WEBX CA
PREREQUISITE

Name of Student	Tejas Dhondibhau Gunjal
Class Roll No	18
D.O.P.	
D.O.S.	
Sign and Grade	

Project Title: - Research Hub – Simplifying Research Access

Project Description: -

Research Hub is a full-stack web application designed to streamline the access, sharing, and discussion of academic resources. Built with **React and TypeScript** on the frontend, **Flask** for the backend, and **MongoDB** as the database, the platform provides a seamless experience for researchers and students.

The goal of **Research Hub** is to build a knowledge-sharing ecosystem where users can not only access research materials easily but also **collaborate, contribute, and discuss** — all in one place. By streamlining access and fostering open academic discussions, Research Hub empowers students, researchers, and professionals to grow together.

SYSTEM REQUIREMENTS:-

Software Requirements

- o **Node.js (v18.16.1 or later):** Required for running the React frontend
- o **npm (v9.5.1 or later):** For managing frontend dependencies
- o **Python (3.11.9 or later):** Required for Flask backend
- o **MongoDB Atlas:** Cloud-based NoSQL database
- o **VS Code:** Recommended IDE for both frontend and backend development

- o **Postman:** For API testing
- o **Git:** For version control and collaboration
- o **Flask Libraries:**
  - Flask
  - Flask-CORS
  - Flask-JWT-Extended (for authentication)
  - pymongo (for MongoDB interaction)
  - python-dotenv (for environment variables)

Hardware Requirements

- **Processor:** Intel Core i5 or higher
- **RAM:** Minimum 8GB (Recommended: 16GB for smooth performance)
- **Storage:** At least 10GB free disk space
- **Operating System:** Windows 10/11, macOS, or Linux

Technology Stack

- **Frontend:** React.js + TypeScript (for strong typing and UI responsiveness)
- **Backend:** Flask (for API and business logic handling)
- **Database:** MongoDB (for flexible document-based data storage)
- **Authentication:** JWT-based login/signup system ensuring secure session management

SETUP INSTRUCTIONS

➤ Frontend Setup (React + TypeScript)

Steps:

1. **Install Node.js and npm**
   o Download from: https://nodejs.org/
   o Verify installation:

   ```
   node -v
   npm -v
   ```

2. **Navigate to frontend directory**:

   cd frontend

3. **Install project dependencies**:

   npm install

4. **Start the development server**:

   npm run dev

5. **Open in browser**:
   Visit http://localhost:5173 to access the frontend.

➤ Backend Setup (Flask - Python)

Steps:

1. **Install Python**
   o Download from: https://www.python.org/downloads/
   o Verify installation:

   ```
   python --version
   ```

2. **Navigate to backend directory**:

   cd backend

3. **Create a virtual environment**:

   python -m venv venv

4. **Activate virtual environment**:

   Windows:

   venv\Scripts\activate

5. **Install Python dependencies**:

   pip install -r requirements.txt

6. **Configure environment variables**:

   o Create a `.env` file in the backend directory
   o Add your MongoDB URI and secret keys:

   env

   MONGO_URI=mongodb+srv://<username>:<password>@cluster.mongodb.net/rese
   archHubDB
   SECRET_KEY=your-secret-key

7. **Start the Flask server**:

   python app.py

8. **Backend will run on**:
   http://localhost:5000

➢ **Database Setup (MongoDB Atlas)**

Steps:

1. **Create an account**:
   Go to https://www.mongodb.com/cloud/atlas

2. **Set up a cluster**:
   o Create a **new project** and **cluster**
   o Choose free-tier if needed

3. **Configure security settings**:
   - o Add **database user** with username/password
   - o Whitelist your **current IP address**

4. **Create a new database**:
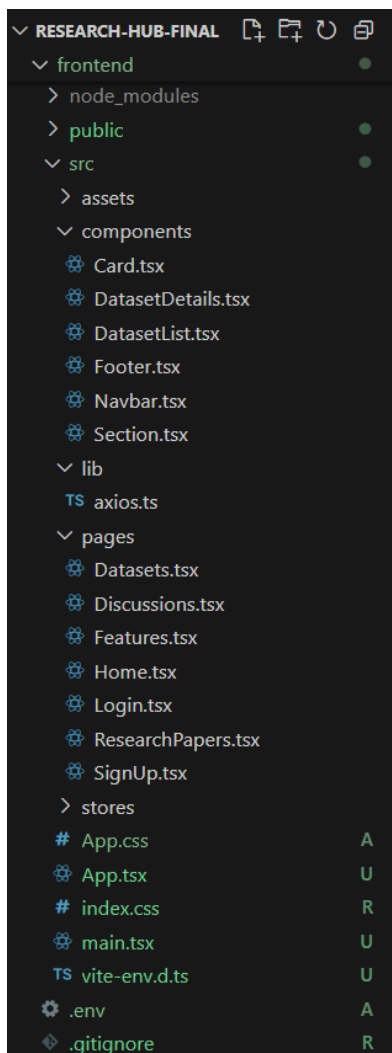   - o Name: researchHub

