necessary RBAC security layers, allowing unauthorized users to manipulate data, while secure enterprise solutions are often beyond the budget of smaller colleges. The **Study Resource Management System** aims to bridge this gap by developing a system that combines the security of RBAC with the simplicity of a dedicated resource manager, ensuring that students have access to verified, curriculum-mapped study materials without the administrative overhead of a full LMS.

2.3 Methodology

The development of the Study Resource Management System follows the **Waterfall Model**. This model is a linear-sequential life cycle approach where each phase must be completed before the next one begins. This methodology was chosen due to the clearly defined project requirements and the fixed academic timeline.

The project is divided into the following sequential phases:

1. **Requirement Analysis:** Gathering all functional and non-functional needs, such as the Admin's ability to edit resources and the Student's viewing permissions.
2. **System Design:** Designing the database schema and system architecture (ERD and DFD) to ensure efficient data flow between roles.
3. **Implementation (Coding):** Translating the design into actual code using the chosen technology stack.
4. **Testing:** Executing test cases to verify that the Admin and Student modules function correctly without errors.
5. **Documentation:** Compiling the final report according to the standards prescribed by the university.

Chapter 3 : System Analysis and Design

3.1 System Analysis

A thorough System analysis was done regarding the system's functional, non-functional requirements, feasibility, data, process modelling.

3.1.1 Requirement Analysis

The requirement analysis of SRMS is done through evaluation of the functional and non-functional requirements of the system.

3.1.1.1 Functional Requirements

- The system must be allowed to register and log in securely using unique credentials in accordance to their respective roles (student, admin),
- The admin should be allowed to upload, organize and manage (edit, delete) academic resources like notes, exam papers,
- The admin must be able to post announcements,
- The system must provide a way for admins to view and respond to feedbacks,
- Student must be able to view all the present resources,
- Student must have the ability to view announcements and give feedbacks.

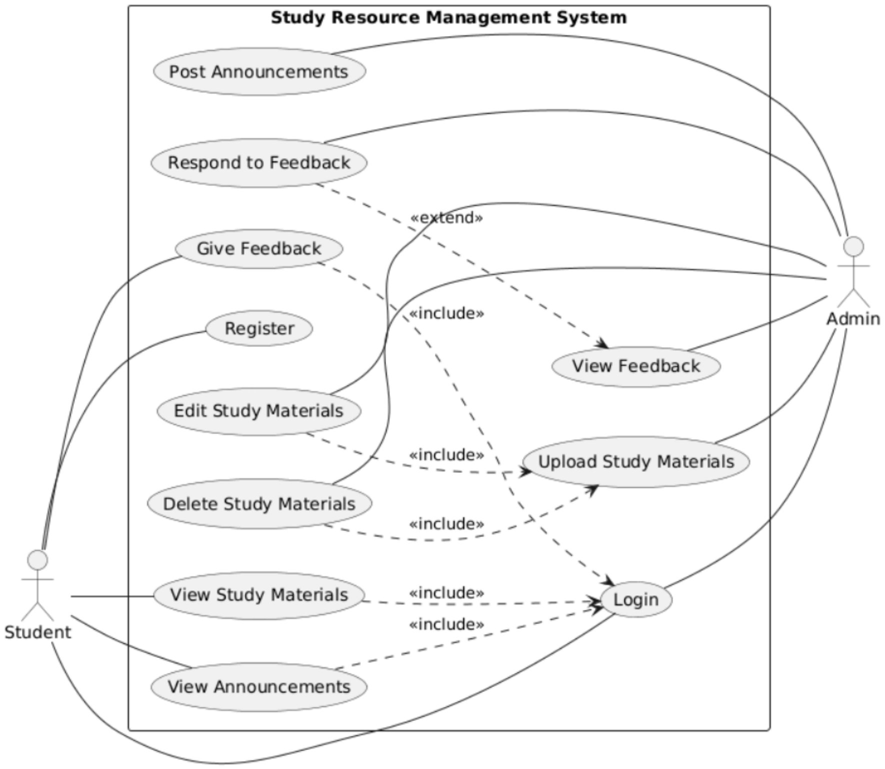


Fig. 3-1:Use-Case Diagram

3.1.1.2 Non-functional Requirements

- The system should load pages within 2-3 seconds under normal network conditions,
- The system should be available 24/7, except during maintenance,
- User passwords shall be stored using hashing/encryption,
- The system shall support at least 200 concurrent users without performance degradation,
- The system shall maintain data accuracy of 100% during normal operations.

3.1.2 Feasibility Analysis

i. **Technical feasibility**

The user interface for both Administrators and Students can be developed using standard web languages like HTML, CSS ensuring the platform is accessible via modern web browsers. Similarly, for back-end technologies the server-side language PHP (using XAMPP) can handle the CRUD operation. For database management MySQL is capable of securely storing user credentials and metadata for the study materials. Hence, it is technically feasible.

ii. **Economic feasibility**

The system is built using open-source tools like XAMPP, MySQL, and PHP, which have no licensing fees. It eliminates the need for physical printing and photocopying of study materials, saving money for both the college and the students. With free-to-use development, deployment, testing tools the acquired final budget is highly possible to be within the allocated budget range. Therefore, the system is economically feasible.

iii. **Operational feasibility**

The targeted users of the system are the students of BCA who are technically literate and familiar with the use of web-based applications. Additionally, both the admin and student interface will contain simple, easy to use and understand UI making it easy to navigate. Thereafter implementation minimal training requirement makes the SRMS operationally feasible.

iv. Schedule feasibility

The project is designed to be completed within the duration of a single academic semester. The development process is divided into distinct phases: Requirement Analysis, System Design, Implementation, and Testing. By utilizing efficient open-source tools like XAMPP and PHP, the project milestones are realistic and can be achieved before the final submission deadline.

