HealthMate: Your Personal Health Companion

ESLIN KIRAN ILANGOVAN
Saint peters university

Professor - Dr. Jean Chu - DS-542 - Python in Data Science

Overview of HealthMate:

- HealthMate is a user-friendly web application designed to assist individuals in maintaining a healthy lifestyle.
- It offers personalized diet and workout recommendations to help users achieve their health and fitness goals.
- With its intuitive interface and smart technology, HealthMate serves as a reliable companion for managing overall well-being.

Application:

- Addressing the challenges of modern health management: In today's fast-paced world, individuals often struggle to maintain healthy habits due to lack of time, resources, or personalized guidance.
- Providing personalized solutions: HealthMate aims to bridge this gap by leveraging advanced technology to deliver tailored diet and workout plans based on individual preferences, needs, and goals.
- Empowering users: By offering personalized recommendations and actionable insights, HealthMate empowers users to take control of their health and make informed choices that support long-term wellness.

Objectives:

- Gather user information on diet preferences, fitness levels, and health goals, ensuring a user-friendly interface for input.
- Utilize OpenAI APIs to analyze user data and provide personalized meal suggestions, as well as tailored workout plans based on user preferences and fitness levels.
- Offer a diverse range of meal options to cater to different dietary needs and preferences, along with a variety of workout routines to accommodate various fitness goals and levels of expertise.

Tools and Technologies:

- Utilize Python programming language for backend development, handling data processing, logic, and algorithms efficiently. Leverage Python's versatility and robust libraries for effective backend functionality.
- Employ Flask, a lightweight web framework, to develop the user interface and manage HTTP requests/responses effectively. Benefit from Flask's simplicity and flexibility for building dynamic web applications.
- Use HTML (HyperText Markup Language) to structure and format the content of web pages, ensuring clarity and coherence. Create visually appealing and responsive user interfaces to enhance the overall user experience.
- Integrate OpenAI APIs to access advanced natural language processing capabilities, enabling the generation of personalized recommendations. Harness the power of AI to analyze user data and provide intelligent suggestions for meals and workouts, enhancing user engagement and satisfaction.

Key Features:

•Users can create profiles containing their diet preferences, fitness levels, and health information. Profiles enable HealthMate to offer personalized recommendations tailored to individual needs and goals.

•HealthMate leverages OpenAI's advanced capabilities to generate personalized and nutritious meal suggestions. Users receive meal recommendations based on their input data, ensuring relevance and suitability to their dietary requirements.

•HealthMate provides users with customized workout plans designed to match their preferences and fitness levels. By considering user inputs and goals, the application offers tailored exercise routines to help users achieve optimal results.

Methodology and User Interaction:

•Users provide basic information about their diet, fitness levels, and health goals through the HealthMate interface. Input data serves as the foundation for generating personalized recommendations, ensuring relevance and accuracy.

•HealthMate utilizes OpenAI's technology to analyze user inputs and generate personalized meal suggestions. OpenAI's advanced algorithms process user data to deliver intelligent recommendations that align with individual preferences and nutritional needs.

•Based on user profiles and input data, HealthMate recommends workout plans tailored to users' preferences and fitness levels. Users receive curated exercise routines that match their goals, making it easier to follow a fitness regimen suited to their individual needs.

Python Flask code

```python
from flask import Flask, render_template_string, request
import openai

app = Flask(__name__)

# Set your OpenAI API key here
OPENAI_API_KEY = "[redacted]"

# Initialize the OpenAI client
openai.api_key = OPENAI_API_KEY

# Define route for the index page
@app.route('/', methods=['GET', 'POST'])
def index():
    if request.method == 'POST':
        # Get the form data submitted by the user
        age = request.form['age']
        weight = request.form['weight']
        veg_or_nonveg = request.form['veg_or_nonveg']
        region = request.form['region']
        food_type = request.form['food_type']
        gender = request.form['gender']
        height = request.form['height']
        generic_diseases = request.form.get('generic_diseases', '')
        allergies = request.form.get('allergies', '')

        # Prepare the user data to be sent to the OpenAI API
        user_data = {
            "model": "gpt-3.5-turbo-instruct",
            "prompt": f"My age is {age}, my weight is {weight} kg. I am {veg_or_nonv
            "max_tokens": 150
        }

        # Make a request to the OpenAI API
        api_response = openai.Completion.create(**user_data)

        # Process the API response as needed
        return api_response.choices[0].text.strip()
```

