



Student Resource Sharing Hub

Your name, course, and university

“A collaborative platform for students to share and access resources”

Problem Statement

Students face difficulties accessing quality study resources.

Many resources are scattered across platforms (WhatsApp, Google Drive, Telegram, etc.).

No centralized, secure, and easy-to-use hub for sharing.



Project Objective

To build a **centralized digital hub** for students to:

Share notes, e-books, question papers, and tutorials.

Collaborate and interact through discussions.

Access resources anytime, anywhere.

Features of the Hub

Upload & Download

Students can upload notes, PDFs, PPTs.

Search & Filters

Easy resource discovery.

User Profiles

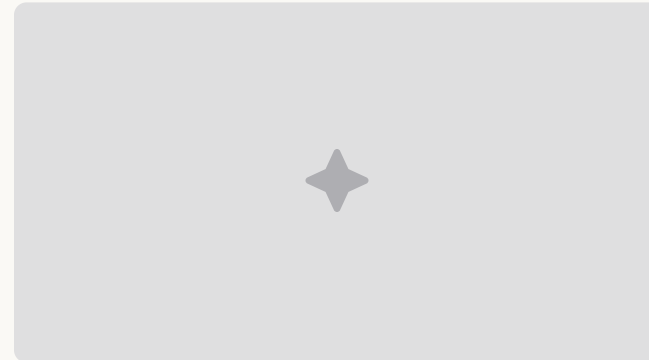
Contributors recognized with points/badges.

Categorization

Resources organized by subject, course, and year.

Discussion Forum / Q&A

Peer-to-peer interaction.



Technology Stack



Frontend: React.js (UI)



Backend: Python (Flask / Django)



Database: SQLite / PostgreSQL



Hosting: GitHub Pages / Heroku / Render



Authentication: Firebase or Django auth

Methodology / How I'll Build It

01

Requirement Analysis

Define scope and features.

02

System Design

ER diagram, flow of resources.

03

Backend Setup

Database schema for resources, users.

04

Frontend Development

Upload, search, and display pages.

05

Testing & Deployment

Check usability and host online.

Benefits



Easy access to academic resources.



Encourages **knowledge sharing & collaboration.**



Helps juniors learn from seniors' notes.



Saves time and promotes digital learning.

Future Scope

- Mobile app version (Flutter/React Native).
- AI-based recommendation system (suggesting notes/resources).
- Integration with online classrooms (Zoom/Teams).
- Gamification (leaderboard for top contributors).



Conclusion

The **Student Resource Sharing Hub** is a step toward collaborative education.

Scalable project with **academic + career relevance**.

Future scope for research & product development.

ROADMAP

Step 1: Define the Scope (MVP – Minimum Viable Product)

Instead of making a huge system right away, start with basic features:

- Student login/signup
- Upload notes (PDF, DOC, images, PPT)
- Browse/download notes
- Search by subject/title

That's enough for version 1. Later, you can add reviews, ratings, tags, plagiarism checks, etc.



◆ Step 2: Choose the Tech Stack

Since you're a beginner, go with simple but scalable tools:

- **Frontend:** HTML, CSS, Bootstrap (for styling), maybe React later
- **Backend:** Python (Flask or Django – Django is better for user management)
- **Database:** SQLite (for starting) → MySQL/PostgreSQL (future)
- **Storage:** Save files locally (start small) → Cloud storage like AWS S3 later