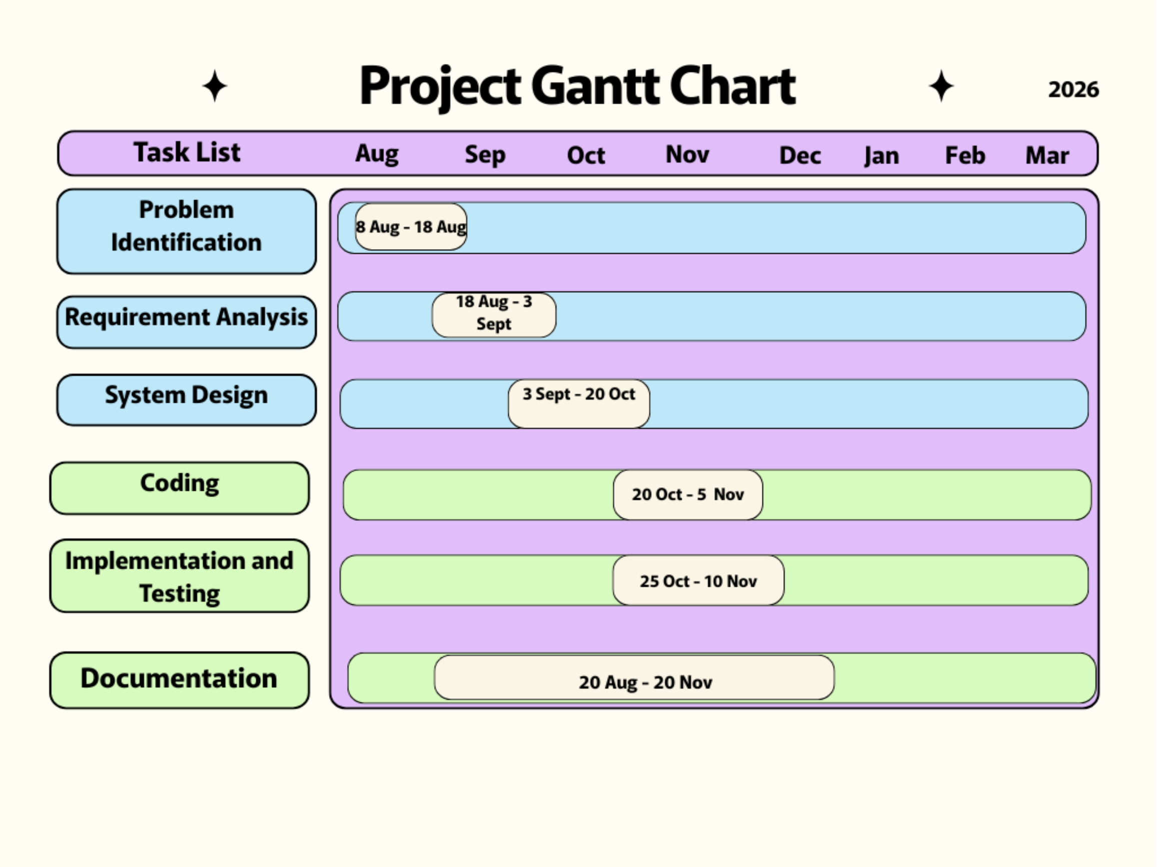


Fig. 3-2: Gantt Chart for Study Resource Management System

3.1.3 Data Modelling (ER-Diagram)

For data modeling, the ER diagram of SRMS is shown below as:

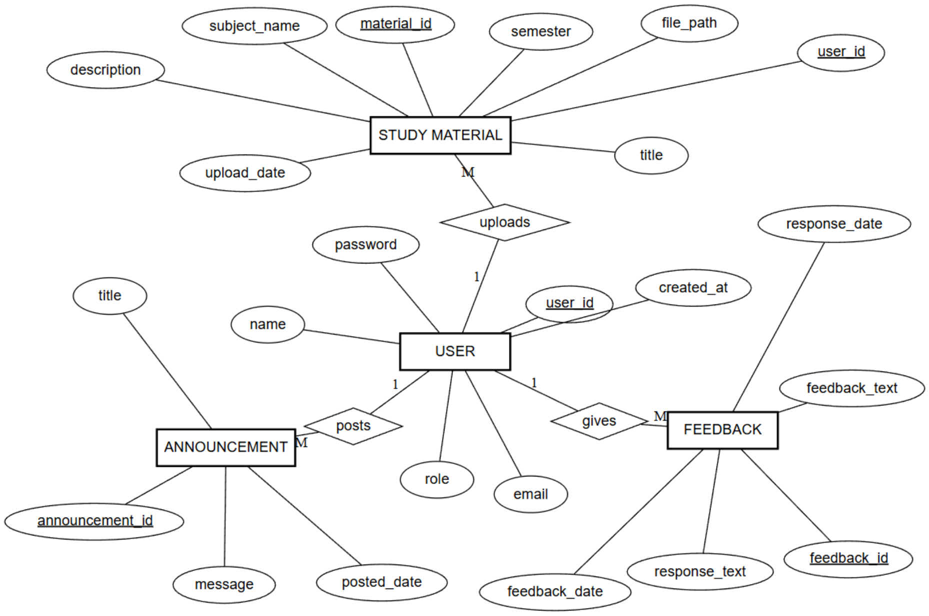


fig. 3-3: ER-Diagram for SRMS

Here, Users entity is related with other entities such as feedback, study material and announcement. A user can upload multiple study materials, give multiple feedbacks and post announcement. Each entity has its own primary key attribute assigned. For e.g.: user_id for user, announcement_id for announcement, material_id for study material and feedback_id for feedback.

3.1.4 Process Modeling (DFD)

For process modeling of SRMS, DFD up to level 1 are as follows:

3.1.4.1 Context Diagram:

In context diagram there are two entities i.e. Admin, Student.

