Project: Python Programming – Time Tracking Application

A project for Data Maverick Premium members

Written by David Venturi (davidanalyst.com)

Welcome to your first project in the bootcamp: Python Programming. In this project, you'll apply the skills you acquired in Chapters 1-10 of Dr. Chuck's Python for Everybody. You have two options for completing this project: the standard project below or the create-your-own-adventure project. Both count towards your progress in the bootcamp the same. Choose the create-your-own-adventure project if you want to tailor your bootcamp projects to a specific industry and/or want to to test your skills without any starter code. Choose the standard project otherwise.

Project Specification: Time Tracking Application

Design and implement a time tracking application using Python. The application allows users to track the time spent on various tasks and provides functionalities to start a task, stop a task, display task history, and display the total time for all completed tasks.

Project Requirements:

1. Implement a `start_task()` function that prompts the user to enter the task name and starts tracking the time for that task.
2. Implement a `stop_task()` function that prompts the user to enter the task name and stops tracking the time for that task. The function should calculate the duration of the task, print it out, and then store it.
3. Implement a `display_task_history()` function that displays the task history, including the task name and the time spent for each task. The function should differentiate between currently running tasks and completed tasks.
4. Implement a `display_total_completed_time()` function that calculates and displays the total time spent on all completed tasks.
5. Create a user-friendly main menu that allows users to interact with the application. The main menu should provide options to start a task, stop a task, display task history, display the total completed time, and exit the program.

Project Rubric

1. **Functionality (60 points)**
    - `start_task()` function tracks the time for a new task accurately. (10 points)
    - `stop_task()` function calculates and stores the duration of the completed task accurately. (10 points)
    - `display_task_history()` function displays the task history, differentiating between running tasks and completed tasks. (15 points)
    - `display_total_completed_time()` function calculates and displays the total time for all completed tasks accurately. (15 points)
    - The main menu provides options to interact with the application and functions as expected. (10 points)
2. **User Interface and Usability (20 points)**
    - The application has a clear and user-friendly interface with appropriate messages and prompts. (10 points)
    - The main menu provides clear instructions and options for the user. (5 points)
    - The application handles user input gracefully and provides meaningful error messages when necessary. (5 points)
3. **Code Quality (10 points)**
    - The code is well-structured, organized, and readable with appropriate naming conventions. (5 points)
    - Proper indentation and consistent coding style are maintained. (3 points)
    - Comments are used to explain complex logic or enhance code understanding. (2 points)
4. **Documentation (10 points)**
    - The project includes a clear and concise project specification explaining the requirements. (5 points)
    - The code includes comments or docstrings to explain the purpose and functionality of each function. (5 points)

Total: 100 points