📘Digital Library Web Application

1. Project Overview

The Digital Library is a Flask-based web application designed to help students and educators share study materials online. Users can upload files like PDFs, Word documents, PowerPoint presentations, and images, categorized by subject and topic. Other users can search, sort, download, or delete materials based on their needs.

2. Problem Statement

Many students face difficulties in accessing quality study materials on time. Traditional means like physical sharing or messaging apps are inefficient for organized distribution. Our goal is to create a centralized digital platform where educational resources can be uploaded, categorized, searched, and retrieved quickly.

3. Solution Overview

Problem	Our Solution
Scattered file sharing	Centralized platform with categorized uploads
No search/filtering capability	Advanced search, filtering (subject, type, category), and sorting options
Unregulated formats and sizes	Secure upload system with allowed file types and max size limits
Lack of accessibility	Clean, mobile-responsive Bootstrap UI with simple navigation
Manual retrieval of old notes	Date-based sorting and easy download access anytime

Key Features of the Digital Library

📤 1. Upload Notes

- Users can upload study materials (PDFs, Word Docs, Presentations, Images).
- Each upload includes metadata: **Title**, **Subject**, **Category**.
- File type and size are validated before saving.
- Files are securely stored in the server’s `uploads` folder.

🔍 2. Search, Filter & Sort

- Users can **search notes by subject**.
- Filter options:
    - **File Type** (e.g., PDF, DOCX, PPTX, JPG)
    - **Category** (e.g., Notes, Assignments, Presentations)
- Sort options:
    - **Title** (A–Z / Z–A)
    - **File Size** (Smallest / Largest)
    - **Upload Date** (Newest / Oldest)

🧾 3. Notes Display

- All uploaded notes are shown as **Bootstrap cards**.
- Each card displays:
    - Title, Subject, Category
    - File type and size
    - Upload timestamp
- Notes are styled for readability over a dark transparent overlay.

⬇️ **4. Download Functionality -** Users can download any note directly with a click.

❌ **5. Delete Functionality** - Deletion removes both the file from storage and the record from the database.

✅ 6. Responsive & User-Friendly UI

- Clean layout using **Bootstrap 5**
- Mobile-responsive design

🔔 7. Flash Messages

- Clear feedback after actions (upload, delete, error)
- Helps users know the success/failure of their operations instantly

---

🔧 Technical Stack Overview

✅ Frontend

- **Languages & Libraries:** HTML5, CSS3, Bootstrap 5
- **Templating Engine:** Jinja2 (used to dynamically display notes, filters, and alerts)
- **UI Features:**
  - Responsive layout using Bootstrap grid system
  - Notes displayed as clean, styled cards with download/delete buttons
- **Visual Enhancements:**
  - Transparent overlays for readability
  - Custom button and card styles

🖥️ Backend

- **Framework:** Flask (Python-based micro web framework)
- **Routing Logic:**
  - `/upload` – Lets users upload new notes (with checks for file type and size)
  - `/uploads/<filename>` – Allows users to securely download a file
  - `/delete/<id>` – Deletes a note and removes its file from storage
- **File Handling:**
  - File type/size validation (only allows PDF, DOCX, PPT, JPG, PNG, etc.)
  - Files stored in `uploads/` directory
- **Security:**
  - Flash messages for user feedback
  - Upload size limit (16MB)

🗄️ Database

- **Engine:** SQLite (lightweight, serverless)
- **ORM:** SQLAlchemy
- **Schema: `Note` model**
  - `title`, `subject`, `category` – Metadata
  - `filename`, `file_type`, `file_size` – File details
  - `uploaded_at` – Timestamp (auto-added)

✅ Progress So Far

- Full upload-download-delete cycle with metadata
- Search, filter, sort functionalities
- Flash messaging for feedback
- Responsive and attractive UI using Bootstrap

🔄 In Progress

- **Authentication System (Login/Logout)**
- **Admin-Only Delete Access**
- **Download Statistics to See Popular Resources**
- **Comments or Ratings for Notes**
- **Tagging System for Smarter Categorization**
- **Switch to PostgreSQL or Cloud Storage (e.g., Firebase, S3) for Scalability**

Conclusion

This project allowed us to apply full-stack development skills in a real-world academic context. We implemented secure file uploads, user-centric search and filtering, and integrated backend logic with a sleek interface. The Digital Library has the potential to grow into a robust educational platform with further enhancements.