



# Study Material Sharing Platform

Atharva A. Bhosale<sup>1</sup>, Dewank B. Badiger<sup>2</sup>, Sameer Bisen<sup>3</sup>, Adityaraj Godbole<sup>4</sup>,

Smita Chunamari<sup>5</sup>

Department of Computer Engineering, Maharashtra, India<sup>1-5</sup>

**Abstract:** The Study Material Sharing Platform is a web-based application developed to provide a centralized, secure, and user-friendly environment for managing academic resources. The platform addresses common issues faced by students and faculty, such as scattered study materials, outdated files, poor discoverability, and a lack of quality control. It enables students to upload, download, bookmark, rate, and review materials, while faculty members can organize and manage content efficiently, and administrators can supervise users and resources. The proposed system follows a modern client-server architecture with a React.js front end and Supabase as the back-end-as-a-service platform, providing a PostgreSQL database for structured metadata storage, built-in authentication, and secure cloud file storage. Major features include role-based access control, subject-wise categorization, keyword search and filters, material versioning, notifications, and discussion support. Testing demonstrated strong reliability and usability, with a System Usability Scale (SUS) score of 77.2, placing it well above average usability benchmarks. The project demonstrates how a focused academic repository can improve accessibility, collaboration, and productivity in engineering education.

**Keywords:** Study Material Sharing, Web Application, React.js, Supabase, PostgreSQL, Role-Based Access Control, Collaborative Learning.

## I. INTRODUCTION

The rapid proliferation of the internet and digital technologies has fundamentally transformed the landscape of higher education. Today's students increasingly depend on digital resources to supplement classroom learning, prepare for examinations, and deepen their understanding of complex subjects. Despite this shift, the infrastructure for sharing and managing academic study materials remains fragmented and inefficient across many educational institutions.

At the institutional level, study materials typically circulate through informal channels such as WhatsApp groups, email threads, and Google Drive folders shared without any structured organization. These ad-hoc solutions lack proper versioning, access control, and search functionality. This paper addresses this gap by presenting a Study Material Sharing Platform tailored to the needs of engineering students and faculty, minimizing duplicated effort and standardizing resource quality.

### A. Problem Statement

The current methods of academic study material distribution in engineering colleges suffer from critical deficiencies, including a lack of centralization, absence of version control, and no quality assurance mechanism. Furthermore, limited accessibility via expiring links, security concerns over intellectual property, poor discoverability without metadata, and zero contextual collaborative discussion features severely hamper students' learning experiences.

### B. Objectives

The platform is developed to achieve the following nine main objectives:

Table I. Objectives of the Study Material Sharing Platform

Sr. No.	Objective
1	Design and develop a centralized web-based platform for organizing and sharing academic study materials.
2	Implement a robust Role-Based Access Control (RBAC) system supporting Student, Faculty, and Administrator roles.
3	Provide intelligent search, filtering, and categorization based on subject, semester, and course.
4	Incorporate document version control to ensure access to the latest study materials.
5	Develop a quality assurance mechanism through user ratings and peer reviews.
6	Integrate a discussion forum for contextual academic discussions tied to materials.
7	Implement secure authentication using Supabase Authentication (JWT and OAuth 2.0).
8	Develop a material recommendation feature based on user engagement patterns.



9	Ensure scalability, performance, and cross-platform compatibility.
---	--

### C. Scope

In-Scope Features: User registration, material upload/download (PDF, DOCX, PPTX), hierarchical categorization, full-text search, bookmarking, ratings/reviews, notifications, role-based access control, and a basic recommendation engine.

Out-of-Scope Features: Live video streaming, real-time collaborative document editing, automated AI-based content summarization, and native mobile application development are reserved for future scope.

## II. LITERATURE SURVEY

### A. Existing Systems and Platforms

Google Classroom: Introduced by Google, it is widely adopted for assignment management and teacher-student communication. However, it is primarily assignment-centric and lacks dedicated features for peer-to-peer study material sharing, advanced cross-course search functionality, and community rating systems.

