

RAJALAKSHMI ENGINEERING COLLEGE

(An Autonomous Institution)

RAJALAKSHMI NAGAR, THANDALAM- 602 105



RAJALAKSHMI ENGINEERING COLLEGE

CS23332 – Database Management System

MINI PROJECT REPORT

NAME: MOHAMMED DHARIQ I B

YEAR/SEMESTER: 2ND YEAR/ III SEMESTER

BRANCH/SECTION: B.TECH INFORMATION TECHNOLOGY - C

REGISTER NO: 2116241001137

COLLEGE ROLL NO: 241001137

ACADEMIC YEAR: 2024. -2025.



RAJALAKSHMI ENGINEERING COLLEGE

(An Autonomous Institution)

RAJALAKSHMI NAGAR, THANDALAM- 602 105

BONAFIDE CERTIFICATE

Certified that this project “**STUDY MATERIAL / NOTES SHARING SYSTEM**” is the bonafide work of Mohammed Dhariq I B (241001137) and Mukesh Raja V (241001140) who carried out the project work under my supervision as part of the CS23332 – Database Management Systems Laboratory course.

Signature of Faculty In-charge

Submitted for the Practical Examination Held on:

Internal Examiner

External Examiner

ACKNOWLEDGEMENT

First, we thank the almighty God for the successful completion of the project. Our sincere thanks to our chairman **Mr.S. Meganathan,B.E., F.I.E** for his sincere endeavour in educating us in his premier institution. We would like to express our deep gratitude to our beloved Chairperson **Dr. Thangam Meganathan**, for her enthusiastic motivation which inspired us a lot in completing this project and Vice-Chairman **Mr. Abhay Shankar Meganathan B.E., M.S.**, for providing us with the requisite infrastructure.

We also express our sincere gratitude to our college principal, **Dr.S.N.Murugesan M.E., PhD.**, for his kind support and facilities to complete our work on time. We extend heartfelt gratitude to **Dr.P.Valarmathie, Professor and Head of the Department of Information Technology** for her guidance and encouragement throughout the work. We are very glad to thank our course faculty **Ms Ragavarthini S M.Tech., Assistant Professor** of our department for their encouragement and support towards the successful completion of this project. We extend our thanks to our parents, friends, all faculty members, and supporting staff for their direct and indirect involvement in the successful completion of the project for their encouragement and support

ABSTRACT

In educational institutions, the sharing of study materials plays a crucial role in supporting student learning and academic performance. Although several large-scale platforms like Google Drive and LMS portals exist, the local academic environment often lacks a simple and organized system tailored specifically for students and faculty. To address this gap, our team developed a database-driven application that efficiently manages and organizes study materials in a centralized manner.

The main objective of this project is to allow users to securely upload, search, and download study materials based on their requirements. This system helps maintain structured storage, user authentication, and easy accessibility of academic resources within a department or institution. By providing an efficient and user-friendly platform, this system enhances resource sharing, reduces dependency on informal methods, and improves the overall academic experience.

TABLE OF CONTENTS

Chapter 1 — Introduction

- 1.1 Overview
 - 1.2 Need for the System
 - 1.3 Problem Statement
 - 1.4 Objectives of the Project
 - 1.5 Scope of the Project
 - 1.6 Modules of the System
 - 1.7 Advantages of the System
 - 1.8 Application Areas
-

Chapter 2 — Literature Review

- 2.1 Existing Systems
 - 2.2 Limitations of Current File-Sharing Platforms
 - 2.3 Research Findings
 - 2.4 Summary of Review
-

Chapter 3 — System Design

- 3.1 System Architecture
 - 3.2 Data Flow Diagram (Level 0, Level 1)
 - 3.3 Use Case Diagram
 - 3.4 Class Diagram
 - 3.5 Sequence Diagram
 - 3.6 Activity Diagram
 - 3.7 ER Diagram & Database Schema
-

Chapter 4 — Project Description & Implementation

- 4.1 Technologies Used
- 4.2 Hardware Requirements
- 4.3 Software Requirements
- 4.4 Functional Requirements
- 4.5 Non-Functional Requirements
- 4.6 SQL Table Creation Scripts
- 4.7 JDBC Connection Code
- 4.8 User Authentication Flow
- 4.9 Upload Module Implementation

4.10 Download Module Implementation

4.11 Search Feature Implementation

4.12 Error Handling & Validation

Chapter 5 — Output Screens

5.1 Login Page

5.2 Dashboard

5.3 Upload Study Material Page

5.4 Materials List Page

5.5 Search Results Page

5.6 Download Confirmation

5.7 Database Screenshots

Chapter 6 — Testing

6.1 Test Cases

6.2 Unit Testing

6.3 Integration Testing

6.4 System Testing

6.5 Validation of Results

Chapter 7 — Conclusion & Future Enhancements

7.1 Conclusion

7.2 Limitations

7.3 Future Scope

Chapter 8 — References

CHAPTER 1 – INTRODUCTION

1.1 Overview

The Study Material / Notes Sharing System is a simple and structured platform developed to help students and faculty store, share, and access academic materials efficiently. Traditional methods such as WhatsApp groups or shared cloud folders often become unorganized and difficult to search as materials accumulate. To overcome these challenges, this system provides a centralized and secure repository where users can upload, retrieve, and search notes through a Java application connected to a MySQL database.

