PROJECT REPORT ON

Workout Assistant

INT 428

Submitted By:

Name:	**Registration no.:**
Kripal Nath Tiwari	12303851
Anshuman	12315317

Under The Guidance Of:

Mr. Dipen Saini

7 April, 2025

LOVELY PROFESSIONAL UNIVERSITY

Introduction

In today's fast-paced world, maintaining a healthy lifestyle is often challenged by time constraints and the overwhelming amount of time information available. This project addresses these challenges by developing "Time Mangment," an intelligent chatbot designed to provide personalized assistance guidance and support. Leveraging the power of artificial intelligence, specifically advanced models like DeepSeek through the Fireworks AI platform, this chatbot aims to act as a virtual personal trainer, offering tailored workout routines, nutritional advice, and motivational support directly through a conversational interface.

This report will detail the development process of the "Time Mangment" , including the system architecture, implementation details, integration of AI models, and the user interface design. It will also discuss the challenges encountered during development and the solutions implemented to overcome them. Furthermore, it will evaluate the chatbot's performance and explore potential future enhancements to further improve its functionality and user experience. Ultimately, this project aims to contribute to the growing field of AI-assisted personal fitness and provide a valuable tool for individuals seeking to achieve a best lifestyle.

Objectives

The primary objectives of the Time Management Tool are:

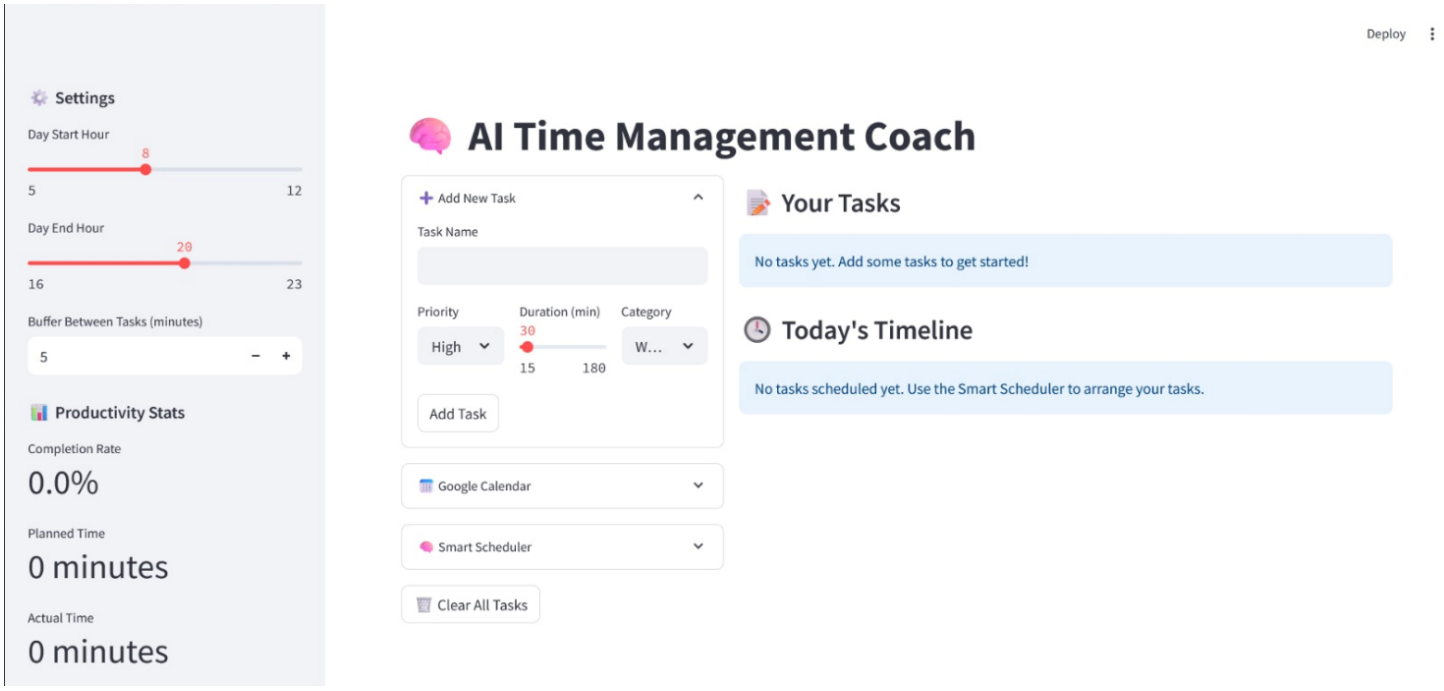
- **Create a Personalized Time Assistance :**Develop a conversational AI chatbot that provides tailored time schedule advice based on individual user needs and preferences.

