Project Reflection

This is a short report for a component I forgot to send; The use of AI if any, examples of prompts used and if not then the Human Intelligence thought process.

Use of AI

Yes, I did use AI, particularly Gemini to help augment my research and debugging due to time constraints and lack of knowledge. The an example of the main prompts I used were as such:

"For the presentation deck, I decided to make it an animated power point, for the flow of it, i was thinking to model it based on the rubrics and cover all bases

Task

Mining Code

Database

Report Generation

Telegram bot

Backend code

Application flow chart that connects the front end, back end, database and telegram api

3 alternative use cases/insights

How does that sound? What is normally considered enough or bare minimum for a presentation deck"

Unlike back in school where a student has a tutor, I have to learn it with the basic fundamentals that I learnt from previous, older projects. To be smart about things, I decided to leverage what I had, which was AI chatbots that could help give inputs to design theory, correct my understanding of topics and gather alternatives when a solution fails.

Work Process

Why I submitted so late, and why it is in python is because I had originally attempted this project in Java. I also helped out at my previous internship company, a robotics startup so the schedule got incredibly packed with milestones but I was able to squeeze it out.

My approach and project design philosophy is the same for both Java and Python. The first goal is to understand the project requirements and what I would need to do or learn to accomplish it. Due to the time constraint, I opted to focus on the base functionality, of which I was able to achieve.

The first stage was to learn how to compile and run a java file, from there I learnt the class syntax and how it was different from or similar to other languages I've touched in the past like Javascript, C++ etc. I got the crawling to work via a Jsoup, however I then learn't what AJAX was when the reports only generated 3 posts.

The next step was a work around to AJAX based applications/websites, and the best way would be to use a web engine that could emulate a browser and scroll to load more posts. Selenium Web Driver was the answer, and combined with Web Driver Manager it allowed for easy, automated installation of the needed web driver. The Selenium web driver would get the page and then Jsoup would parse the html contents that we needed from each post.

Then came the obstacle that stopped my Java version of the project. Dependencies and compatibility. So far they had not been an issue as I had tested and recompiled the pom.xml for the Maven project on Visual Studio Code as much as I could. After 2 days of work arounds, reinstalls and debugging the Lombok incompatibility still persisted.

With little time left I made the call to download a new IDE, IntellJ, and learn it from scratch, I also decided to change to python to speed it up as I've worked with python extensively over the last year in school and work projects. Additionally I started to do version control via Github repository.

Following the same track as before with some new augmentations:

- Basic reddit crawling
- Generate a report with crawled posts
- Integrated Selenium Web driver for python + adding sort and post count configuration

Now I had reached where I stopped with Java, next I worked on the flask webservice library to create an endpoint that I could call. So simply navigating to the index route would call the crawler, I integrated the index page/main_page.html later as I wanted to get the functionality testing done first.

From there I could slowly ramp up, integrating the Telegram Chatbot API with the unique bot token that you can generate via @BotFather. That was imported via .env and with gitignore to ensure credential security.

The database system I chose is SQLite as it was easy to use, configure and manage, furthermore it was lightweight and perfectly suited for a project of the scope. Using SQLAlchemy I was able to manage and query the database instance, automatically generating it if needed.

After which I spruced up and finalised the remaining route functions and code, before moving on to the Presentation Deck.

Improvements

It is definitely not a perfect project and I only wished I had more time to properly work on these improvements. The hiring assignment and company description gives me the understanding that the user experience of the client is the main focus of these projects and possibly work at Hepmil as a software engineer.

My project is readable and functional but it lacks simple user feedback features like visible response messages for the client and a loading or indicator bar to show the status of the application. These features I took note of if I had time after pushing out the core functionality and presentation deck.

Besides user ergonomics for the UI, installation and execution too. We want the clients to be able to run and use the application with ease, so consideration has to be taken for those with next to no computer skills. For those who have basic knowledge on coding, running a 'pip install -r requirements' and then 'python application.py' would be easy. But otherwise it would be a learning experience for others. An improvement would be to use a packager like pyinstaller to compile all the requirements and libraries into a single executable installer, that would allow clients to simply run and install the packages with a single click.

To go further, a shortcut exe file that can start the web service application too. Something that can be pinned to the home screen or task bar of a central office computer. For the database and data visualisation, there are numerous possibilities with the data that can be crawled from reddit. Timestamps with vote/comment counts could give indicators of how controversial a post is, a more robust media content helper function could help download more of the media that my code missed, specifically the reddit hosted mp4 and spoiler image files.

Conclusion

In summary, yes. I used AI to help me learn and to debug as if it was a mentor or group mate, I used it as a productivity tool to help push me along as I learnt a totally new project. I was able to achieve the functionality as well as made up a decent presentation deck so I am satisfied. It was a fun upskilling project and I thank you for the time you took to read my hastily put together reflection/report.