1.2 Need for the System

Academic materials shared through informal platforms often become scattered, duplicated, and difficult to locate. There is minimal control over who accesses or modifies files, and important notes may be lost or buried in chats. A dedicated system ensures structured storage, secure authentication, better searchability, and reduced duplication, which improves the overall accessibility and management of study materials.

1.3 Problem Statement

To develop a Java–MySQL based application that enables secure login, organized uploading, efficient searching, and easy downloading of academic materials, ensuring proper authentication and reliable data storage.

1.4 Objectives

The system aims to provide authenticated access, allow study material uploads with essential metadata, enable students and faculty to browse and download notes easily, support keyword-based searching, and maintain structured storage through a MySQL backend.

1.5 Scope

The project covers basic file management features suited for educational institutions. It includes role-based access, document storage, a simple Java GUI, secure database connectivity through JDBC, and organized retrieval of files. It is ideal for small colleges or study groups requiring a structured system without heavy cloud infrastructure.

1.6 Modules

The system includes modules for authentication, uploading of materials, browsing available notes, searching based on keywords, and downloading files. Each module communicates with the database to ensure accurate and real-time data handling.

1.7 Advantages

The system is easy to use, provides centralized file storage, supports fast search, ensures secure login, prevents data duplication, and allows structured metadata storage. It improves organization and reduces the clutter seen in informal sharing platforms.

1.8 Application Areas

This system is suitable for schools, colleges, coaching institutes, study groups, and research teams that require organized storage and quick access to academic documents.

CHAPTER 2 – LITERATURE REVIEW

2.1 Existing Systems

Popular platforms like Google Drive, OneDrive, Dropbox, and LMS tools such as Moodle offer file sharing but are meant for general use rather than academic-focused workflows. They provide cloud storage and sharing features but lack built-in academic categorization, offline usability for low-bandwidth environments, and simple configurations suitable for small institutions.

2.2 Limitations of Existing Platforms

General cloud storage tools do not support academic workflows such as subject-wise organization, faculty–student role separation, or metadata-driven categorization. They require internet connectivity, lack database integration for student learning, and make permission management difficult for institutions. These shortcomings justify the need for a dedicated and lightweight academic notes system.

2.3 Research Findings

Studies in educational technology show that centralized repositories improve consistency, reduce duplication, and make learning resources more accessible. Efficient search mechanisms and metadata-driven indexing greatly enhance retrieval. Institutions also prefer simple and locally deployable systems that do not depend heavily on cloud infrastructure, making lightweight solutions highly effective.

2.4 Summary

Existing platforms are powerful but not tailored for academic use. The proposed system provides structured categorization, metadata storage, efficient search, and database-backed organization, making it ideal for educational environments and DBMS learning contexts.

CHAPTER 3 – SYSTEM DESIGN

3.1 System Architecture

The system follows a three-tier architecture. The presentation layer handles user interaction through Java Swing. The business logic layer manages validation and functional operations. The data access layer interacts with MySQL using JDBC. This separation ensures maintainability and clarity in the system's workflow.

3.2 Data Flow Diagram

At a high level, users interact with the system to upload, view, search, and download materials. Each action communicates with the database for storing or retrieving information, ensuring all operations remain consistent and reliable.

3.3 Use Case Diagram

Students mainly view, search, and download materials, whereas faculty members additionally upload study notes. All users must log in to access the system, ensuring secure authentication.

3.4 Class Diagram

The system consists of classes for database connection, users, materials, services for logic, and DAOs for database operations. Users can upload multiple materials, forming a one-to-many relationship. The layered design improves modularity.

3.5 Sequence Diagram

During an upload action, the user interacts with the UI, which communicates with the service layer. The service then reaches the DAO, which performs the database insertion through the connection class. Responses flow back through each layer to update the UI.

3.6 Activity Diagram

The workflow begins with login validation. Depending on user actions—uploading, searching, viewing, or downloading—corresponding modules interact with the database and file system to complete operations.

