Name of student	Soham Satpute
Class/Roll no	D15A_51
D.O.P	20/03/25
D.O.S	27/03/25
Sign and Grade	

StoreIT

Project Description:

StoreIt is a cloud-based file storage and management platform that allows users to securely upload, organize, and share files via a responsive web interface. Inspired by platforms like Google Drive, StoreIt is built with **React** on the frontend and **Flask** on the backend, using **MongoDB** and **GridFS** for data and file storage. Key features include OTP-based user authentication, real-time storage usage tracking, categorized file views (Documents, Images, Media, Others), and shareable download links.

Designed with scalability and modularity in mind, the platform supports secure data transactions, efficient file handling, and a user-friendly dashboard. StoreIt ensures fast performance, strong security practices, and a responsive UI for seamless user interaction across all devices.

Requirement Gathering:

Functional Requirements:

- **User Authentication:** Secure OTP-based login with JWT session handling.
- **File Management:** Upload, delete, preview, and download categorized files.
- **Dashboard:** Real-time storage analytics and recent file activity display.
- **Sharing:** Generate public or restricted shareable links for files.

Non-Functional Requirements:

- **Security:** Encrypted communication, token-based auth, and OTP validation.
- **Scalability:** Modular architecture supporting future enhancements.
- **Responsiveness:** Optimized UI for various screen sizes.
- **Performance:** Fast file operations using GridFS and async processing.
- **Maintainability:** Clean code structure with RESTful APIs and component-based design.

System Requirements:

1. Hardware Requirements:

- Processor: Intel Core i5 / AMD Ryzen 5 or higher (dual-core, 2.0 GHz or faster)
- RAM: Minimum 8GB (16GB recommended)
- Storage: At least 1GB free space (256GB SSD recommended)
- Network: Stable internet connection (for MongoDB Atlas)

2. Software Requirements:

- Operating System: Windows 10/11, macOS 10.15+, or Ubuntu 20.04+
- Code Editor: Visual Studio Code or compatible IDE
- Version Control: Git 2.25+
- Python: Version 3.8 or higher
- Node.js: LTS version

Technologies Used:

Development	VS Code , Postman , Git
Frontend	HTML/CSS/Typescript(or Streamlit/Flask Templates)
Backend	Flask (Python 3.8+)
ML Model	Scikit-learn
Styling	SCSS / Bootstrap
APIs	RESTful Flask APIs

Setup Instructions:

1. Python 3.8+ & Flask Backend

- Install Python from https://www.python.org/downloads/

- Create virtual environment: python -m venv venv

- Activate environment and install dependencies: pip install -r requirements.txt

- Run Flask server: python app.py

2. MongoDB Setup

- Use MongoDB Atlas or install MongoDB locally

- Obtain MongoDB URI and configure it in the Flask backend

3. React Frontend

- Install Node.js from https://nodejs.org/

- Navigate to frontend folder and run: npm install

- Start dev server: npm run dev

Backend Setup:

1. Navigate to backend folder:
cd project

2. (Optional) Create a virtual environment:

python -m venv venv

venv\Scripts\activate # For Windows

3. Install dependencies:
pip install -r requirements.txt

4. Start the Flask server:
cd api
python app.py

Backend will run at: http://localhost:5000

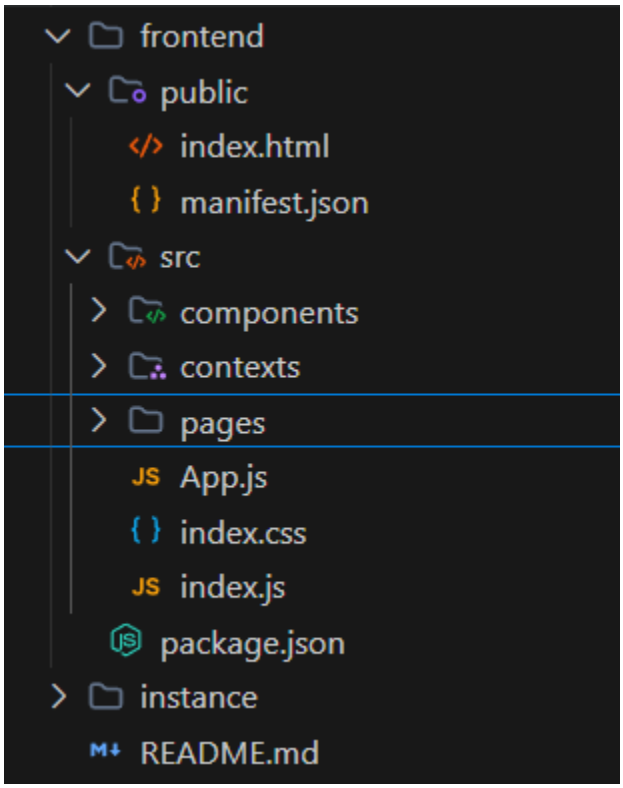
Frontend Setup :

