

WEBX-CA PREREQUISITES

Name of Student	Mohit Patil
Class Roll No	D15A_36
D.O.P.	20/3/2025
D.O.S.	27/3/2025
Sign and Grade	

Title: Brainly

Introduction:

Brainly is a second brain application that helps users organize their digital life by capturing and managing content from multiple sources. It allows users to save tweets, summarize YouTube videos, create and edit notes, and manage daily checklists—all in one seamless interface. Designed to boost productivity and retention, Brainly transforms scattered information into structured, accessible knowledge. With features like tagging, categorization, and a responsive UI, it serves as a smart companion for creators, learners, and thinkers in the digital era.

System requirements:

1. 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

2. Software Requirements

- **Node.js** (v18.16.1 or later) – Required for running the React frontend
- **npm** (v9.5.1 or later) – For managing frontend dependencies
- **Python** (3.11.9 or later) – Required for Flask backend

- **MongoDB Atlas** – Cloud-based NoSQL database
- **VS Code** – Recommended IDE for both frontend and backend dev
- **Postman** – For API testing

Technology Stack:

1. Frontend

- **React.js** (JavaScript library for building UI)
- **TypeScript** (Strongly typed JavaScript)
- **Tailwind CSS** (Utility-first CSS framework for styling)

2. Backend

- **Flask** (Python-based web framework)
- **Flask-CORS** (To handle cross-origin requests)
- **PyMongo** (For MongoDB connection)
- **PyJWT** (To encode and decode JWT tokens for secure authentication)
- **python-dotenv** (To manage environment variables securely)
- **Flask-Limiter** (To implement rate limiting and prevent abuse of APIs)
- **flask-login** (To manage user sessions and handle login/logout functionality)

3. Database

- **MongoDB Atlas** (Cloud-based NoSQL database for storing product details)

Setup Instructions:

1. Install Node.js and NPM

Download and install Node.js from [Node.js Official Website](https://nodejs.org/en/).

Verify installation:

node -v

2. Install Python and Flask

Download and install Python from the [Python Official Website](#).

Verify installation:

```
python --version
```

```
pip --version
```

Install Flask and required dependencies:

```
pip install flask flask-cors pymongo
```

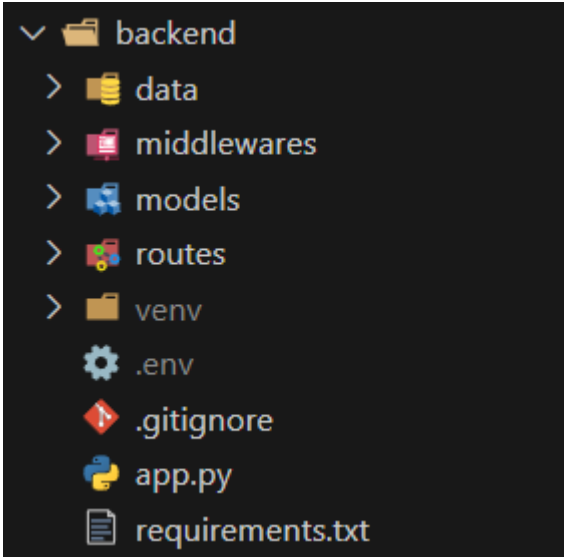
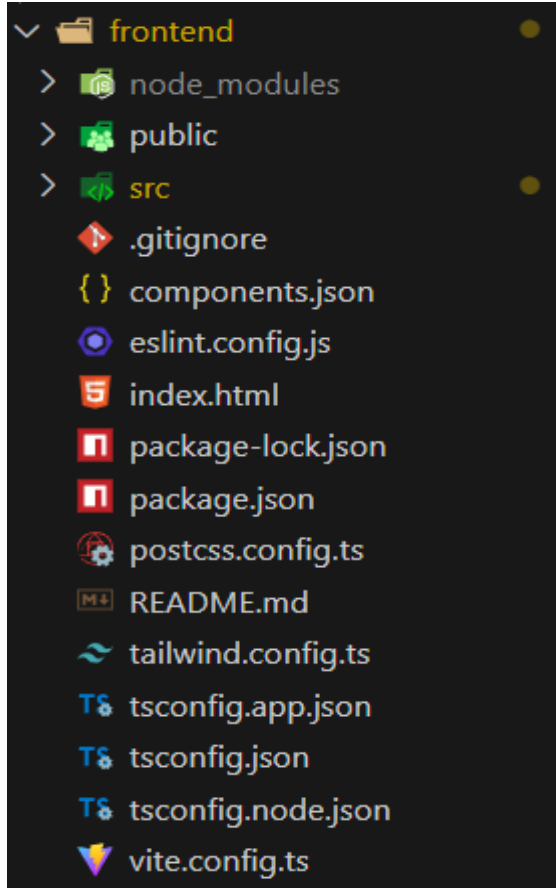
3. Setup MongoDB

Use MongoDB Atlas for cloud storage or install MongoDB locally from [MongoDB Official Website](#).

If using a local setup, start MongoDB using:

```
mongod --dbpath /path/to/data
```

Project Structure:

Backend	Frontend
 <p>A screenshot of a file explorer showing the backend directory structure. The 'backend' folder is expanded, revealing subfolders 'data', 'middlewares', 'models', 'routes', and 'venv'. Files include '.env', '.gitignore', 'app.py', and 'requirements.txt'.</p>	 <p>A screenshot of a file explorer showing the frontend directory structure. The 'frontend' folder is expanded, revealing subfolders 'node_modules', 'public', and 'src'. Files include '.gitignore', 'components.json', 'eslint.config.js', 'index.html', 'package-lock.json', 'package.json', 'postcss.config.ts', 'README.md', 'tailwind.config.ts', 'tsconfig.app.json', 'tsconfig.json', 'tsconfig.node.json', and 'vite.config.ts'.</p>

Conclusion:

This document outlines the essential prerequisites for setting up the UpNest Skill Enhancement Platform, including the required hardware and software. Ensuring all dependencies, tools, and configurations are properly installed will enable smooth development and efficient execution of the project.