COSC-1104-01 - Scripting

Student ID: 100938484

Name: Mihir Limbad

Problem Statement:

Task management is a common challenge for many people, making it essential to have an effective way to organize tasks. This project involves building a Python-based Task Manager application to help users manage their tasks efficiently. The application will allow users to perform basic task management operations such as adding, viewing, editing, marking tasks as completed, and removing them. Additional features include searching tasks, filtering by status, and highlighting overdue tasks. The tasks and their details will be stored persistently using an SQLite database.

Anticipated Difficulty:

This project is of moderate difficulty. It requires knowledge of:

1. Handling databases using the sqlite3 library for CRUD operations.
2. Input validation (e.g., date formats) to ensure accurate data entry.
3. Managing task statuses (e.g., pending, completed) and deadlines.
4. Implementing a user-friendly, menu-based interface to guide interactions.
5. Using helper functions for modular and maintainable code.

The challenge lies in effectively handling user input, designing a robust database schema, and providing a clear display of tasks, including overdue highlights.

Justification:

Creating this Task Manager application is beneficial for both users and developers:

1. User Benefits:
    - Organization: Users can manage their daily tasks efficiently by categorizing them and setting due dates.
    - Progress Tracking: Highlighting overdue tasks and separating completed tasks fosters accountability and focus.
    - Persistence: SQLite ensures task data remains intact across sessions.
2. Developer Benefits:
    - Database Knowledge: Gain hands-on experience with SQLite for designing and querying databases.
    - Modular Code: Practice organizing Python code into functions for scalability and maintenance.
    - Real-World Applicability: Develop a practical tool that aligns with common productivity needs.

SQLite3 Library

SQLite is chosen for this project due to its simplicity, reliability, and built-in Python integration. Key advantages include:

1. Lightweight: SQLite is a disk-based database that doesn't require a separate server process.
2. Serverless: Makes it ideal for small-scale applications without external dependencies.
3. Cross-Platform: Runs seamlessly on various operating systems.
4. Ease of Use: The Python sqlite3 module provides a user-friendly interface to perform database operations using SQL queries.

Using SQLite, the application achieves persistence without complexity, ensuring users can maintain their task data effortlessly.

Reflection:

1. Did you choose a problem that felt just right in terms of difficulty, or was it too easy or too hard?

The selected problem felt just the appropriate level of difficulty. By including basic ideas like database management, CRUD operations, and user input processing, it offered a well-balanced challenge. It was a good option for practicing Python abilities because it wasn't too difficult or too easy.

2. What was the hardest part of solving this problem?

The most difficult aspect of tackling this issue was making sure the program reacted appropriately to different user selections and efficiently handled user input. Careful thought had to go into implementing a strong menu system that correctly deduced and carried out the user's objectives. Another significant challenge was maintaining the SQLite database, which included creating accurate SQL queries.

3. What did you use to learn about the libraries or new features you used?

For this project, I referred to the SQLite documentation to understand and implement database operations in Python. The documentation provided insights into creating tables, executing queries, and managing data within the SQLite database. Additionally, I learned about datetime formatting for handling task deadlines, relying on Python's built-in datetime module documentation.

Problem 2: Health and Fitness Tracker

Problem Statement:

My second problem is on health and fitness tracker. In our increasingly busy lives, maintaining a healthy lifestyle can be challenging. To address this, I propose developing a comprehensive Python program that acts as a Health and Fitness Tracker. Features like calorie intake tracking, exercise routine logging, and customized motivational messages will be available with this health and fitness tracker. Exercise type, length, and intensity are just a few of the characteristics that users can enter about their workouts. After that, the application will use the entered data to compute and log the number of calories burned. To get a comprehensive picture of their energy balance, customers can also enter their daily caloric intake.

Anticipated Difficulty:

Because of a few important considerations, creating a health and fitness tracker is a somewhat difficult undertaking. Creating an interface that is easy to use and intuitively collects exercise regimens, calorie consumption, and motivating messages is a huge problem. Complexity is increased by the complexities of data validation, which calls for strong procedures to reliably manage a range of user inputs. It can be difficult to convert recorded data into useful graphs when using the **matplotlib library** for data visualization, but it also makes for interesting representations of users' fitness progress. Putting in place a system for customized motivational messages is even more challenging since it necessitates knowing user preferences and creating content that is specific to each person's fitness objectives.

Justification:

The proposed Health and Fitness Tracker scenario has the potential to significantly help consumers in their quest for a healthier way of living. With its all-encompassing approach to health tracking, the tool gives users the ability to monitor their calorie consumption and activity habits, giving them a complete picture of their overall health. Personalized motivational message integration creates an environment that is consistently supportive and encouraging, which is essential for keeping consumers motivated to stick with their fitness objectives. Users may visually follow their progress over time through incorporating data visualization with the **matplotlib toolkit**, which turns abstract fitness measures into concrete and inspirational insights.

Mataplot Library:

Its adaptability, which offers a wide range of plotting functions to accommodate various fitness data types—from pie charts to line plots that show the distribution of calories consumed—is one of its main advantages. Additionally, a great deal of customisation is possible with matplotlib, allowing developers to adjust visualization appearance according on user preferences.

Another key features of Mataplot are:

- It allows for extensive customization of plot elements, such as colors, labels and annotations.
- It supports interactive plotting, allowing users to explore and interact with the visualized data.

- The library is known for its capability to generate publication-quality graphics.

Reflection:

1. Did you choose a problem that felt just right in terms of difficulty, or was it too easy or too hard?

In terms of difficulty, the Health and Fitness Tracker scenario offered a balance. It was doable given the limitations of Python programming, even if it was difficult enough to necessitate careful consideration of user interface design, data validation, and API interaction. The scenario's complexity fit the learning objectives perfectly, enabling the application of fundamental programming ideas to the resolution of a practical issue.

2. What was the hardest part of solving this problem?

Creating an easy-to-use user interface that smoothly integrated tracking calories consumed, exercise routine logging, and customized motivational messages was the most difficult part of tackling this problem. User interaction design principles had to be carefully considered in order to ensure a user-friendly experience while processing a variety of inputs. Furthermore, there were difficulties in comprehending and successfully implementing the functionalities of the external health APIs that were integrated to improve the computation of calories.

3. What's the most important thing you've learned from doing this assignment?

The assignment's most valuable takeaway was how to use data visualization to solve problems in the real world. Using matplotlib to create visual representations of fitness progress improved my comprehension of the library and made clear how important visual communication is. This encounter demonstrated the effectiveness of data visualization in improving user accessibility and motivation for complicated fitness data.