HTML Code

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>HealthMate - Diet and Workout Recommendations</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f0f0f0;
            margin: 0;
            padding: 0;
        }

        .container {
            max-width: 600px;
            margin: 50px auto;
            background-color: #fff;
            padding: 20px;
            border-radius: 10px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
        }

        h1 {
            text-align: center;
            color: #333;
        }

        label {
            font-weight: bold;
            color: #666;
        }

        input[type="text"],
        select {
            width: 100%;
            padding: 10px;
            margin-bottom: 15px;
            border: 1px solid #ccc;
            border-radius: 5px;
            box-sizing: border-box;
        }

        input[type="submit"] {
            width: 100%;
            background-color: #4CAF50;
            color: white;
            padding: 14px 20px;
            margin: 8px 0;
            border: none;
            border-radius: 4px;
            cursor: pointer;
        }

        input[type="submit"]:hover {
            background-color: #45a049;
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>HealthMate - Diet and Workout Recommendations</h1>
        <form action="/" method="post">
            <label for="age">Age:</label>
            <input type="text" id="age" name="age" required>

            <label for="weight">Weight:</label>
            <input type="text" id="weight" name="weight" required>

            <label for="veg_or_nonveg">Vegetarian/Non-vegetarian:</label>
            <select id="veg_or_nonveg" name="veg_or_nonveg" required>
                <option value="vegetarian">Vegetarian</option>
                <option value="non-vegetarian">Non-vegetarian</option>
            </select>

            <label for="region">Region:</label>
            <input type="text" id="region" name="region" required>

            <label for="food_type">Food Type:</label>
            <input type="text" id="food_type" name="food_type" required>

            <label for="gender">Gender:</label>
            <select id="gender" name="gender" required>
                <option value="male">Male</option>
                <option value="female">Female</option>
            </select>

            <label for="height">Height:</label>
            <input type="text" id="height" name="height" required>

            <label for="generic_diseases">Generic Diseases (if any):</label>
            <input type="text" id="generic_diseases" name="generic_diseases">

            <label for="allergies">Allergies (if any):</label>
            <input type="text" id="allergies" name="allergies">

            <input type="submit" value="Submit">
        </form>
    </div>
</body>
</html>
""")