- **Integrate Advanced AI Models:** Implement and utilize advanced AI models, such as DeepSeek, to generate accurate and relevant fitness recommendations.

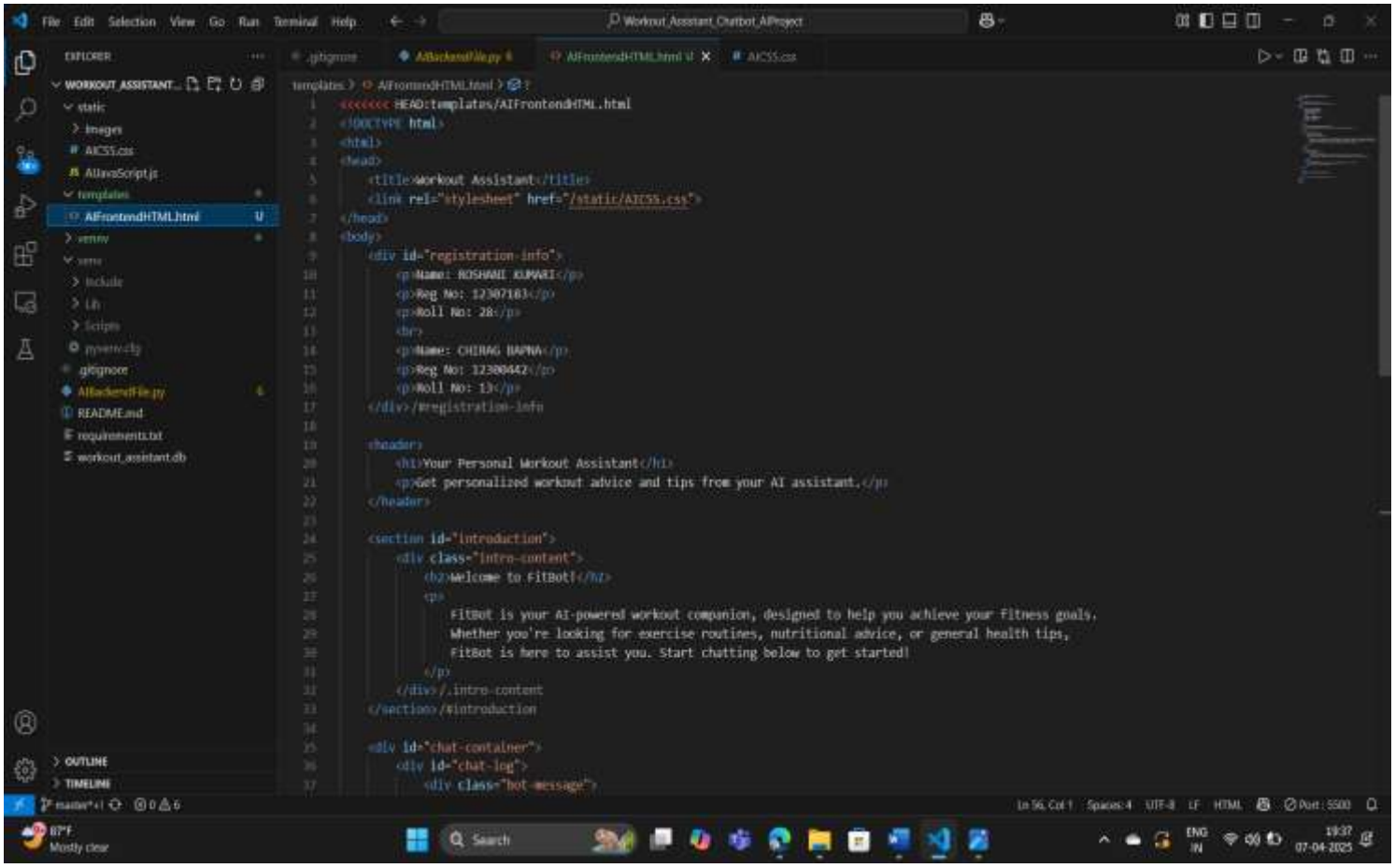
- **Offer Comprehensive Work Support:** Design the chatbot to cover a wide range of Time Management, including all the work schedule, meal planning, and motivational support.

