Project: Student Job Tracker – AI Tools Usage

Overview: Throughout the development of the Student Job Tracker application, a strong emphasis was placed on thorough documentation and manual coding practices to ensure clarity and maintainability. AI tools were selectively employed to enhance efficiency in specific areas, particularly during the initial setup and certain frontend components.

AI Tools Utilized:

1. **ChatGPT:**

- Assisted in setting up MongoDB configurations and resolving initial setup challenges. DEV Community+2Medium+2Medium+2
- Provided guidance on implementing specific frontend features, such as search functionality and filter components.

2. **GitHub Copilot (VSCode Extension):**

- Offered code suggestions for repetitive tasks, aiding in the rapid development of UI components.
- Facilitated the generation of boilerplate code, which was subsequently reviewed and customized to align with the project's requirements.

Directory Structure Setup: To streamline the initial setup of the React project, a Bash script was utilized to automate the creation of the directory structure and essential files. This approach ensured consistency and saved time during the development process.

Bash Script for React Project Setup:

```
# 1. Make project folder and enter it

mkdir job-tracker

cd job-tracker



# 2. Initialize Git

git init



# 3. Scaffold React app (this will populate
package.json, src/, public/, etc.)
```

```
npx create-react-app .



# 4. Create additional directories under src/

cd src

mkdir api assets components pages hooks utils



# 5. Create starter files

touch api/jobs.js \

      components/Button.jsx
components/JobCard.jsx components/FilterBar.jsx
\

      pages/Home.jsx pages/AddJob.jsx
pages/ViewJobs.jsx pages/EditJob.jsx \

      hooks/useJobs.js \

      utils/formatDate.js



# 6. (Optional) Create or update .gitignore if
needed

cd ..

echo "node_modules/" >> .gitignore

echo "build/"     >> .gitignore



# 7. Make initial commit

git add .

git commit -m "chore: initial scaffold with
React and directory structure"
```

Documentation Approach: The project prioritized comprehensive documentation to ensure that every component and functionality was well-understood and easily maintainable. This approach not only enhanced collaboration but also ensured that the codebase remained accessible to future developers.

Conclusion: While AI tools like ChatGPT and GitHub Copilot were instrumental in streamlining certain aspects of the development process, the project's foundation was built upon meticulous documentation and hands-on coding practices. This balanced approach ensured both efficiency and clarity throughout the development lifecycle.