Phase 3: Abstract

Cem Oguz, Matriculation: 32008124, 29th April 2023

https://github.com/CemOguz/HabitTracker

Objective

The main goal of the project is to develop an app that allows the user to keep track of their habits on daily and weekly basis. For this purpose, I used a database table and displayed the data frame on the startup dashboard of the application. App is able to read, add, delete, keep track and analyze habits.

Development Process

App is separated into two folders. One contains the main.py, other contains Data Frame. The purpose of doing so is mostly for the sake of developing process, to ease it, making the app neat, and not putting everything into the same basket. Each time a transaction is completed, app goes back to where it starts: displaying the Data Frame, which actually displays the habit table on HabitTracker.db. Completing a transaction also commits these updated data to the database.

One of the main classes of the application is SQL class under SqliteHelper.py. This is where most of the pondering was done, making the app as requested. Especially, adding a habit streak function has many parameters and checks which were created by use of various conditions & nested loops. From the view of the developer, this is the most valuable and time-spent part of the entire project. This function returns different results in accordance with different conditions met or not.

Gains

Though it's a beginner level project, OOP and its capabilities were used as wide as possible. As a prospective data scientist, I benefited from ready libraries and modules of python 3.

Datetime module was powerful enough to make use of any potential problems a developer would face, making them available in hand. Especially timedelta object is used repetitively for any transaction about a habit.

sqlite3 provided me the chance to combine the knowledge gained from database course and incorporate them within another development area. Tabulate module is used to make the table look readable and unified so as to present a user-friendly look. Pandas is already one of the most popular library used for data manipulation and analysis, so is a must for the project, and used in data frame.

Using SQLite for the database engine provided convenience, speed and reliability. Since SQL is more professional, and could potentially be used more often during my future career, I preferred it over other technologies to store, handle and query the data. By using SQLite Studio and data frame in the app itself, I was able to double check the latest status of a habit on real time like a live dashboard. Seeing the results right after running the main.py gave quick insight to the developer.

Tests

Until the 2nd phase of the app, it had 5 predefined habits, but for testing purposes two new ones were added: test1 and test2. For testing, I used pytest as being a popular testing framework. It provided a simple and intuitive syntax for writing tests and supports a wide range of testing capabilities.

It is easy to use and allowed to write better tests in less time. I also included two test files which were not present until this phase, called **test_main_2.py** and **func1_test.py**.

First one, **main_2.py** tests if an attempt to check off a **daily habit (test1)** can be checked off without breaking the habit, and its streak count can be increased in compliance with nested loops alghoritms under addHabitStreak() function. In the last part of the test, assertion double checks if two values of streak_days counts differ by 1. Similarly, the second one, **func1_test** applies the same order and algorithm and runs the procedure for the **weekly habit, test2**. Since it's a weekly one, the assertion happens over a 7 days difference instead of one.

Analysis

As the name suggests, this part is for collecting, analyzing, and interpreting data to gain insights. Six different types of questions are presented, and they query the data with the functions which run SQL commands in SqliteHelper.py file. Since they return the actual numbers collected from habit table, they work as they should.

The pitfalls

Every step had the potential to cause minor mistakes, such as syntax errors, an unnecessary indentation, or a valuable which had to be a global, but defined as a local one. Thus, all sub steps challenged me in their own circles. However, nested loops under habit streaks and their logical order were slightly a bigger problem than others. As a whole, all steps contributed to the complexity of entire project.

During testing, I figured out that one of the main advantages of pytest is its flexibility and extensibility. It can be used to test any type of Python code, from small scripts to large applications, and can be easily extended with plugins to support additional testing capabilities or integrate with other tools and systems.

As stated earlier, most of the coding job was done in the addHabitStreak() function where there are nested loops. The controls under this function was first to see the habit's type, daily or weekly, then, check when was the last time it was checked off, then check it was on time so that it was counted as a part of streak, then check the record of that particular habit.

Else, app considers the habit as broken and timer is reset to zero. This way, user is encouraged to be punctual and complying to their personal goals.

Conclusion

The app works well. Its Data Frame is displayed over the app, and could be seen over a DB browser. In my personal tests, I did not come up with any errors, but still I would be pleased to be know if there is any. As a whole, project can do the following:

- The user can add and delete habits.
- The user can "check-off" habits. Application saves all the data to habit table and retrieves them whenever the app runs again. However, breaking the habit will result in resetting the streak days.
- Record column is not reset until a higher score is achieved. It's not linked to a habit streak was broken or reset. It's another type of data that user should see and be encouraged to try more.
- The asked Analytics are shown in the main table and come sorted so that the same periodicities are shown together.