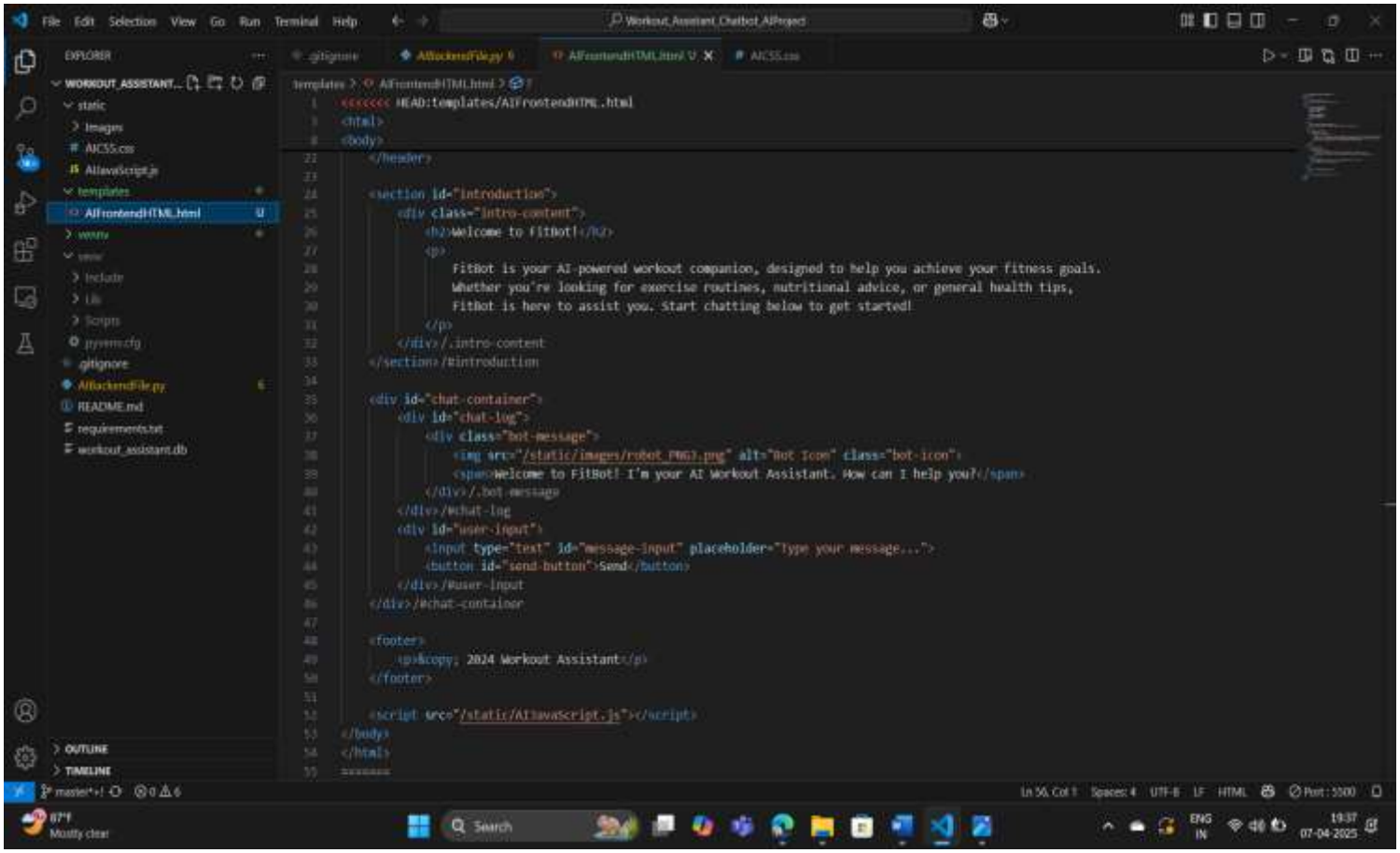
- **Ensure User Accessibility and Engagement:** Build an intuitive and user-friendly interface that makes time guidance easily accessible and engaging.