5. **Get your connection string**:
   mongodb+srv://<username>:<password>@cluster0.mongodb.net/researchHubDB

   Paste this in your backend .env file
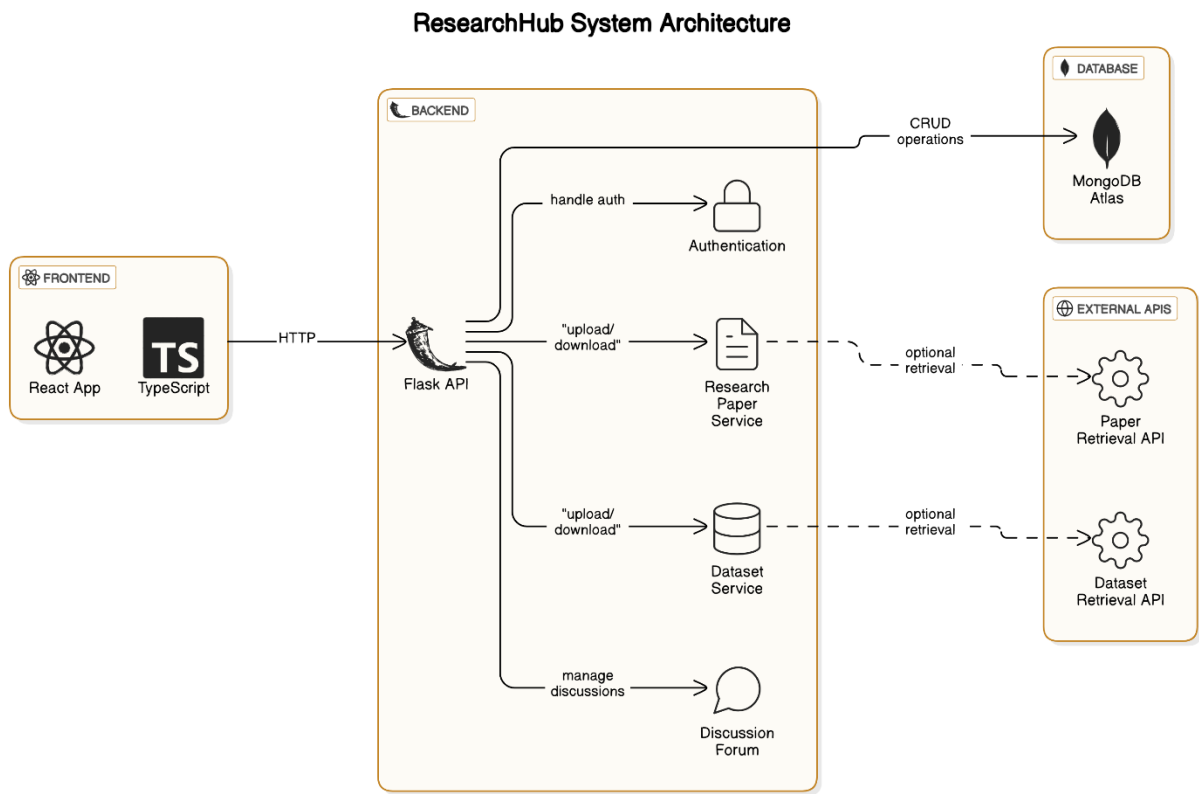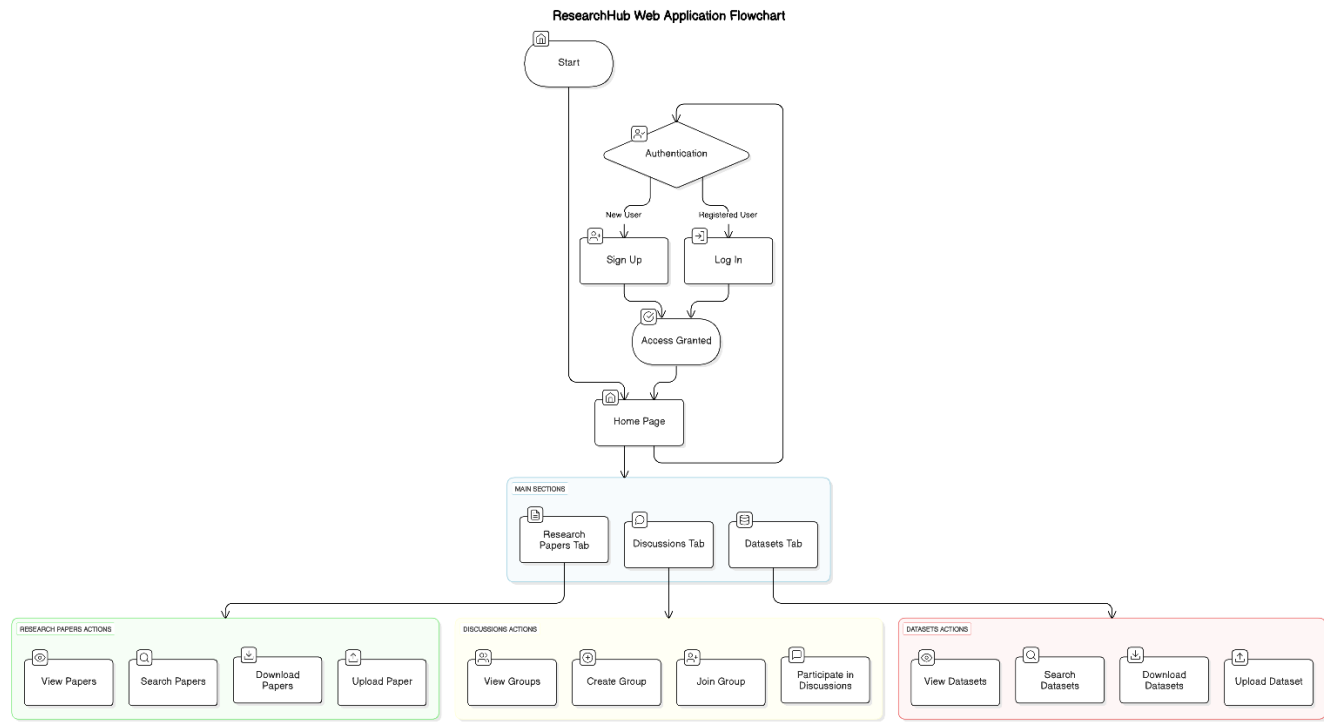
PROJECT DIRECTORY STRUCTURE