Moodle: An open-source Learning Management System (LMS) deployed universally. While comprehensive, Moodle's interface is notoriously complex. Its study material repository features require significant administrative overhead and lack modern intuitive UI/UX design principles.

Scribd and SlideShare: These serve as broad document sharing platforms. However, Scribd operates on a paid subscription model, and SlideShare focuses heavily on presentation slides. Neither is tailored for the organized, semester-wise academic context of a specific engineering institution.

Course Hero and StudyLib: These platforms aggregate materials from multiple institutions globally. They are heavily commercialized, locking premium content behind paywalls, and often raise academic integrity concerns.

### B. Comparative Study

When evaluated against the needs of a modern engineering college, existing tools fall short in specific intersections. The proposed Study Material Sharing Platform bridges this gap by offering free, peer-driven, institutionally categorized material sharing with strict version control and quality ratings.

Table II. Comparative Summary of Existing Platforms and the Proposed System

Platform	Primary Strength	Observed Limitation	Suitability
Google Classroom	Simple assignment management and teacher-student communication	Limited peer-to-peer sharing and weak cross-course search	Moderate
Moodle	Comprehensive LMS capabilities	Complex UI and higher administrative overhead	Moderate
Scribd / SlideShare	Large public document repositories	Generic platforms; not institution-specific	Low
Course Hero / StudyLib	Large user-generated academic collections	Commercialization and academic integrity concerns	Low
Proposed Platform	Centralized repo with role control, ratings, and contextual discussions	Focused initially on web deployment only	High

### C. Research Gap

The literature survey indicates that no existing free platform seamlessly combines institutional focus with comprehensive peer-sharing capabilities, version control, and contextual discussion threads. Current systems either prioritize assignment management over knowledge exchange or commercialize the shared content. There is a clear need for a focused, open-access, hierarchical repository for engineering students.

## III. SYSTEM ANALYSIS

### A. Feasibility Study

Technical Feasibility: The project relies on React.js and Supabase (PostgreSQL), which are robust, well-documented open-source technologies capable of handling the required metadata and file storage safely.



**Economic Feasibility:** Leveraging the Supabase free-tier and hosting platforms like Vercel ensures that initial deployment and testing occur at zero software licensing costs.

**Operational Feasibility:** The target users are highly tech-literate engineering students and faculty, ensuring a minimal learning curve for the modern web interface.

### B. Functional Requirements

- FR-01: Secure registration and login using JWT for session management.
- FR-02: Role-Based Access Control (RBAC) for Students, Faculty, and Admins.
- FR-03: Advanced search/filtering and file management (Upload/Download).
- FR-04: Quality assurance through ratings and contextual discussion forums.

### C. Non-Functional Requirements

- Performance: Search queries must return results in under 3 seconds.
- Security: Use of password hashing and data sanitization to prevent malicious attacks.
- Scalability: The system should support up to 1,000 concurrent users.
- Usability: The interface must be intuitive and require minimal training.

## IV. SYSTEM DESIGN

### A. System Architecture

The platform follows a modern client-server architecture. The React.js Single Page Application (SPA) communicates with Supabase via RESTful API calls and real-time subscriptions. Supabase provides a PostgreSQL relational database, cloud file storage buckets, and built-in JWT-based authentication. The SPA is deployed on Vercel's global edge network, ensuring low-latency delivery worldwide.

*Fig. 1. High-Level System Architecture of the Study Material Sharing Platform*

### B. Database Design

The system utilizes a structured relational mapping approach in PostgreSQL. Key entities include:

*Table III. Core Database Entities and Their Purpose*

Entity / Table	Purpose	Key Attributes
Users	Stores user profile and academic metadata	Role, department, semester, authentication linkage
Materials	Stores uploaded resource metadata	Title, file URL, type, size, course, tags, version, uploader ID
Bookmarks & Ratings	Connects users with saved and rated materials	Bookmark state, star value, review text, timestamps
Discussions	Supports threaded conversation on materials	Parent-child comment links, material reference, timestamps

### C. Module Design

The application is modularized into discrete functional components:

*Table IV. Major Application Modules*

Module	Responsibility
Authentication Module	Handles user onboarding, token-based authentication, and session persistence.
Material Management Module	Validates uploads and manages document storage through Supabase buckets.
Search and Discovery Module	Supports efficient filtering and retrieval using PostgreSQL search features.
User Interaction Module	Processes bookmarks, ratings, reviews, and discussion-related actions.