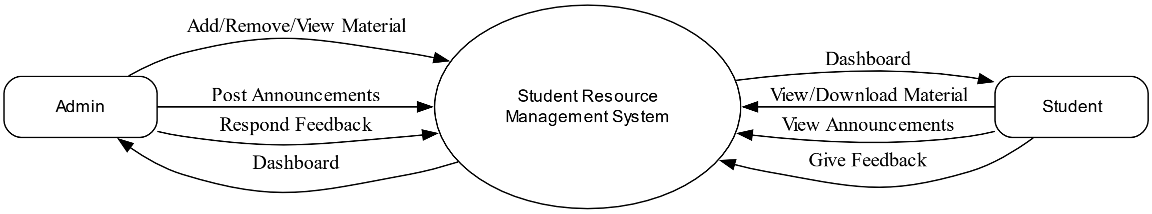


fig. 3-4: Context Diagram of SRMS

3.1.4.2 Level 1 DFD:

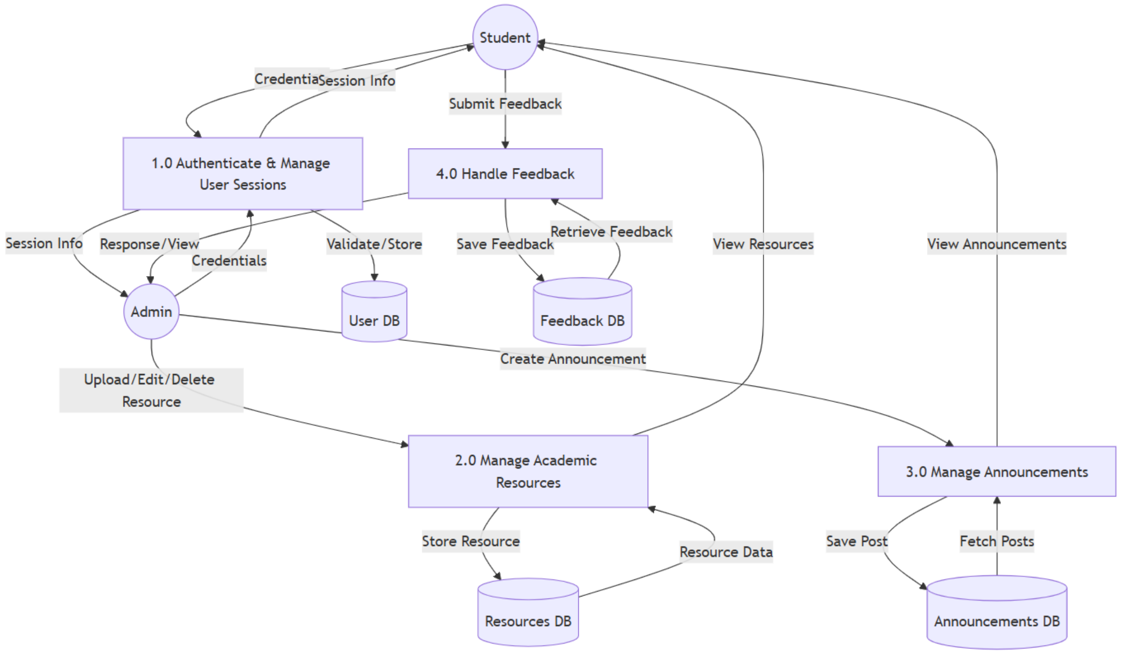


fig. 3-5: DFD for SRMS (level 1)

3.2 System Design

3.2.1 Architectural Design

The architectural design of the Study Resource Management System is based on the **Client-Server Architecture**, which separates the user interface (client) from the data storage and processing (server). This architecture ensures centralized data management and secure access for different user roles.

The system is divided into three distinct layers:

- **Presentation Layer (Client Side):** This is the top-most level of the application which the users interact with directly. It is developed using web technologies (HTML, CSS, JavaScript) to provide a responsive interface for both Admin (for uploading resources) and Students (for viewing resources).

- **Application/Logic Layer (Server Side):** This layer acts as the intermediary between the user interface and the database. It handles the core functional logic, such as authentication, role-based access control, and the processing of file upload/download requests.

- **Data Layer (Database):** This layer consists of the relational database management system (MySQL/PostgreSQL) and the physical file storage. It is responsible for storing resource metadata (titles, descriptions, categories) and the actual physical study materials securely.

