

StoreIT

PREREQUISITES - StoreIT

Name: Soham Satpute

Class/Roll No: D15A/51

Title: StoreIt

StoreIt is a cloud-based file storage and sharing application that provides users with a secure, scalable, and user-friendly solution to upload, manage, and share their digital content. The project uses React for the frontend, Flask for the backend, and MongoDB as the primary storage and authentication service. With responsive UI and RESTful APIs, StoreIt ensures seamless interaction between users and the system from any device.

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 (SSD preferred)
- Network: Stable internet connection (required for MongoDB services)

2. Software Requirements:

- Operating System: Windows 10/11, macOS 10.15+, or Ubuntu 20.04+
- Code Editor: Visual Studio Code or any modern IDE
- Version Control: Git 2.25+

Technology Stack

Layer	Technology
Frontend	React (with Vite)
Backend	Flask (Python 3.8+)
Database	MongoDB (Local or Atlas)
Storage	GridFS or Local Storage
Auth	JWT-based Auth
APIs	RESTful Flask APIs

Setup Instructions

Frontend Setup

- Install Node.js (LTS version) from <https://nodejs.org>.
- Check installation: `node -v` and `npm -v`
- Navigate to the React project folder: `cd frontend`
- Install dependencies: `npm install`
- Start development server: `npm run dev`
- Frontend will run at: `http://localhost:5173`

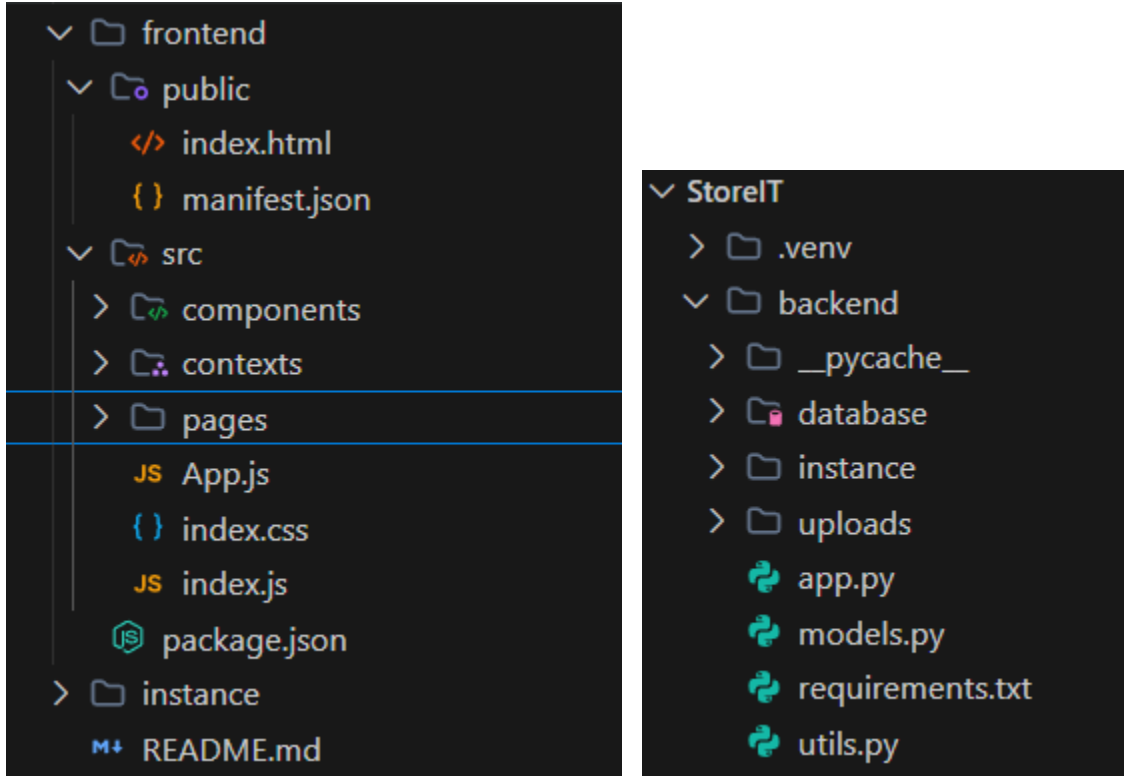
Backend Setup

- Navigate to backend folder: `cd backend`
- Create a virtual environment: `python -m venv venv`
- Activate environment: `venv\Scripts\activate` (Windows) or `source venv/bin/activate` (macOS/Linux)
- Install dependencies: `pip install -r requirements.txt`
- Start the Flask server: `python app.py`
- Backend will run at: `http://localhost:5000`

MongoDB Setup

- Create a MongoDB project at <https://firebase.google.com/>.
- Enable Firestore, Authentication, and Storage in the console.
- Download your `firebaseConfig` object and place it in the React app.
- Ensure rules are set to secure your data appropriately.

Project Structure:



Features Overview

- User Authentication: MongoDB Auth for login and sign-up.
- File Upload: Upload files to MongoDB Storage.
- File Management: View, delete, and organize uploaded files.
- Sharing Links: Generate and share access links.
- Responsive Design: Works smoothly across mobile and desktop platforms.

Conclusion

With the above environment and tools correctly set up, StoreIt will be fully operational, providing cloud storage functionalities with user authentication, file handling, and secure data storage. Ensure that React and Flask communicate properly through REST APIs and that MongoDB services are configured correctly for optimal performance.