➢ **Frontend**

```
∨ RESEARCH-HUB-FINAL
  ∨ frontend                    ●
    > node_modules
    > public                    ●
    ∨ src                       ●
      > assets
      ∨ components
        ⚙ Card.tsx
        ⚙ DatasetDetails.tsx
        ⚙ DatasetList.tsx
        ⚙ Footer.tsx
        ⚙ Navbar.tsx
        ⚙ Section.tsx
      ∨ lib
        TS axios.ts
      ∨ pages
        ⚙ Datasets.tsx
        ⚙ Discussions.tsx
        ⚙ Features.tsx
        ⚙ Home.tsx
        ⚙ Login.tsx
        ⚙ ResearchPapers.tsx
        ⚙ SignUp.tsx
      > stores
      # App.css                 A
      ⚙ App.tsx                 U
      # index.css               R
      ⚙ main.tsx                U
      TS vite-env.d.ts          U
    ⚙ .env                      A
    ◆ .gitignore                R
```

➢ **Backend**

```
∨ RESEARCH-HUB-FINAL
  ∨ backend                     ●
    > __pycache__               ●
    ∨ src                       ●
      > __pycache__
      ∨ controllers             ●
        > __pycache__
        🐍 user_controller.py   U
      ∨ db                      ●
        🐍 __init__.py          U
      ∨ models                  ●
        🐍 user_model.py        U
      ∨ routes                  ●
        > __pycache__
        🐍 user_routes.py       U
      ∨ utils                   ●
        > __pycache__
        🐍 __init__.py          U
        🐍 api_error.py         U
        🐍 api_response.py      U
        🐍 async_handler.py     U
        🐍 auth.py              U
        🐍 cloudinary.py        U
      🐍 app.py                 U
      🐍 config.py              U
      🐍 constants.py           U
      🐍 extensions.py          U
      🐍 main.py                U
    ⚙ .env                      M
    ☰ requirements.txt          U
```

Proposed Architecture

ResearchHub System Architecture

Flowchart

ResearchHub Web Application Flowchart

Conclusion:-

Having clearly defined the necessary tools and environment, this document ensures a smooth foundation for the successful development of the Research Hub web application. By setting up all essential hardware specifications and installing required software like Node.js, Python, Flask, React with TypeScript, and MongoDB Atlas, developers can work in a stable and compatible setup. These prerequisites streamline the development workflow and help avoid common configuration issues, ensuring efficient collaboration and deployment.