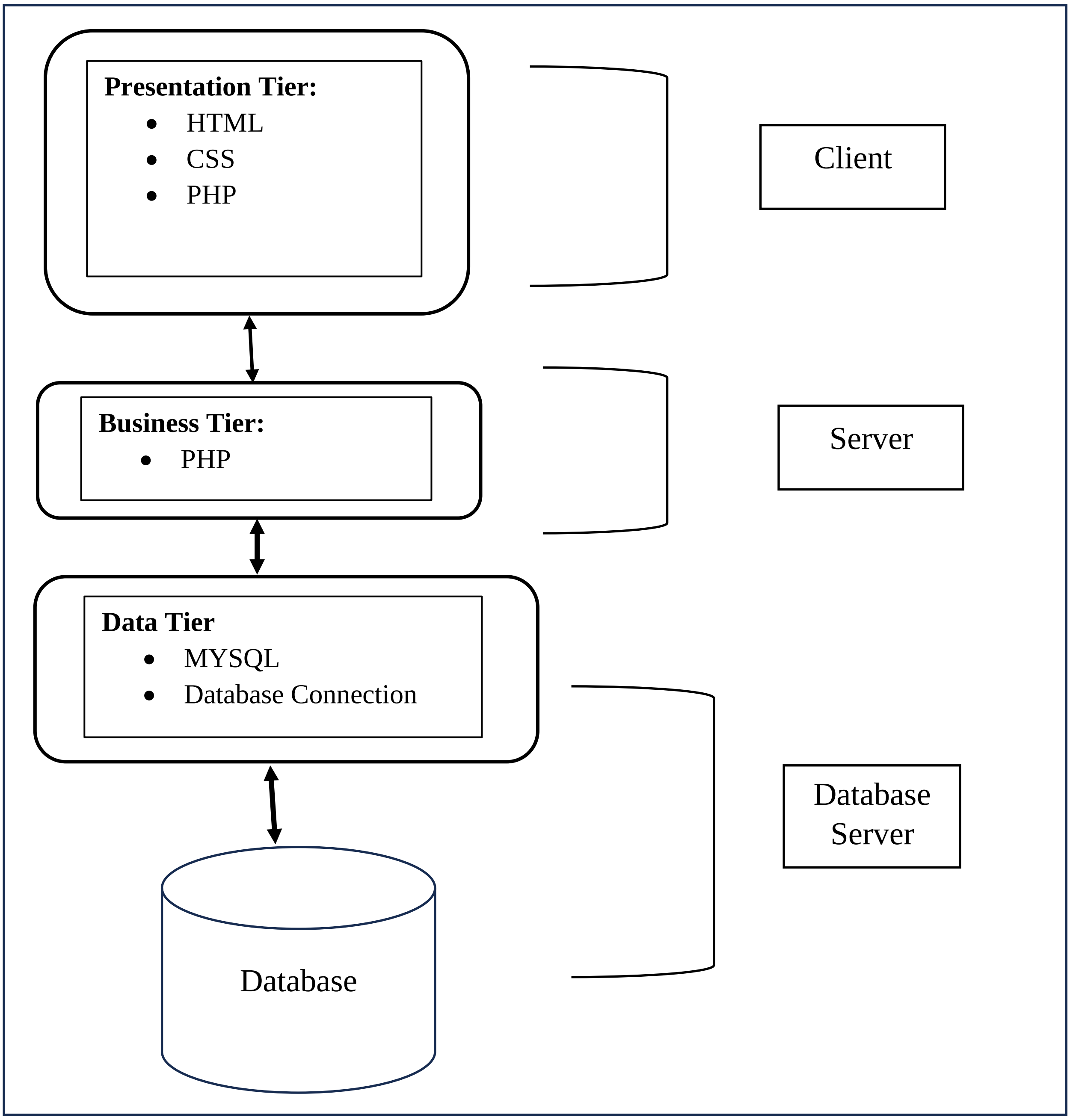
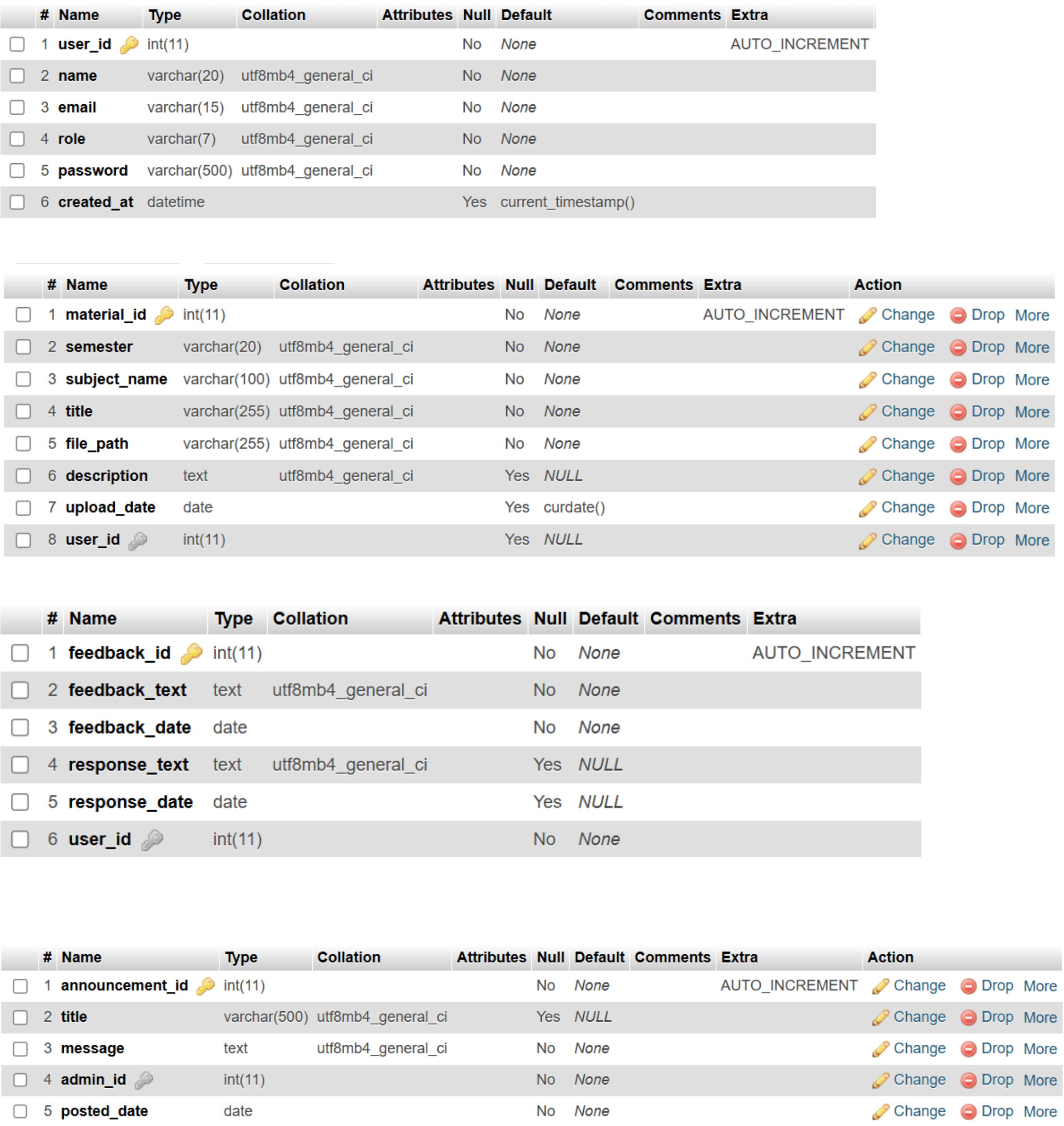


fig 3-6: Architectural Design

3.2.2 Database Schema Design



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
□	1	**user_id** 🔑	int(11)			No	*None*		AUTO_INCREMENT
□	2	**name**	varchar(20)	utf8mb4_general_ci		No	*None*		
□	3	**email**	varchar(15)	utf8mb4_general_ci		No	*None*		
□	4	**role**	varchar(7)	utf8mb4_general_ci		No	*None*		
□	5	**password**	varchar(500)	utf8mb4_general_ci		No	*None*		
□	6	**created_at**	datetime			Yes	current_timestamp()		

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	**material_id** 🔑	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
□	2	**semester**	varchar(20)	utf8mb4_general_ci		No	*None*			Change Drop More
□	3	**subject_name**	varchar(100)	utf8mb4_general_ci		No	*None*			Change Drop More
□	4	**title**	varchar(255)	utf8mb4_general_ci		No	*None*			Change Drop More
□	5	**file_path**	varchar(255)	utf8mb4_general_ci		No	*None*			Change Drop More
□	6	**description**	text	utf8mb4_general_ci		Yes	*NULL*			Change Drop More
□	7	**upload_date**	date			Yes	curdate()			Change Drop More
□	8	**user_id** 🔑	int(11)			Yes	*NULL*			Change Drop More

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
□	1	**feedback_id** 🔑	int(11)			No	*None*		AUTO_INCREMENT
□	2	**feedback_text**	text	utf8mb4_general_ci		No	*None*		
□	3	**feedback_date**	date			No	*None*		
□	4	**response_text**	text	utf8mb4_general_ci		Yes	*NULL*		
□	5	**response_date**	date			Yes	*NULL*		
□	6	**user_id** 🔑	int(11)			No	*None*		

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	**announcement_id** 🔑	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
□	2	**title**	varchar(500)	utf8mb4_general_ci		Yes	*NULL*			Change Drop More
□	3	**message**	text	utf8mb4_general_ci		No	*None*			Change Drop More
□	4	**admin_id** 🔑	int(11)			No	*None*			Change Drop More
□	5	**posted_date**	date			No	*None*			Change Drop More