3.7 ER Diagram & Database Schema

The database includes Users and Materials tables. Users store login credentials and roles, while Materials store file metadata. A user may upload multiple materials, and foreign keys ensure referential integrity.

CHAPTER 4 – PROJECT DESCRIPTION

4.1 Technologies Used

The system uses Java Swing for the interface, MySQL for structured storage, JDBC for connectivity, and file handling for storing actual documents. IntelliJ IDEA supports development and debugging.

4.2 Hardware Requirements

The system runs on basic hardware with 4GB RAM, dual-core processors, and minimal storage. It works on Windows, Linux, or macOS.

4.3 Software Requirements

Java JDK, IntelliJ or Eclipse, MySQL Workbench, and the MySQL JDBC connector are required to run the project.

4.4 Functional Requirements

The system supports login authentication, file uploading with metadata, file storage and retrieval, keyword search, and viewing of uploaded materials.

4.5 Non-Functional Requirements

It focuses on reliability, ease of use, maintainability through modular design, fast database operations, and strong data integrity using keys and constraints.

4.6 SQL Scripts

Tables for users and materials are created to manage authentication and metadata. They include primary keys, foreign keys, and constraints to ensure consistency.

4.7 JDBC Connection

A reusable Java class establishes a secure database connection using MySQL Connector/J. It centralizes connection logic and ensures stable access to the database.

4.8 Authentication Module

Login validation checks the username and password against the database using prepared statements to prevent SQL injection. Valid users proceed to the dashboard, while invalid attempts are rejected.

4.9 Upload Module

The system copies selected files to a storage folder and inserts metadata into the database. Basic validation prevents empty inputs or invalid selections.

4.10 Download Module

The selected file's path is retrieved from the database and provided to the user, allowing offline access after download.

4.11 Search Module

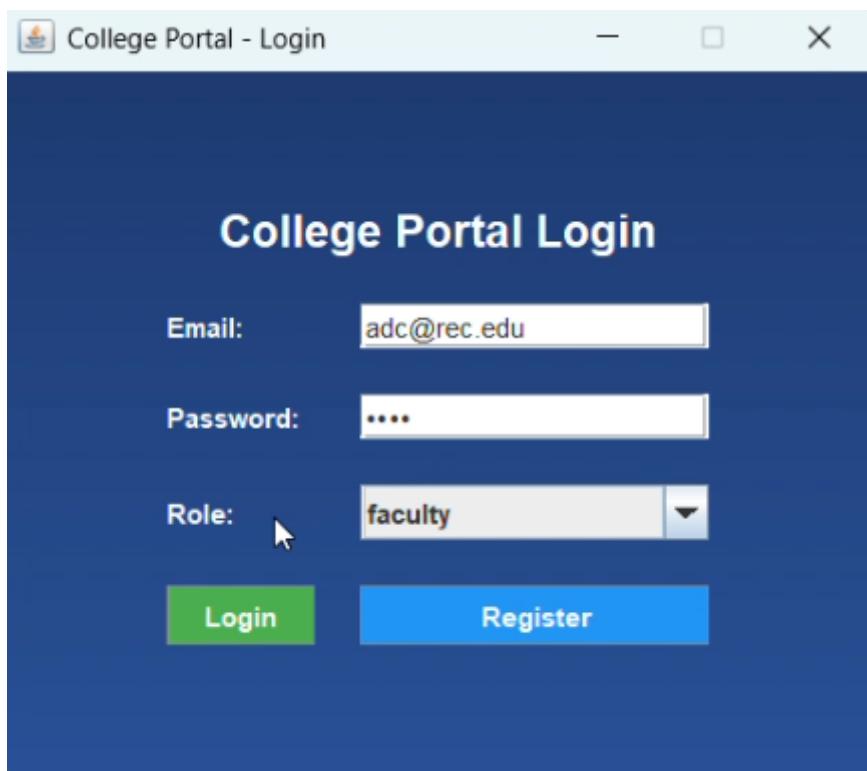
Keyword-based search retrieves matching titles or subjects from the database, making retrieval faster even with many uploaded materials.

4.12 Error Handling

Errors such as missing files, failed database connections, invalid logins, and SQL issues are handled gracefully using exception blocks to prevent program crashes.

CHAPTER 5 – OUTPUT SCREENS

Login Page:



Faculty Dashboard:

A screenshot of a Windows application window titled "Dashboard - adc (faculty)". The title bar shows "Dashboard - adc (faculty)". The main area displays a welcome message "Welcome, adc (faculty)" above a table. The table has two rows: one for "Book1" with subject "DBMS" and another for "111.pdf" with subject "dbms". To the right of the table is a vertical sidebar with five blue buttons: "Upload File", "Delete File", "Add Student", "Remove Student", and "Download File". At the bottom right is a red "Logout" button.

