EduDoc – Charter Document

# Purpose and Objective

**The Notes Management System** is designed to empower teachers with a streamlined platform to **create, organize, and store educational notes**. It ensures **easy categorization** by chapter and topic and supports **scanned uploads** for existing handwritten or printed material. The goal is to enhance teaching effectiveness and enable better classroom planning through structured note organization.

# Project Justification

* **Demand-Based**: Teachers often manage **scattered notes** across multiple formats. This platform consolidates and organizes them in one place.
* **Opportunity-Based:** Existing systems focus on student-facing content. A backend-only system **dedicated to teacher** note management is rare and highly needed.

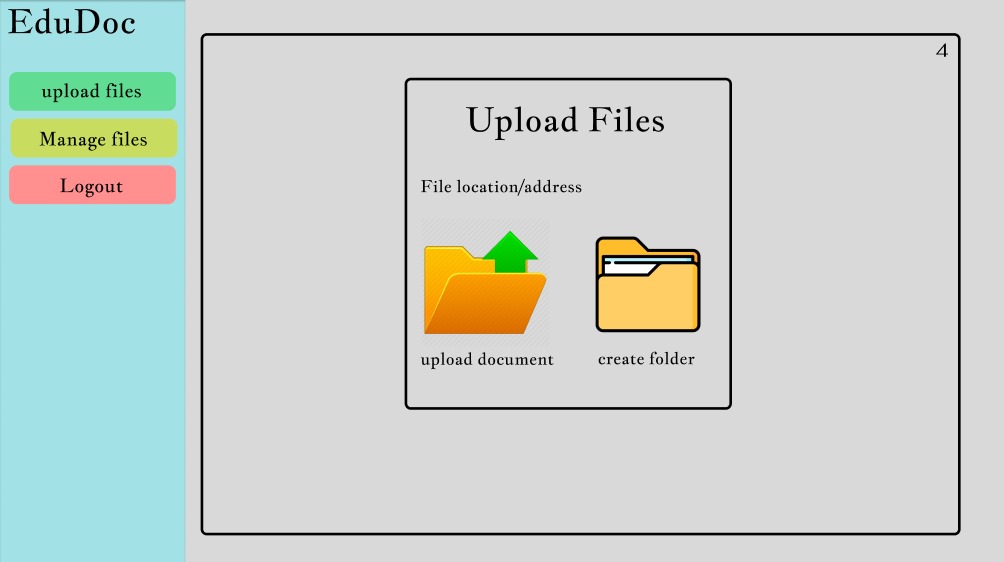
# Business Requirements

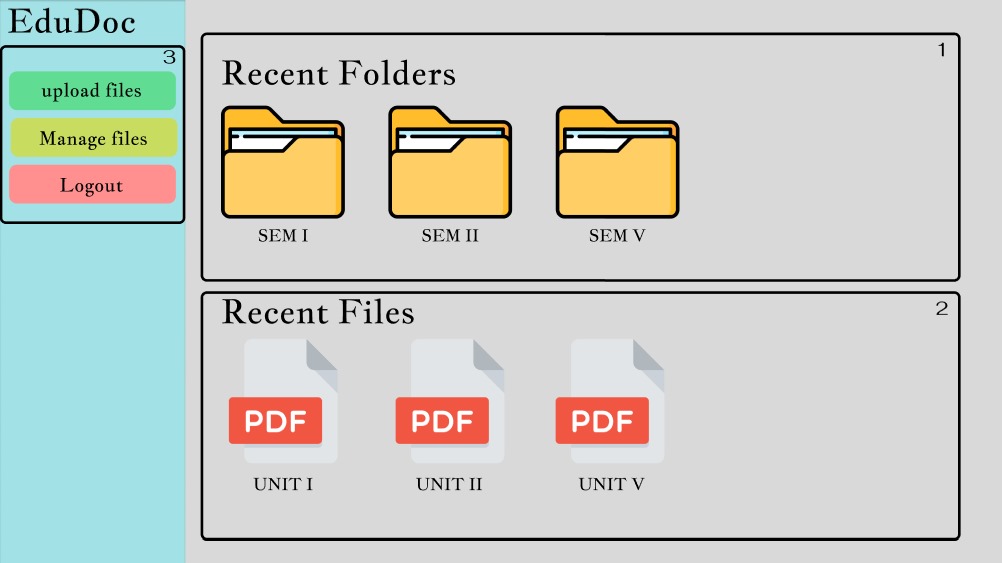
**Teachers must be able to**:  
 - Create notes using a rich-text editor.  
 - Structure notes under subjects → chapters → topics.  
 - Add sections under each topic for “Class Practice Problems” and “Homework Problems.”  
 - Upload scanned notes (PDF/Image) and tag them to topics.  
 - Search and filter notes by keywords, chapter, or topic.  
 - Pin or favorite frequently used notes.  
 - View version info (last updated time).  
 - Auto-save drafts during editing.  
 - Toggle light/dark mode for comfort.

