Habit Tracker Application

Report

Tatjana Kiriakov

Approach/First steps

In order to start a meaningful project productively, it is important to understand the task well and to know exactly what the goals of the application are. For this I have dealt intensively with the task and marked the most important points.

The first phase of the project helped a lot in creating a structure and referring to it when writing the code. This phase made the technical area clearer, and it also automatically became clear how many and which files the code should have. Other than that, I needed to figure out which modules the project needed. But this point was finished quickly because by knowing what functionalities the project should have I could research what modules could be useful for that.

The first step was to create the Python files and give them a meaningful name. After that I dealt with the individual classes one after the other. For each class I first defined the functions that were necessary according to the task. For example, functions to output all habits or to analyze which streak was the longest belonged in the "analyze.py" file. Functions that allow the user to define habits and determine a time span belong in the "habits.py" file and so on.

As soon as the individual files were completed, I started building the "main.py" file. There I was able to use the previously created functions. So I could fully focus on main.py without distracting myself from building other functions. With "main.py" it was very important for me that the output is as understandable and structured as possible. Because that is what the user ultimately sees and interacts with. I had the idea of labeling the menus with numbers so that the user can navigate the menus by typing one of the numbers. Thus, he does not have to write out a whole word or even sentence. Also, don't forget to provide the option to switch back from one menu to the other and exit the program from the main menu. I labeled this functionality with an "x" or "exit".

It is obvious that it can happen to every user that he makes an incorrect entry. Error messages and error handling in general are very relevant for this. Otherwise the program would crash and the user would not be able to continue the app. Therefore, if an incorrect entry is made, an error message appears asking the user to type the entry again - this time correctly.

Not to forget is the documentation that I mostly tried to do while coding. Some docstring/comments were however added afterwards if I saw that they were needed for a better understanding.

The README.md file was also created after the project was finished so that I could insert some screenshots of the “user interface” and explain how the application should be used. For that the code needed to be working already without showing errors.

What went well & issues

Creating the basic structure was relatively easy for me, since the task was clear to me from the start. I was also able to quickly create the menus with the selection options. Error handling was also not a problem for me, since I saw the app from the user's perspective and analyzed where mistakes could be made.

Of course, there were also aspects that took more time and also produced more errors.

For one thing, I had to invest quite a bit of time researching databases and how to connect them to a Python project. The commands were often used in the "analyze.py" file to get a specific element from the database. I also had to research these commands and find out what exactly I had to write in order to get the output I wanted.

In addition, several errors occurred when using the datetime module. In order to use functionalities such as overdue or reset, I had to try around for a while until the code ran error-free.

Another unexpected point that took time was renaming the files. I first started the files with a capital letter and then changed them - e.g. "Analyze.py" to "analyze.py". Unfortunately, I didn't take into account that these changes have a major impact on the whole code, since it is case-sensitive. So I had to go through each file and change the naming everywhere there as well.

In general, however, it can be said that the project was manageable with the help of online resources and a good planning in the beginning. I had no huge issues that caused the whole code to had to be changed and with the help of Error management I could avoid having issues because of wrong inputs.

Github link to the project:

https://github.com/tatjanakiriakov/habitsTracker