REPORT

ON THE DEVELOPMENT OF A HABIT TRACKER APPLICATION FROM A BACKEND PERSPECTIVE

Habit tracking is a powerful tool for building and maintaining positive behaviours. The development of a habit tracker application involves planning out the features and functionality, defining data structures, and utilizing programming concepts to manipulate user information. This report discusses the development process of a habit tracker application from a backend perspective.

The first step in developing a habit tracker application involved planning out the features and functionality. This was achieved by defining the data structures required to create, manipulate, and store user information such as habits, goals, and progress. The object-oriented programming paradigms were used to create habit objects that encapsulate information about the habit. These objects were then added to the database, and functional programming concepts were utilized to manipulate the database, mostly, though SQL queries.

The development process involved the choice of libraries. The Tracker class was made in "tracker.py". "db.py" had the function definitions. "analysis.py" had habit analysis functions, and "tests.py" had two tests for analysing test data. The main program, "main.py," checks for broken habits and interacts with the user. One of the most critical components was the database schema, which defined the structure of the data stored in the application's database. Tables were created for storing active habits, broken habits, and tracking operations. Two scripts were used to define the database schema and four weeks of user data for testing purposes.

The link to the project in GitHub: **https://github.com/Alianger18/ Projects_Portfolio.**

The development of a habit tracker application from a backend perspective was challenging but rewarding. The project provided an opportunity to learn about SQL queries and how to create a large project instead of small ones. The project also provided an opportunity to learn some basic Git concepts like cloning, committing, and pushing.

The aspect of the project, I'm most proud of, for me was developing the algorithm responsible for verifying whether a habit was still active or broken. it needed a high degree of focus, logic, and knowledge with date and time data types in Python. Although there was a desire to create a fancy interface and a way for users to log in, time constraints prevented these features from being implemented. The focus was on the essential parts of the project.

Overall, this project was a valuable learning experience and provided insight into the development of habit tracker applications from a backend perspective. However, I am excited to work on applications that rely on artificial intelligence, particularly Chat-GPT-based applications. There is currently a growing trend of relying on AI, and by working on such projects in the future, I hope to gain expertise in dealing with machine learning algorithms. As a data scientist, consistent work with machine learning algorithms is required to provide insights and data-driven solutions to potential problems. Therefore, working on AI projects will allow me to develop skills and gain experience that will be valuable in this field.