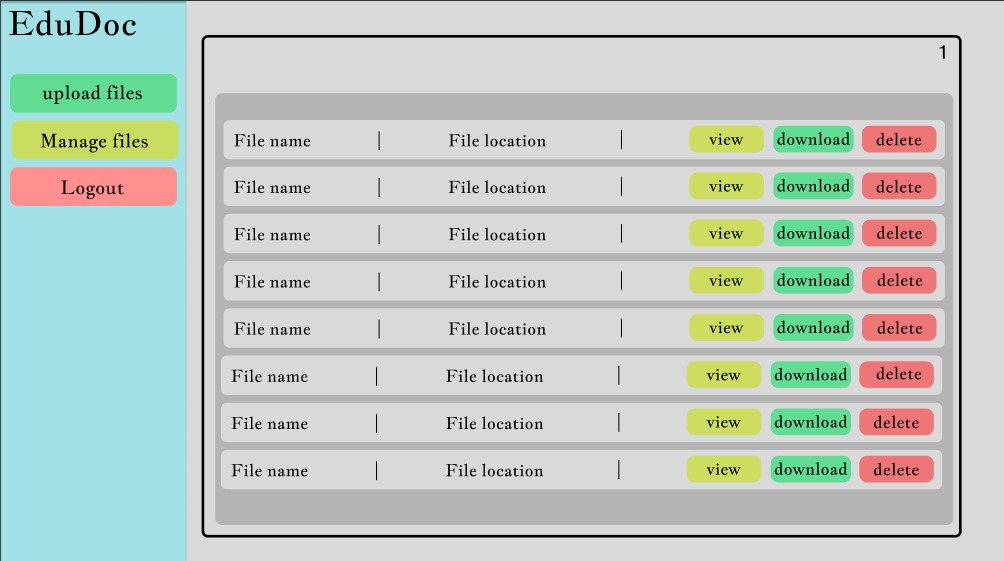
# Technical Requirements

- User-friendly and responsive UI.  
- Fast loading and seamless performance.  
- Basic authentication for teacher login.  
- Secure file upload and storage.  
- Local auto-save using browser’s localStorage.  
- Efficient text search and topic filtering.  
- Clean and minimal interface design.

* **UI-Ux Wireframes**







# Stakeholders

* **Initial Development Phase:**  
  - Developers: Ojas Gharde, Kanishka Garud, Yash Kannawar  
  - Testers: Akhilesh Ukey, Siddhi Chapre  
   - Supervisor/Guide: Jayesh Raut ,Anuradha Pawar

# Technical Stack

- Frontend: React.js (with Tailwind CSS)  
- Backend: Firebase (for storage and auth) or Node.js + Mysql  
- Storage: Firebase Storage  
- PDF/Image Support: jsPDF, Cloudinary (optional)  
- Editor: Quill.js / TinyMCE  
- Hosting: Docker

# Project ROI

- Centralized note management for teachers.  
- Time-saving structure and auto-saving mechanism.  
- Improved lesson planning with reusable, organized content.  
- Scanned note digitization reduces physical clutter.

# Resources Required

* Human Resources:  
  - Developers: 1–2  
  - Testers: 1 (if needed)



* Technical Resources:  
  - Hosting Platform: Docker   
  - Editor Library: Quill.js  
  - Image/PDF Support: File Uploads + Preview  
  - Database: Firebase Firestore / MySql   
  - Version Control: GitHub
* **Risk Analysis**

