Task Instructions

Welcome! We would like to invite you to participate in our study and thus make a valuable contribution to our research. Our research goal is to study developers' programming behavior with a Flask web application.

Instructions:

- The task requires Python (at least version 3.10.)
- You can use an IDE of your choice for solving the task. Before starting with the task, please disable all IDE Plugins that use AI (plugins such as GitHub Copilot, Tabnine, …)
- **NOTE: PLEASE ONLY USE the paid version of Gemini** (https://gemini.google.com) for solving the task. The access data for a Google account with a paid Google One subscription (AI Premium Version) will be provided. Within this account you will have access to Gemini Advanced.
  - Follow the link: https://accounts.google.com/
  - Login with the provided Google Account:
    - **Username: dcs.study.05**
    - **Password: VIOq759F6Yc2**
  - Follow the link https://gemini.google.com to use Gemini
- Please do not enter login data such as passwords or personal data into the chat in Gemini. All chats in the provided account will be deleted after the participation.
- Please **do not use any other AI tools except the paid version of Gemini**.
- In order to solve the task, you are allowed to use any kind of source available on the Internet that may be helpful or has valuable information, except Websites that include AI such as ChatGPT or Microsoft Copilot/Bing Chat. Please use only the premium version of Gemini instead.
- You may import any needed external libraries.
- How to start the application: *python -m flask run*

Link to Task:

- https://classroom.github.com/a/dc4CCrD_

Task:

1. Please follow the link above and connect your GitHub Account to GitHub Classroom. Select your Pseudonym from the list.
2. Clone the code to your IDE
3. Install the requirements with *pip install -r requirements.txt*
4. Complete the TODOs in the code
5. Push the completed code to GitHub
6. Please fill in the following survey after working on the task: https://study.hcs.ruhr-uni-bochum.de/jfe/form/SV_0Mqq9MxygspL7YG

Implementation Hints:

The Python code sets up a Flask web application with user authentication and basic CRUD functionalities for managing user websites. It uses Flask's login extension for managing user sessions and SQLite for database operations. The User class represents user objects with their credentials. There are routes for registering, logging in, logging out, viewing the dashboard and managing websites. Functionalities that should be implemented are marked with TODO. Please pay attention to the security aspects during implementation. In the future, it is planned that users will be able to share their profiles containing their websites publicly, therefore making them visible to others.