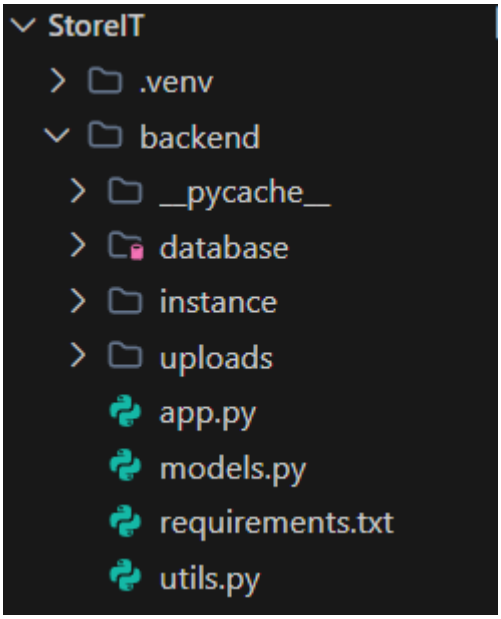
1. **Navigate to frontend folder:**
cd project

2. **Install dependencies:**
npm install

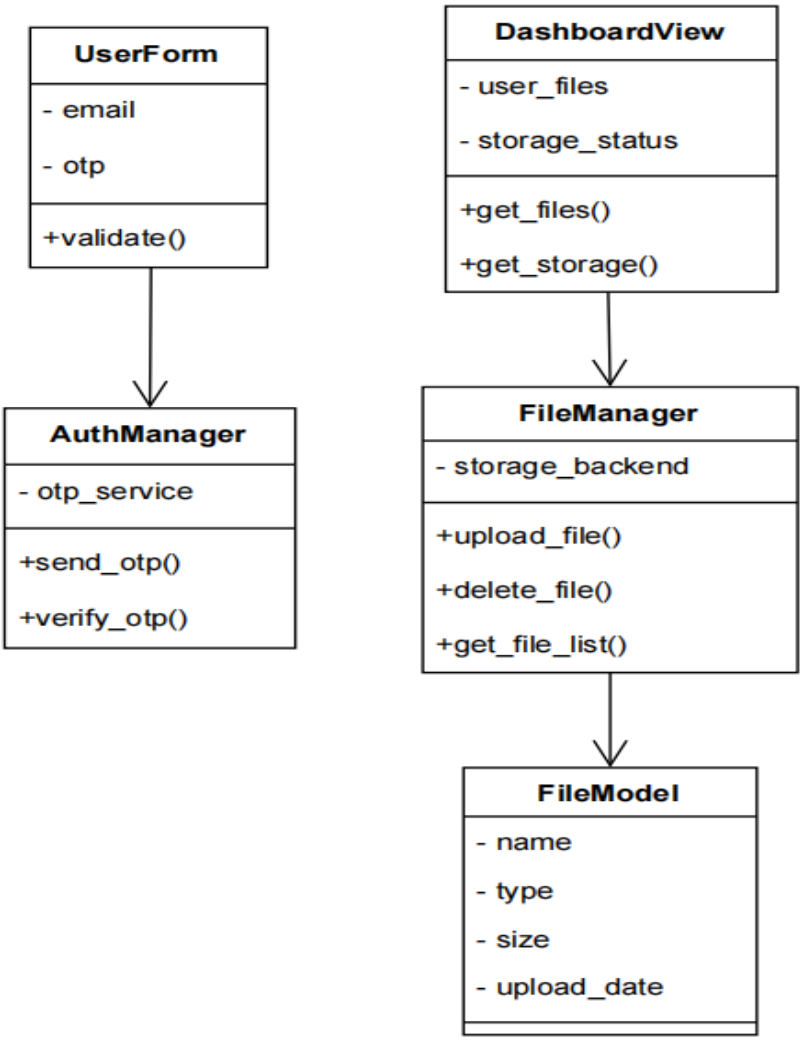
3. **Npm run dev**
Frontend will run at: http://localhost:8081