Student Dashboard:

The screenshot shows a web-based student dashboard titled "Dashboard - abc (student)". The main content area displays a table with three rows, each representing a file or resource. The columns are labeled "Title" and "Subject". The data is as follows:

Title	Subject
Book1	DBMS
111.pdf	dbms
bsdb.pdf	ww

On the right side of the dashboard, there are two buttons: a blue "Download File" button and a red "Logout" button. The cursor is visible at the bottom center of the dashboard area.

CHAPTER 6 – TESTING

6.1 Test Cases

Functional testing verified login, upload, search, download, and error messages. All major test cases passed as expected.

6.2 Unit Testing

Individual components such as authentication, upload handling, search operations, and database connectivity were tested separately to ensure accuracy.

6.3 Integration Testing

Integration testing checked how well the combined modules work together.

Tested Integrations

1. Login → Dashboard

- After successful login, user is redirected to the dashboard.

2. Upload Module → Database

- File metadata inserted into **materials** table.
- File physically stored in uploads/ directory.

3. Search → Database → UI

- Search query retrieves correct results and updates UI table.

4. Download → File System

- File path retrieved from database → file copied to user-selected directory.

All integration tests passed without errors.

6.4 System Testing

End-to-end testing validated the entire workflow on multiple operating systems. User acceptance tests confirmed that the system meets essential requirements.

6.5 Validation

Input fields, file formats, login attempts, and SQL queries were validated to ensure secure and correct data handling.

Test Case ID	Module	Input	Expected Output	Actual Result	Status
TC01	Login	Correct username & password	User logged in & dashboard displayed	As expected	Pass
TC02	Login	Incorrect password	Error message: " <i>Invalid Credentials</i> "	As expected	Pass
TC03	Upload	Valid PDF file	File stored in server folder + DB updated	As expected	Pass
TC04	Upload	No file selected	Warning message displayed	As expected	Pass
TC05	Search	Valid title keyword	List of matching materials shown	As expected	Pass
TC06	Search	Non-existing title	“No Results Found” message	As expected	Pass
TC07	Download	Select a valid file	File downloaded/copied to selected path	As expected	Pass
TC08	Database	Insert new material	Record inserted successfully	As expected	Pass
TC09	Error Handling	MySQL off / no connection	“Database Connection Failed” message	As expected	Pass
TC10	Validation	Empty title during upload	Input validation error	As expected	Pass

CHAPTER 7 – CONCLUSION & FUTURE SCOPE

7.1 Conclusion

The project successfully provides a simple and secure platform for managing academic materials. It demonstrates practical implementation of Java, JDBC, MySQL, authentication, CRUD operations, and file handling. The system is suitable for small institutions and effectively organizes study resources.

7.2 Limitations

Despite achieving its goals, the system still has certain limitations:

- **✗ No subject categorization** – All materials are stored under a single category, making large repositories harder to navigate.
- **✗ Lack of an admin dashboard** – Admin cannot manage users, track uploads, or monitor system activities.
- **✗ No preview option** – Files must be downloaded to view; no in-system preview for PDFs, images, or documents.
- **✗ Minimal UI** – The system uses a simple console/Java GUI, not a modern web or mobile interface.
- **✗ No version control for files** – Re-uploading files overwrites old ones without history tracking.

for files. These limitations restrict scalability for large repositories.

7.3 Future Enhancements

Future upgrades may include subject-wise organization, file previews, role-based access, cloud storage integration, analytics features, and a modern web or mobile interface to improve usability.

CHAPTER 8 – REFERENCES

1. Oracle JDBC Documentation

Oracle. *Java Database Connectivity (JDBC) Guide*. Available at: <https://docs.oracle.com/javase/> (Accessed 2025).

2. Java File I/O Tutorials

Oracle. *Java Platform Standard Edition – File Input/Output*. Official Java Tutorials for handling files and streams.

3. MySQL User Manual

Oracle Corporation. *MySQL 8.0 Reference Manual*. Covers SQL commands, database creation, and server configuration.

4. Research Articles on LMS and E-Learning

Various authors. Studies on Learning Management Systems, digital repositories, and academic content sharing platforms.

Includes research on:

- Centralized academic storage
- Digital resource retrieval
- Metadata usage in educational systems

5. JDBC and MySQL Connector/J Guide

Oracle Corporation. *MySQL Connector/J Developer Guide*. Explains JDBC–MySQL integration and driver usage.

6. Software Engineering & System Design Books

Sommerville, I. *Software Engineering*. Concepts on requirements, testing, and system architecture used for designing modules.

7. W3Schools Java & SQL Tutorials

Online reference for Java basics, JDBC examples, and SQL commands.