### D. User Interface Design

The user interface follows a clean, card-based design language. Key screens include the Dashboard (subject-wise



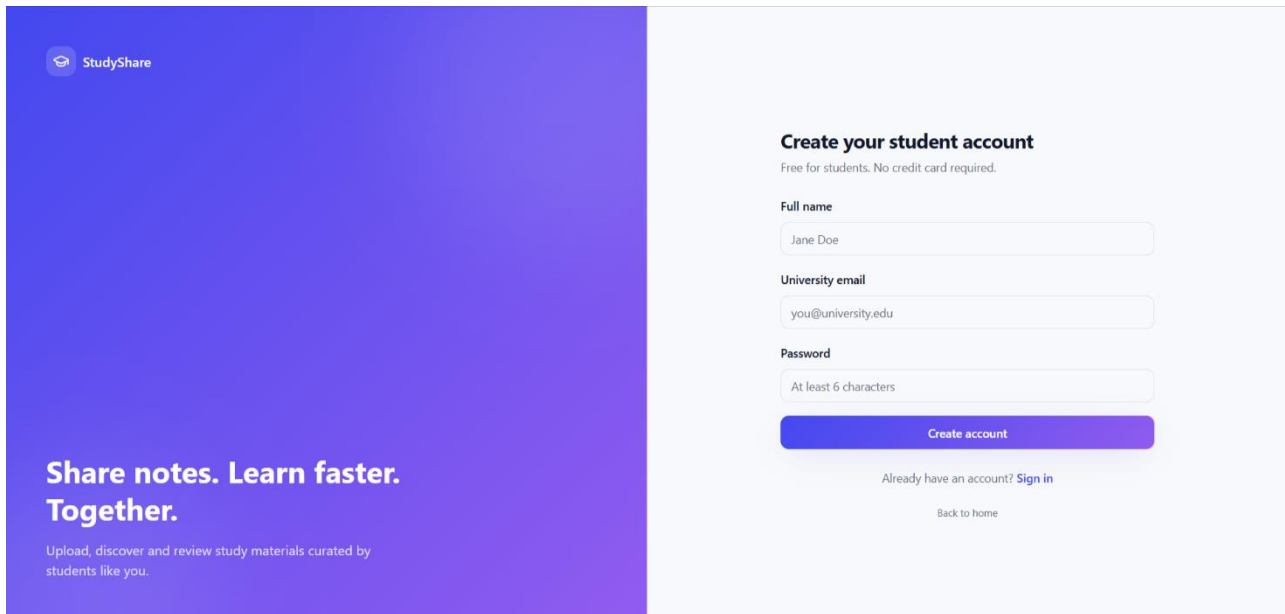
categorized material feed), Upload Portal (drag-and-drop with metadata tagging), Material Detail Page (preview, download, ratings, and discussion thread), and Admin Panel (user management, flagged content review).

*Fig. 2. Screenshot of the Platform Dashboard showing subject-wise material categorization*

*Fig. 3. Upload Portal with drag-and-drop interface and metadata tagging*

## V. IMPLEMENTATION

### A. Development Environment



The front-end application was scaffolded using Vite for fast Hot Module Replacement (HMR) and optimized builds. Node.js serves as the runtime environment for local development. Supabase was configured via its web console to set up the PostgreSQL database schema, storage buckets, and authentication providers.

### B. Front-End Implementation

React Router was implemented to manage protected and public routes dynamically. Component-based UI architecture was enforced, breaking down complex pages into reusable components such as MaterialCard, SearchBar, and CommentThread. TanStack Query was utilized for data fetching, caching, and background synchronization, ensuring the UI always reflects the current database state.

### C. Back-End and Database Implementation

Instead of building a traditional Node.js/Express backend, the system leverages Supabase's auto-generated APIs. The PostgreSQL database schema was constructed using raw SQL statements to define tables, primary keys, and foreign keys. Supabase Storage buckets store PDF, DOCX, and PPTX files, linking the resulting public URLs to the Materials metadata table.