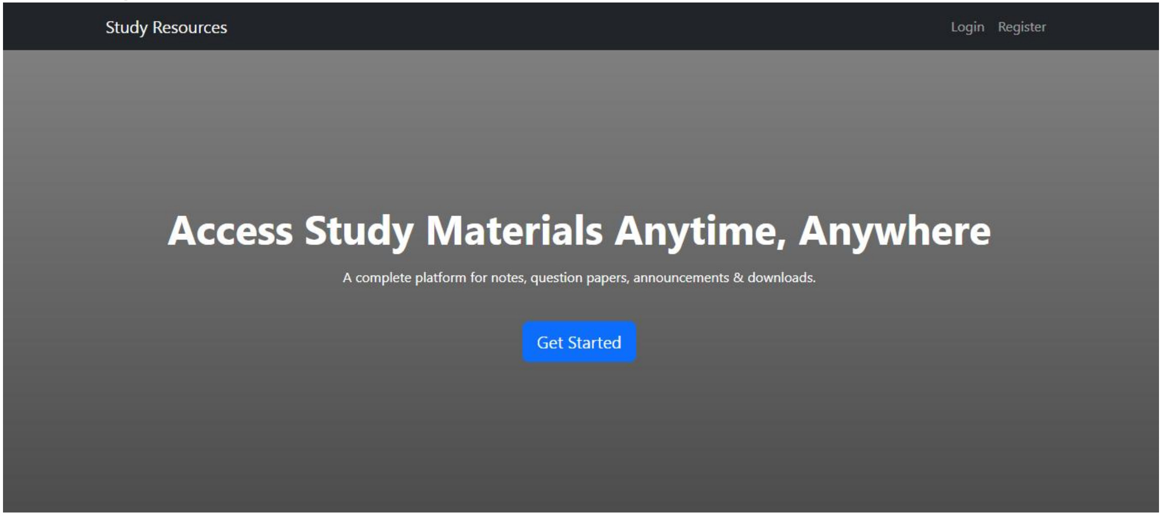
Fig 3-7 Database Schema Design

3.2.3 Interface Design (UI Interface/Interface Structure and Diagrams)

The interface design of the Study Resource Management System defines how users interact with the system. It provides a simple and user-friendly layout that allows students and administrators to easily access system features. The system includes interfaces such as login, dashboard, resource upload, and resource viewing pages. Each interface is designed to ensure easy navigation, clarity, and efficient management of study materials.

3.2.3.1 Interface Structure Diagram

a) Homepage Interface

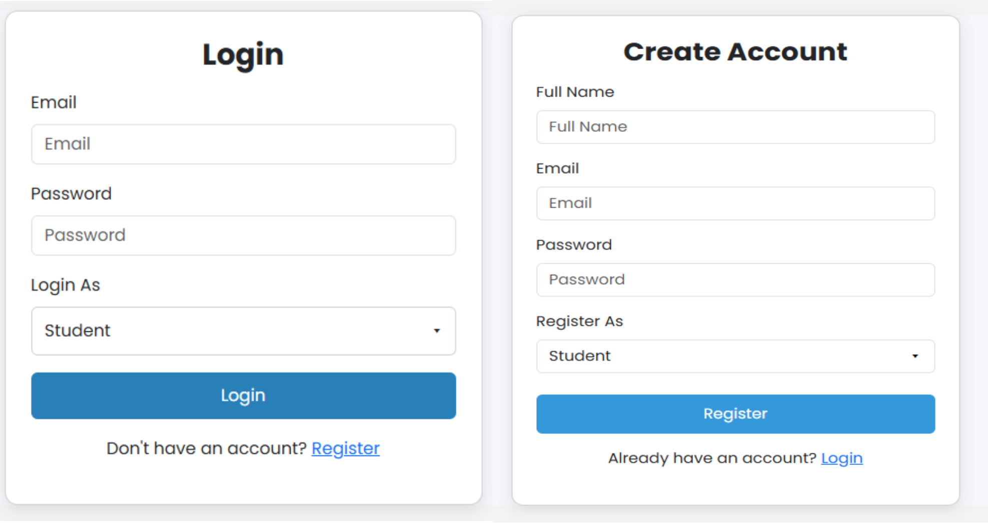


Access Study Materials Anytime, Anywhere

A complete platform for notes, question papers, announcements & downloads.

Fig 3-8 Homepage Interface

b) Login and Register Interface



Login

Email: Email

Password: Password

Login As: Student

Don't have an account? Register

Create Account

Full Name: Full Name

Email: Email

Password: Password

Register As: Student

Already have an account? Login

Fig 3-9 Login and Register Interface

c) Admin Dashboard Interface

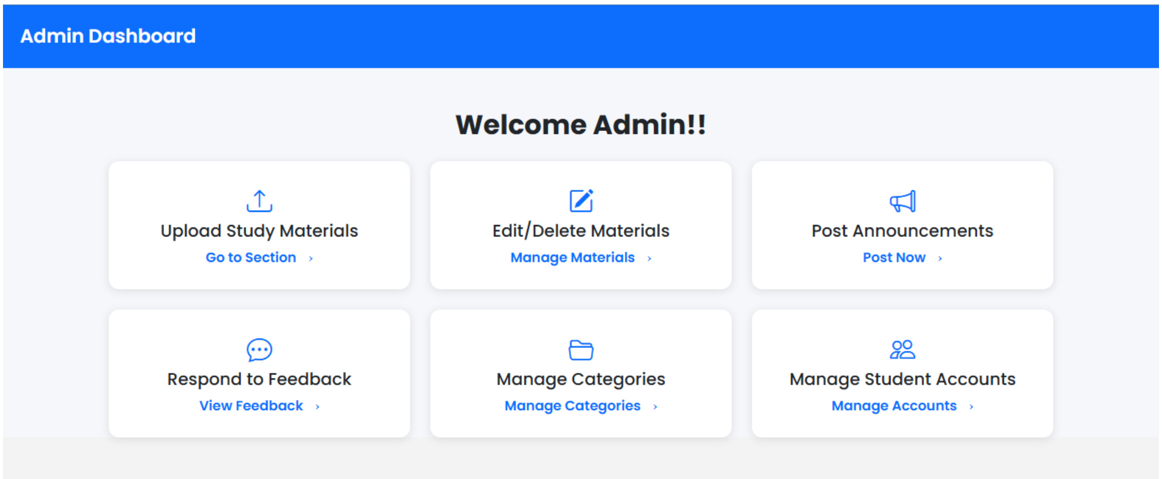
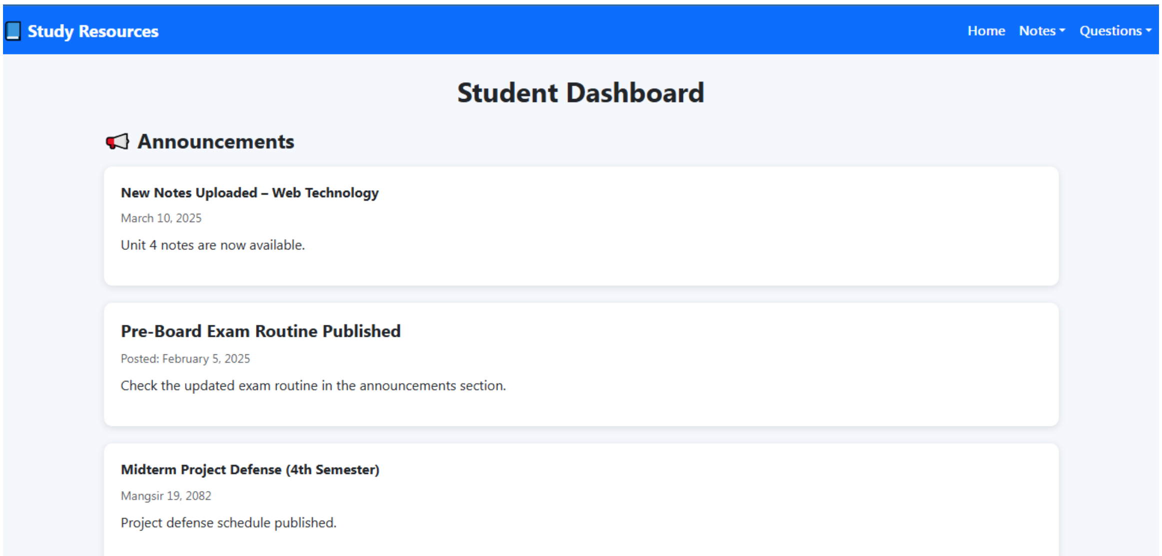