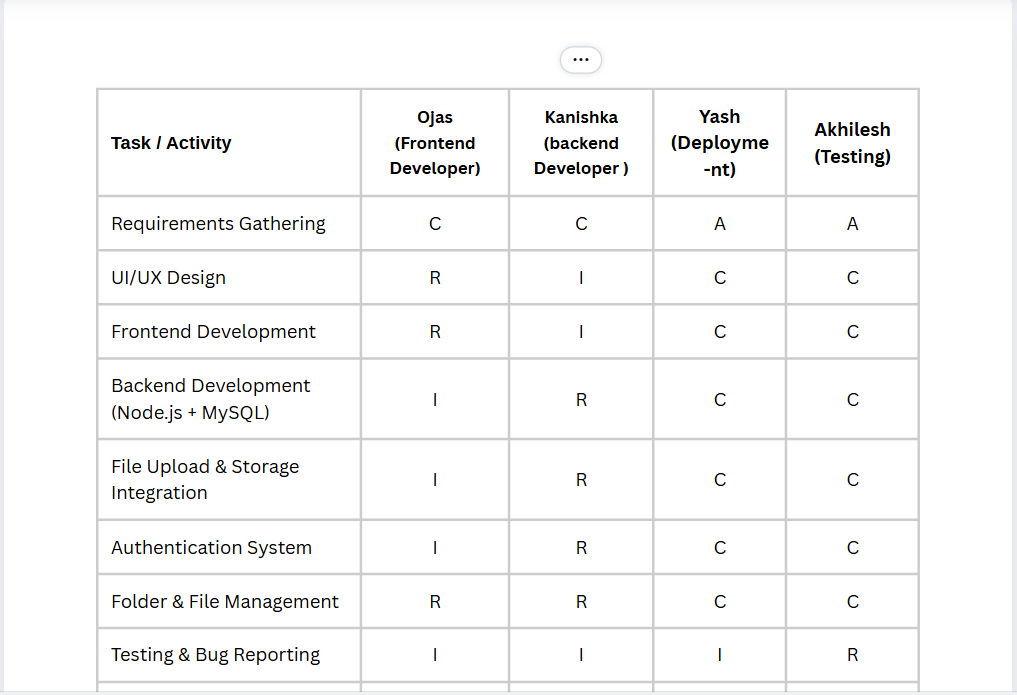
|  |  |  |  |
| --- | --- | --- | --- |
| Risk | Impact | Likehood | Mitigation Strategy |
| Storage Overflow | Medium | Medium | Compress files before upload, set limits |
| Loss of Work (Session Timeout) | High | Mediun | Use local Storage auto-save |
| Image Load Failures | Medium | Medium | Use stable libraries, test image formats |
| Performance Lag with Large Notes | Medium | Low | Use lazy loading, paginate content |

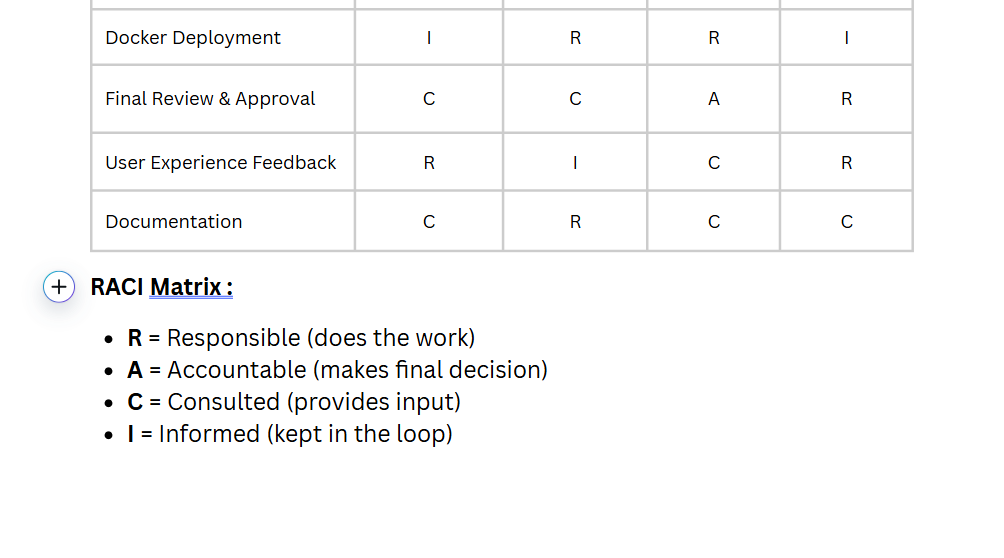
# Timeline & Milestones and Gantt Chart

|  |  |  |  |
| --- | --- | --- | --- |
| Phase | Milestone | Tasks | Duration |
| Requirement Gathering | Finalize scope & features | List key sections and mockups | 2 days |
| UI/UX Design | Build wireframes | Design layout and flows | 3 days |
| Development | | Core functionality | Build editor, upload, tag & save logic | 1 week |
| Testing | Bug fixing & improvements | Validate all workflows | 3 days |
| Finalization | Cleanup & deploy | Optimize UI, deploy to Docker | 3 days |

# 

* **RACI Matrix**





# Concept Maps