Project Structure:





Architectural Diagrams :



Features Implemented:

• Secure user login and sign-up using JWT

• File upload, download, and delete functionality

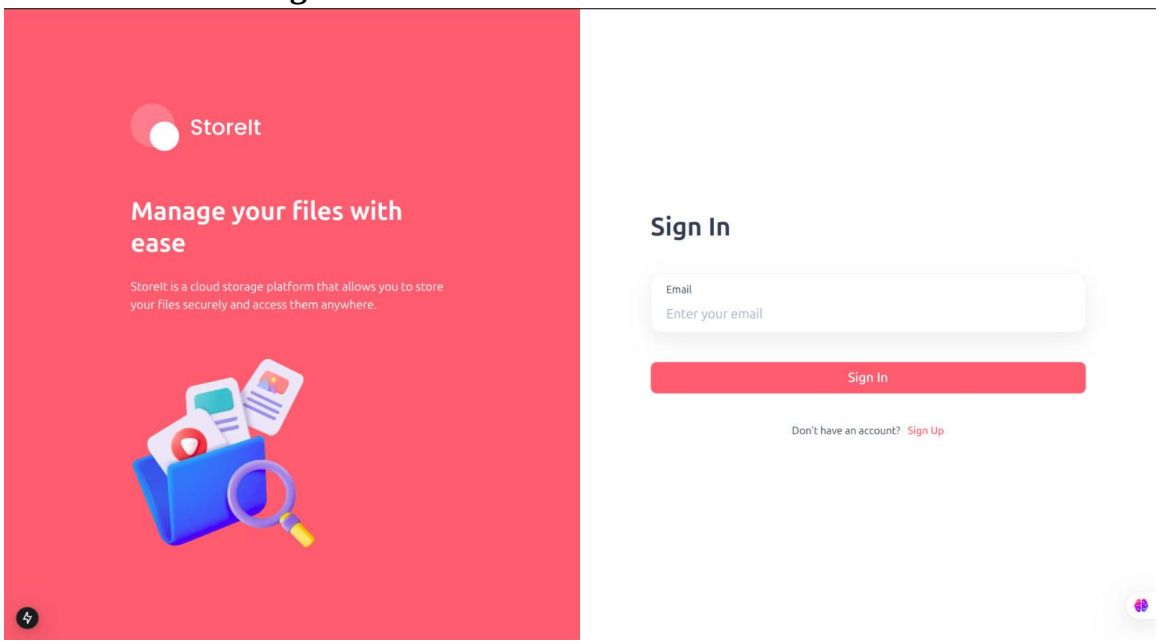
• File sharing via generated URLs

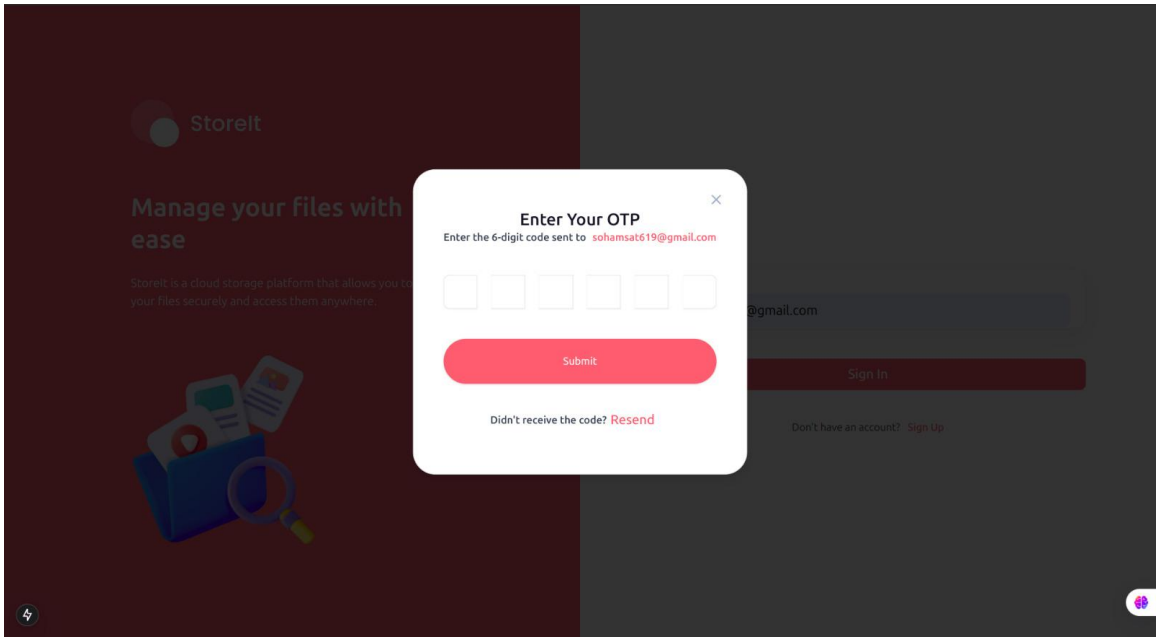
• Dashboard showing file size, name, and upload time