Fig 3-10 Admin Dashboard Interface

d) Student Dashboard Interface



New Notes Uploaded – Web Technology

March 10, 2025

Unit 4 notes are now available.

Pre-Board Exam Routine Published

Posted: February 5, 2025

Check the updated exam routine in the announcements section.

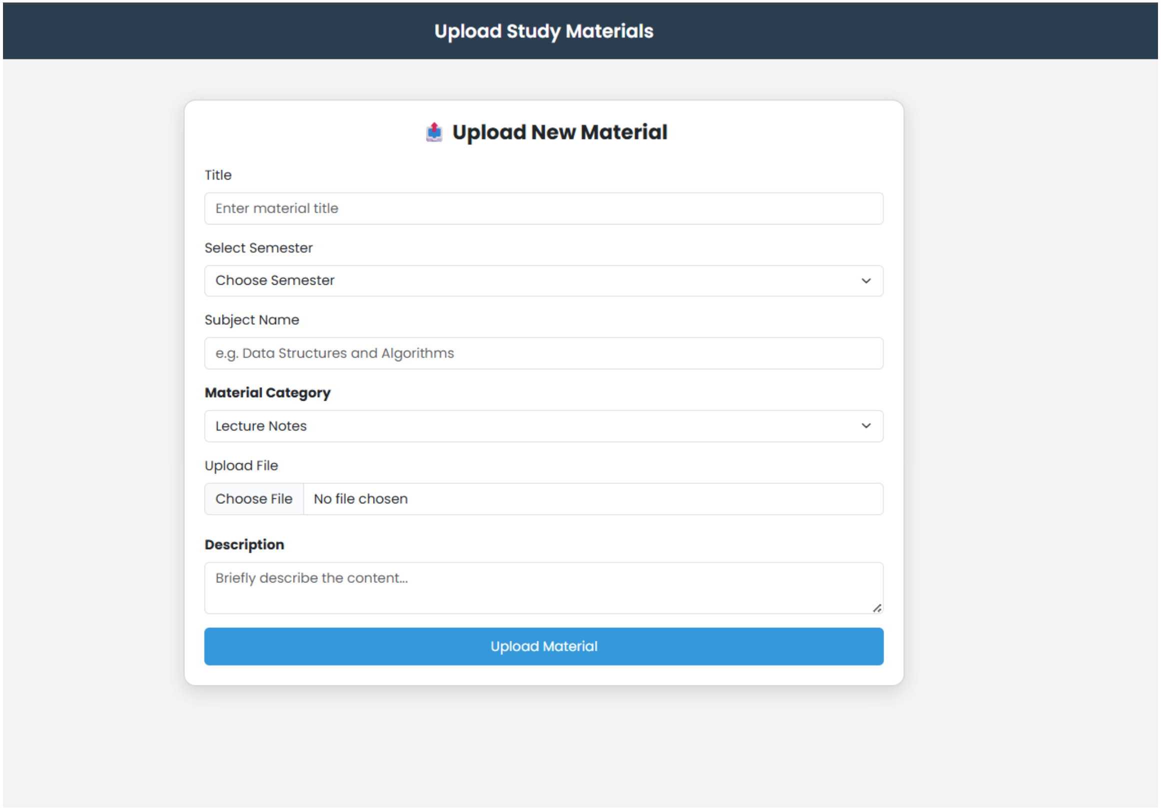
Midterm Project Defense (4th Semester)

Mangsir 19, 2082

Project defense schedule published.

Fig 3-11 Student Dashboard Interface

e) Upload Resource Interface



Upload Study Materials

Upload New Material

Title

Enter material title

Select Semester

Choose Semester

Subject Name

e.g. Data Structures and Algorithms

Material Category

Lecture Notes

Upload File

Choose File No file chosen

Description

Briefly describe the content...

Fig 3-12 Upload Resource Interface

3.2.3.2 Description of Interfaces

a) Homepage Interface

The homepage is the main entry point of the Study Resource Management System. It provides basic information about the system and allows users to navigate to the login and registration pages easily.

b) Login and Register Interface

The login and registration interface allows users to create a new account or access the system using valid credentials. It ensures secure authentication and redirects users to the appropriate dashboard based on their role.

c) Admin Dashboard Interface

The admin dashboard allows the administrator to manage users, upload resources, post announcements, respond to feedback and organize study materials by category.

d) Student Dashboard Interface

The student dashboard provides options to view announcements, give feedback and view, and download study resources such as notes and questions of any semester.

e) **Upload Resource Interface**
This interface allows users to upload study materials with details like title, semester, subject, and file type.

Chapter 4: Implementation and Testing

4.1. Implementation

Study Resource Management System is developed using appropriate programming languages, tools, and technologies. All system modules are implemented and integrated to perform the required operations efficiently.

