Health and Lifestyle Web App -

—---

General instructions

1. Implement the solution in Python.
2. Every configuration and code written should be pushed on GitHub/Bitbucket (Private Repo).
3. You are not permitted to share the doc with anyone, even with your colleagues.
4. Code Commenting has to be added.

Requirement -

1. Implement using the code with an Object Oriented Design
2. Confine to the SOLID principle
3. Confine to programming best practices and standard Python Project Layout

Acceptance Criterion -

1. Create **Unit Test** cases for the application with **80% code coverage**
2. Implement **logging** for the application for debug level.
3. Push the code to a repo on **Git** in separate files.
4. Make sure to have **Visual Representation** drawn on draw.io/miro for the call/data flows, flow chart and functional architecture for both the Problem statement and the solution.
5. Create a **readme** file on git and define the code flow and how to run it.

—---

Consider the following features in the flask web application-

1. The App must contain these sections -

a. Free section for all the users with features -
    i. Free health tips on food and lifestyle
    ii. Exercises to do for daily routines
b. Section for authenticated users with features -
    i. User need to register himself on the app for accessing all the features in this section
    ii. Connect you with a dietician for personalized diet tips based on your particular needs which you will fill through a form
    iii. Users will start getting a personalized diet for all the 3 meals everyday.

Application feature details -

- Enter the app homepage with 4 buttons options -
    a. Free health tips on food
    b. Free health tips on lifestyle
    c. Benefits of different exercises
    d. Sign In/sign up Option
- Integrate these buttons with the routes(APIs) which will redirect to another page on the application and open the page with the respective feature.
- Create a Signin/SignUp page for the app with options for user to sign up for accessing the special features in application by entering the username and creating a password and sign in to the application only with the same username and password for authentication and authorization using jwt tokens.
- After signing in the application, redirect to a page with special feature sections with following button options -
    a. Connect with a dietitian which will redirect to a form
    b. After successfully submitting the form, you will be redirected to the same page and a personal diet box will start showing the personalized diet tips for all the 3 meals based on your particular needs.
- Enable a button for dietitians use only and disable for rest of the users, for clicking on this the dietitian can post the diet for a particular user and update it anytime.
- Deploy the application using the docker
- Create a swagger API documentation

Implementation Requirements -

- For application interface elements use Jinja template

- For saving and fetching the user login information and posting the diets of the users and updating them everyday, use SQLite database and perform the CRUD options in the application
- User JWT tokens and sessions for authentication and authorization.
- For directing the user to different pages of the application create routes(like REST APIs)
- Using hosted API from the internet for showing the free health tips on food and lifestyle and benefits of exercises.
- Implementation with OOPS
- Best practices in python, microservices principle
- Considering the 12 factors pattern