• Responsive and user-friendly UI

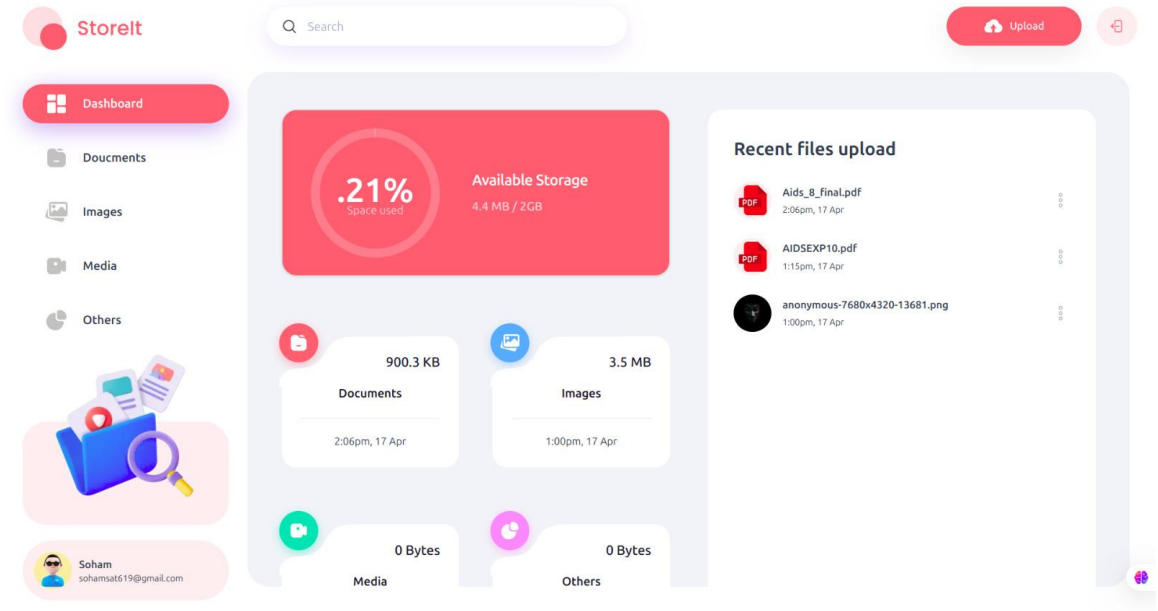
Screenshots of implementation:

1Authentication Page:

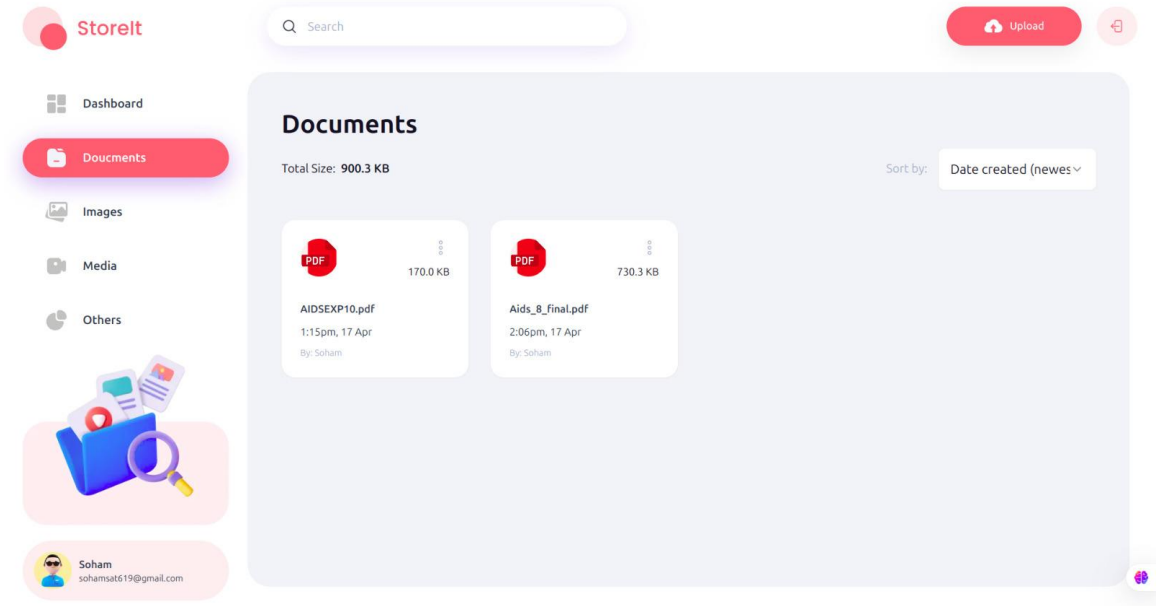


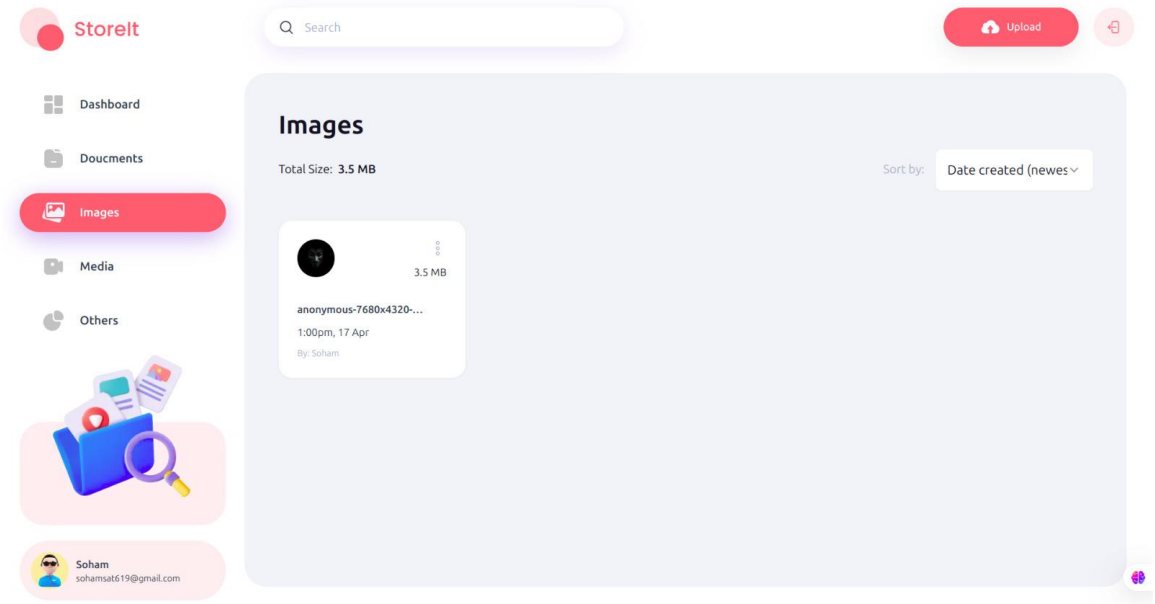


2.Home Screen:



3.Other Pages:





Future Scope:

- Add folder support and drag-drop uploads
- Real-time collaboration and file comments
- Admin dashboard with user analytics
- Cloud-native deployment with CI/CD support

GitHub Link:

https://github.com/Soham2784/Webx-Project

Conclusion:

StoreIt is a practical and robust cloud storage system built for simplicity and security. It bridges modern web development with backend data engineering using Flask and MongoDB. The modular architecture makes it easy to maintain, expand, and deploy. It serves as an ideal project for demonstrating full-stack capabilities with real-world application potential.