4.1.1. Tools Used (CASE tools, Programming languages, Database platforms)

a) **CASE Tools (Computer-Aided Software Engineering):**
CASE tools were used to support the development, implementation, and management of the Study Resource Management System. These tools helped in designing, developing, testing, and maintaining the system efficiently.

i) **phpMyAdmin**
phpMyAdmin is a web-based database management tool used to manage the MySQL database of the Study Resource Management System. It was used to create, modify, and manage database tables, as well as to insert, update, and delete records. phpMyAdmin also helped in executing SQL queries and maintaining the database efficiently.

ii) **Git and GitHub**
Git is a version control system used to track and record changes made to the source code during the development of the Study Resource Management System. GitHub was used as an online repository to store the project files and maintain a history of code updates. These tools helped in tracking updates, managing different versions of the code, and maintaining a proper record of project development.

iii) **XAMPP**
XAMPP is a local server environment that includes Apache, MySQL, PHP, and Perl. It was used to run and test the Study Resource Management System on a local machine. XAMPP provided a platform to host the application locally and allowed smooth interaction between the frontend, backend, and database.

b) **Programming Languages:**

i) **Frontend**

- **HTML5**
  HTML5 was used to create the basic structure and content of the user interface of the Study Resource Management System.
- **CSS3**
  CSS3 was used to design and style the web pages, improving layout, readability, and visual presentation.

- **Bootstrap**
  Bootstrap was used in a limited manner to support basic layout alignment and responsive design, without relying on advanced components or frameworks.

ii) **Backend**

- **PHP**
  PHP was used as the server-side programming language to implement the core functionality of the Study Resource Management System. It was used to handle user authentication, process form data, interact with the MySQL database, and manage system logic.

c) **Database Platforms:**

i) **MySQL:**
MySQL was used as the database platform to store and manage system data such as user details and study resources. It provides efficient data management, data security, and fast access to stored information.

4.1.2 Implementation Details of Modules (Description of procedures/functions)

1) **User Authentication Module**
This module is responsible for user registration and login. During registration, user details are collected through a form and stored in the database. During login, the system verifies the entered username and password with the stored records. If the credentials are valid, the user is redirected to the appropriate dashboard; otherwise, an error message is displayed.

2) **User Management Module (Admin)**
This module allows the administrator to manage users of the system. It provides functions to view user details and control user access. The administrator can perform operations such as updating user information and managing user records stored in the database.

3) **Resource Upload Module**
The resource upload module allows users to upload study materials such as notes and assignments. The system validates the uploaded file and stores it in the server directory. Related information such as resource title, subject, and uploader details are saved in the database for future access.

4) **Resource Viewing Module**
This module enables users to view a list of available study resources. Users can select a resource and view it from the system. The module retrieves resource information from the database and provides access to the stored files.

5) **Category / Subject Management Module**
This module helps in organizing study resources by subject or category. It allows the system to group resources properly, making searching and filtering of study materials easier and more efficient.

6) **Admin Dashboard Module**
The admin dashboard module provides a centralized interface for the administrator to monitor system activities. It displays summary information and provides access to user management and resource management functions.

4.2 Testing

Testing is carried out to ensure that the Study Resource Management System functions correctly and meets the specified requirements. Different test cases were executed to verify the correctness, reliability, and performance of individual modules as well as the complete system.

4.2.1 Test Cases for Unit Testing

Unit testing focuses on testing individual modules of the Study Resource Management System separately to ensure that each function performs as expected.

Table 4-1 Test Case for Registration Module

Test Case ID	**Module**	**User Type**	**Input**	**Expected Output**	**Actual Output**	**Status (Pass/Fail)**
Proj_REG_01	Registration	Admin	Valid Email: Admin1@gmail.com & Password: Admin@123	Admin registered successfully and redirected to login page	Admin registered successfully	Pass
Proj_REG__02	Registration	Admin	Blank Email / Password	Error message: "All fields are required"	All fields are required	Pass
Proj_REG_03	Registration	Admin	Weak Password (less than 6 characters)	Error message: "Password too weak"	Password too weak	Pass
Proj_REG_04	Registration	User	Valid Email: user1@gmail.com & Password:user@123	User Logged in successfully and redirected to login page	User registered successfully	Pass

Test Case ID	**Module**	**User Type**	**Input**	**Expected Output**	**Actual Output**	**Status (Pass/Fail)**
Proj_REG_05	Registration	User	Blank Email / Password	Error message: "All fields are required"	All fields are required	Pass
Proj_REG_06	Registration	User	Weak Password (less than 6 characters)	Error message: "Password too weak"	Password too weak	Pass

Table 4-2 Test Case for Login Module

Test Case ID	**Module**	**User Type**	**Input**	**Expected Output**	**Actual Output**	**Status (Pass/Fail)**
Proj_LOG_01	Login	Admin	Valid Email & Password	Admin dashboard opens successfully	Admin dashboard opens successfully	Pass
Proj_LOG_02	Login	Admin	Invalid Email	Error message: "Invalid Email"	Invalid Email	Pass
Proj_LOG_03	Login	Admin	Invalid Password	Error message: "Invalid Password"	Invalid Password	Pass
Proj_LOG_04	Login	User	Valid Email & Password	User dashboard opens successfully	User Dashboard opens successfully	Pass
Proj_LOG_05	Login	User	Invalid Email	Error message: "Invalid Email"	Invalid Email	Pass
Proj_LOG_06	Login	User	Invalid Password	Error message: "Invalid Password"	Invalid Password	Pass

Table 4-3 Test Case for User Management Module

Test Case ID	**Module**	**Input / Action**	**Expected Output**	**Actual Output**	**Status (Pass/Fail)**
Proj_UM_01	User Management	View all users	List of all registered users displayed	List of all registered users displayed	Pass
Proj_UM_02	User Management	Update user details	User information updated successfully	User information updated successfully	Pass
Proj_UM_03	User Management	Delete a user	Selected user removed from database	Selected user removed from database	Pass
Proj_UM_04	User Management	Search user by username/email	Correct user details displayed	Correct user details displayed	Pass

Table 4-4 Test Case for Resource Upload Module

Test Case ID	**Module**	**Input / Action**	**Expected Output**	**Actual Output**	**Status (Pass/Fail)**
Proj_RU_01	Resource Upload	Upload valid study material (PDF, DOC, etc.)	Resource uploaded successfully and displayed in list	Resource uploaded successfully	Pass
Proj_RU_02	Resource Upload	Upload unsupported file type	Error message: "File type not allowed"	File type not allowed	Pass
Proj_RU_03	Resource Upload	Upload file larger than allowed size	Error message: "File too large"	File too large	Pass
Proj_RU_04	Resource Upload	Blank fields while uploading	Error message: "All fields are required"	All fields are required	Pass