### D. Security and Access Control

Security is implemented at the database level using PostgreSQL Row Level Security (RLS) policies. These policies ensure users can only mutate their own records while allowing public read access to approved materials. Supabase Auth handles secure JWT token issuance and validation for all API requests.

*Fig. 4. Material Detail Page showing rating, reviews, and discussion thread*

## VI. TESTING AND RESULTS

### A. Testing Strategy

The testing phase followed a structured V-Model approach, incorporating comprehensive Unit Testing, Integration Testing, and User Acceptance Testing (UAT). Front-end logic was tested using Vitest and React Testing Library to ensure component stability and correct rendering behavior.

### B. Test Results

A total of 87 specific unit test assertions were written and executed across key modules, achieving a 100% pass rate.



Integration testing verified seamless data flow between the React front end and Supabase backend. UAT was conducted with a cohort of 50 student participants and 10 faculty members.

Table V. Testing Summary and Observed Results

Test Aspect	Observation / Result
Unit Testing	87 assertions executed with 100% pass rate.
Integration Testing	Primary end-to-end workflows passed without data loss or corruption.
UAT Participation	50 students and 10 faculty members participated.
Usability Score	System Usability Scale (SUS) score: 77.2 (Good to Excellent).
Performance	Search and major interactions remained under 3 seconds under normal use.
Concurrent Load	System remained stable up to approximately 500 simulated users.

The platform achieved an overall SUS score of 77.2, categorizing its usability as "Good" to "Excellent." Performance metrics indicated response times remained under 3 seconds even when simulating concurrent user loads of up to 500 simulated users.

## VII. CONCLUSION AND FUTURE SCOPE

The Study Material Sharing Platform successfully modernizes the academic resource distribution process for engineering students. By fulfilling all nine primary objectives, the project established a secure, centralized, and highly usable repository. The implementation of Role-Based Access Control, robust PostgreSQL metadata management, and contextual discussion threads directly solves the fragmentation and quality-control issues prevalent in informal sharing networks.

While the web-based SPA is highly responsive, the lack of a native mobile application limits accessibility for users requiring background downloads or push notifications. Future enhancements include the development of a native mobile application using React Native, integrating AI-powered features for automated document summarization, and implementing formal LMS integrations (such as LTI standard compliance for Moodle or Canvas).

## REFERENCES

- [1] Google LLC, "Google Classroom - Learn Without Limits," 2014. [Online]. Available: <https://classroom.google.com>. [Accessed: Apr. 2026].
- [2] M. Dougiamas and P. C. Taylor, "Moodle: Open Source Course Management System," in Proc. ED-MEDIA, 2003.
- [3] LinkedIn Corporation, "SlideShare - Share What You Know," 2006. [Online]. Available: <https://www.slideshare.net>. [Accessed: Apr. 2026].
- [4] Course Hero Inc., "Course Hero - Academic Success Platform," 2006. [Online]. Available: <https://www.coursehero.com>. [Accessed: Apr. 2026].
- [5] S. Khan, "Let's Use Video to Reinvent Education," TED Talks, 2011. [Online]. Available: <https://www.ted.com>. [Accessed: Apr. 2026].
- [6] Supabase Inc., "Supabase Documentation," 2023. [Online]. Available: <https://supabase.com/docs>. [Accessed: Apr. 2026].
- [7] Meta Platforms Inc., "React - JavaScript Library," 2013. [Online]. Available: <https://react.dev>. [Accessed: Apr. 2026].
- [8] TanStack, "TanStack Query Documentation," 2020. [Online]. Available: <https://tanstack.com/query>. [Accessed: Apr. 2026].
- [9] Vite, "Vite Documentation," 2020. [Online]. Available: <https://vitejs.dev>. [Accessed: Apr. 2026].
- [10] Ministry of Education, Government of India, "National Education Policy 2020," [Online]. Available: <https://www.education.gov.in/nep/about-nep>. [Accessed: Apr. 2026].