if __name__ == '__main__':
    # Run the app on all network interfaces and port 5000
    app.run(debug=True, host='0.0.0.0', port=5000)
```

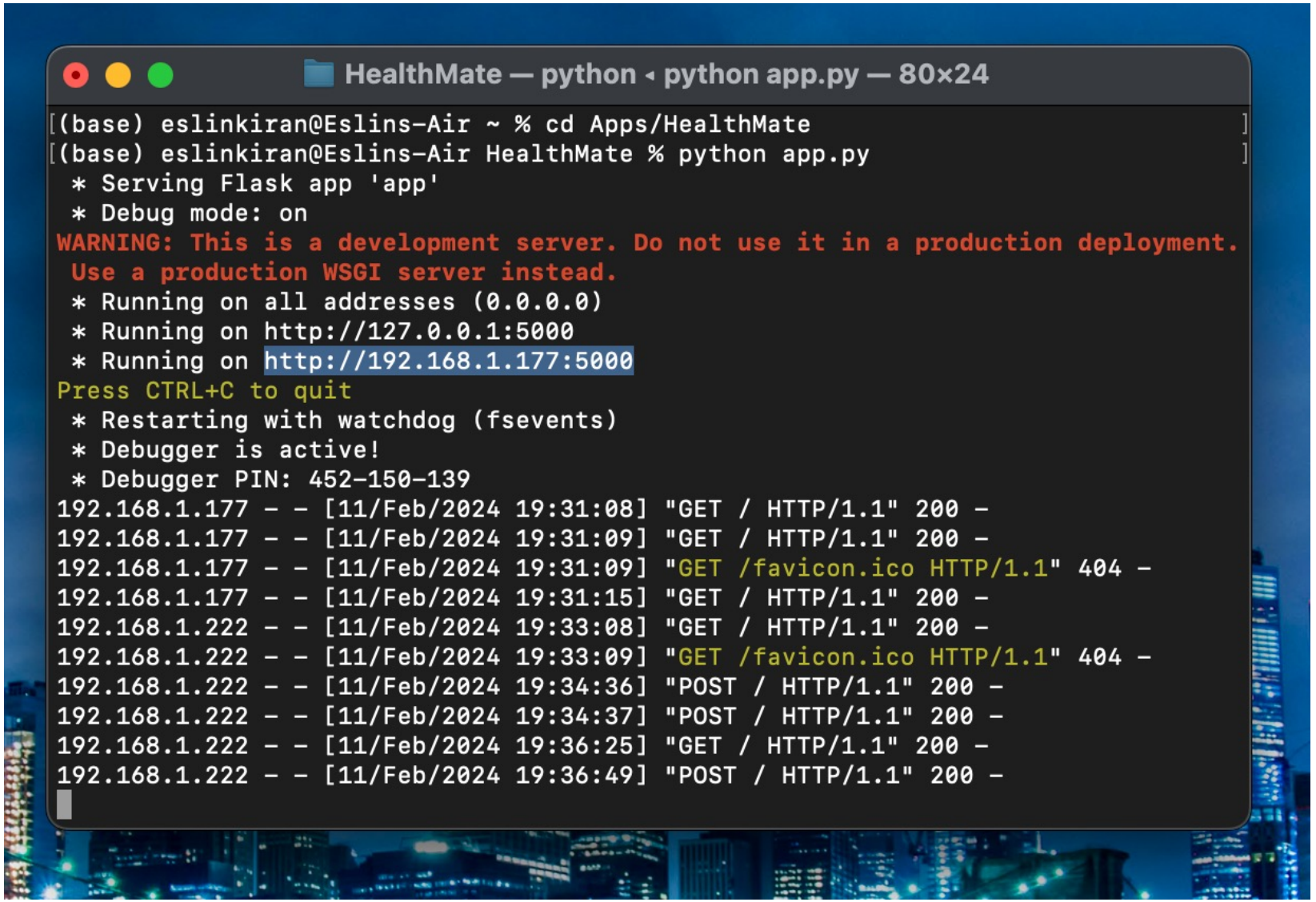
Conclusion:

- HealthMate aims to be a user-friendly health companion, providing personalized diet and workout recommendations to help users maintain a healthy lifestyle effectively.

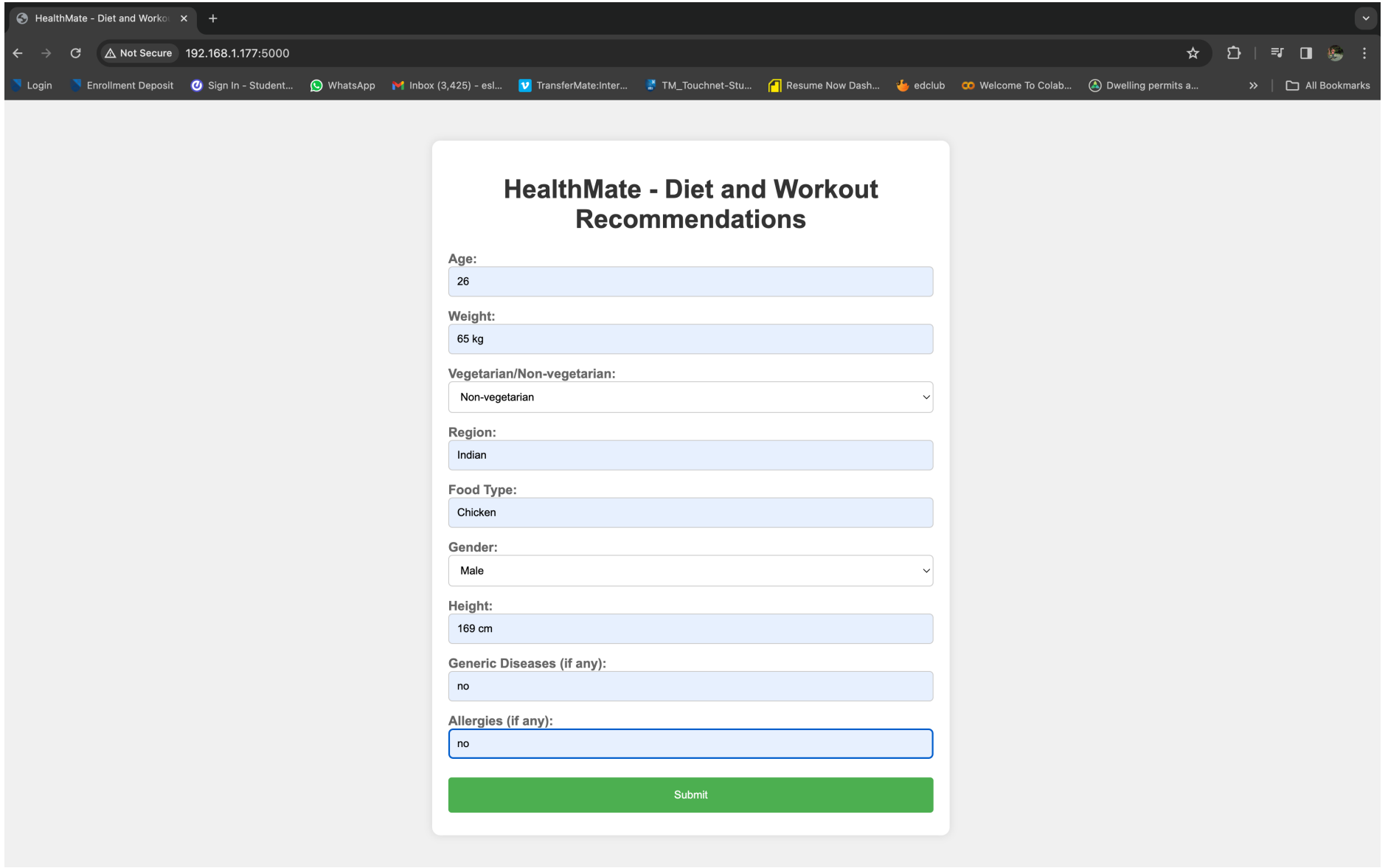
- With its intuitive interface and smart recommendations, HealthMate strives to create a supportive environment that encourages users to prioritize their well-being.

- As more than just a health management tool, HealthMate is committed to empowering users on their journey to better health, serving as a valuable ally in achieving wellness goals.