Table 4-5 Test Case for Resource Viewing Module

Test Case ID	**Module**	**Input / Action**	**Expected Output**	**Actual Output**	**Status (Pass/Fail)**
Proj_RV_01	Resource Viewing	Click on a resource to view	Resource content displayed in readable format	Resource content displayed in readable format	Pass
Proj_RV_02	Resource Viewing	Search resource by name/category	Relevant resources displayed	Relevant resources displayed	Pass
Proj_RV_03	Resource Viewing	Attempt to view non-existent resource	Error message: "Resource not found"	Resource not found	Pass

Table 4-6 Test Case for Category / Subject Management Module

Test Case ID	**Module**	**Input / Action**	**Expected Output**	**Actual Output**	**Status (Pass/Fail)**
Proj_CM_01	Category Management	Add new category / subject	Category added successfully and listed	Category added successfully and listed	Pass
Proj_CM_02	Category Management	Update existing category	Category updated successfully	Category updated successfully	Pass
Proj_CM_03	Category Management	Delete category	Category removed from list	Category removed from list	Pass
Proj_CM_04	Category Management	Add duplicate category	Error message: "Category already exists"	Category already exists	Pass

4.2.2 Test Cases for System Testing

System testing was performed on the Study Resources Management System to ensure that all modules function correctly and interact seamlessly. It verified end-to-end operations such as user/admin login, registration, resource upload, viewing, download, user management, and category management, ensuring the system meets the specified requirements.

Table 4-7 Test Case for System Testing

Test Case ID	Module	User Type	Input / Action	Expected Output	Actual Output	Status (Pass/Fail)
Proj_ST_01	Login	Admin	Enter valid username & password	Admin dashboard opens successfully	Admin dashboard opens successfully	Pass
Proj_ST_02	Login	User	Enter valid username & password	User dashboard opens successfully	User dashboard opens successfully	Pass
Proj_ST_03	Login	Admin/User	Enter invalid username or password	Error message displayed	Error message displayed	Pass
Proj_ST_04	Registration	Admin/User	Fill valid details & submit	Account created successfully and redirect to login	Account created successfully and redirect to login	Pass
Proj_ST_05	Registration	Admin/User	Leave required fields blank	Error message displayed	Error message displayed	Pass
Proj_ST_06	Resource Upload	Admin	Upload valid study resource (PDF, DOC)	Resource uploaded successfully and listed	Resource uploaded successfully and listed	Pass
Proj_ST_07	Resource Upload	Admin	Upload unsupported file type	Error message: "File type not allowed"	Invalid File Type	Pass
Proj_ST_08	Resource Upload	Admin	Leave required fields blank	Error message: "All fields are required"	All fields are required	Pass
Proj_ST_09	User Management	Admin	Add / Update / Delete user	User list updated correctly	User list updated	Pass

Test Case ID	**Module**	**User Type**	**Input / Action**	**Expected Output**	**Actual Output**	**Status (Pass/Fail)**
Proj_ST_10	Category/Subject Management	Admin	Add new category	Category added successfully	Category added successfully	Pass
Proj_ST_11	Category/Subject Management	Admin	Update existing category	Category updated successfully	Category updated successfully	Pass
Proj_ST_12	Category/Subject Management	Admin	Delete a category	Category removed from the list	Category removed	Pass
Proj_ST_13	Logout	Admin/User	Click Logout button	Redirected to login page	Redirected to login page	Pass

Chapter 5: Conclusion and Future Recommendations

5.1. Lesson Leant / Outcome

Through the development of the Study Resource Management System, our team gained practical experience in web application development. We learned how to collaboratively design user-friendly interfaces, implement backend functionalities using PHP and MySQL, and manage data efficiently. Using Git and GitHub, we were able to track code changes, manage versions, and coordinate work among team members effectively. Tools like phpMyAdmin and XAMPP helped in managing the database and running the local server environment. This project also emphasized the importance of system testing and unit testing to ensure a reliable and functional application. Overall, we developed a better understanding of the software development life cycle, teamwork, and problem-solving in real project scenarios.

5.2. Conclusion

The Study Resource Management System successfully provides a platform for users to register, login, view, and download study materials, while enabling admins to manage resources, users, and categories efficiently. The project demonstrates a functional web-based application developed collaboratively by our team using PHP, MySQL, HTML, CSS, and Bootstrap. By using Git and GitHub, we maintained version control and coordinated our work efficiently. This project strengthened our programming skills, database knowledge, and understanding of web application development from design to implementation and testing. As our first team project, it serves as a foundation for more advanced projects in the future.

5.3 Future Recommendations

For future enhancements, the Study Resource Management System can be improved with features like advanced search filters, user access control, resource rating and feedback, and mobile responsiveness. Integration with cloud storage and adoption of modern frameworks like Laravel or React could improve scalability, performance, and maintainability. Implementing these features would make the system more robust, user-friendly, and suitable for use in larger educational environments.

REFERENCES

[1] P. Researcher, "EduShare: Building a Scalable and Collaborative Educational Resource Sharing Platform Using Web Technologies," *Preprints.org*, Mar. 2025. [Online]. Available: https://www.preprints.org/manuscript/202503.1547/v1/download

[2] Synergy Learning, "LMS vs CMS: What are the differences and which do I need?," *Synergy Learning Blog*, Apr. 16, 2025. [Online]. Available: https://synergy-learning.com/blog/lms-vs-cms-what-are-the-differences-and-which-do-i-need/

[3] R. J. Robles, M. K. Choi, and S. S. Yeo, "Application of Role-Based Access Control for Web Environment," in *2008 International Symposium on Ubiquitous Multimedia Computing*, Hobart, Australia, 2008, pp. 115-118. doi: 10.1109/UMC.2008.41.

[4] H. Dharmadhikari et al., "A Survey on Digital Library Notes Sharing Systems for Academic Resource Management," *International Journal of Computer Science Engineering Techniques (IJCSE)*, vol. 9, no. 6, Dec. 2025. [Online]. Available: https://www.ijcsejournal.org/survey-digital-library-resource-management/

[5] B. Bhattrai, S. Chimariya, and S. Luitel, "Online Note Sharing System Proposal," Dept. of Computer Science and Information Technology, Nepalgunj Campus, Tribhuvan University, Nepal, 2023.

[6] Tribhuvan University, *Bachelor of Computer Application (BCA) Project I Syllabus (CACS256)*, Kathmandu, Nepal, 2024.

[7] W. W. Royce, "Managing the Development of Large Software Systems," in *Proc. IEEE WESCON*, Aug. 1970, pp. 1–9. [The original paper defining the Waterfall phases].