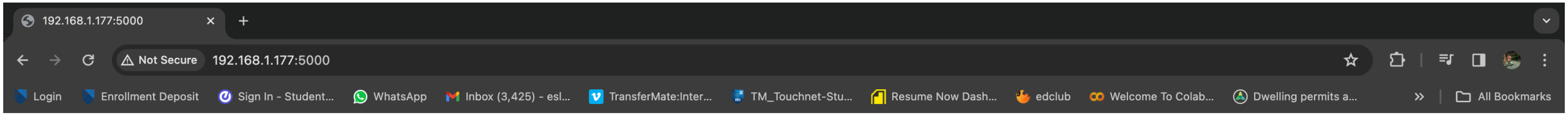
Running flask application



Web page generation using http://192.168.1.177:5000



Result output for diet and workout plan



Diet Plan: Breakfast: - 2 boiled eggs (or 1 egg omelette) - 2 slices of whole wheat toast with peanut butter or avocado - 1 cup of green tea Mid-Morning Snack: - 1 small fruit (apple, banana, or any seasonal fruit) - A handful of mixed nuts (almonds, walnuts, and cashews) Lunch: - Grilled or roasted chicken breast (100-150 g) - 1 cup of brown rice or Quinoa - 1 cup of steamed vegetables (broccoli, carrots, and bell peppers) - 1 small bowl of lentil soup or dal Afternoon Snack: - 1 cup of Greek