Screenshot of Tool



Screenshots of code




<<<<<<< HEAD:templates/AIfrontendHTML.html
<html>
<body>
</header>
<section id="introduction">

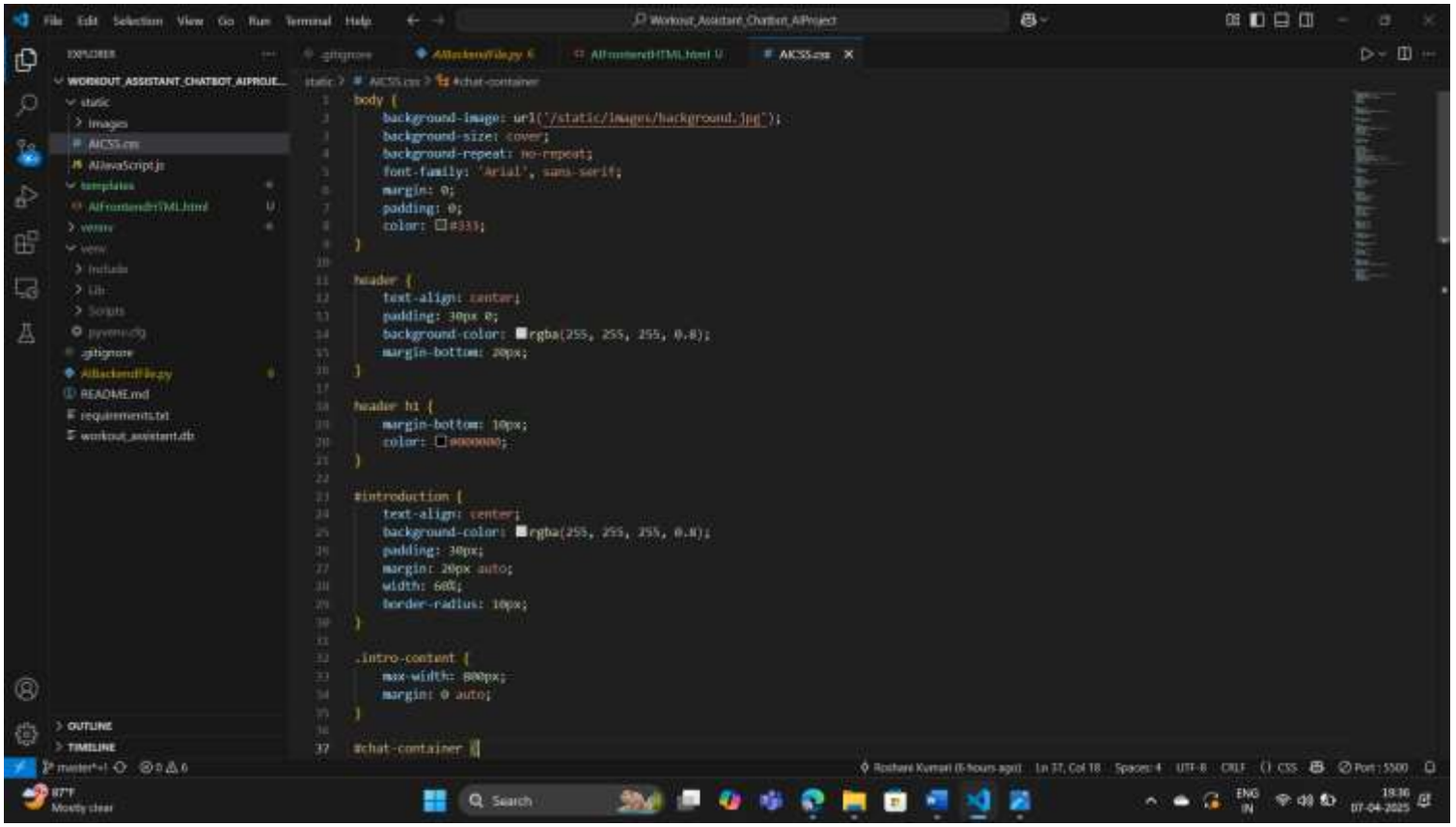
<h2>Welcome to FitBot!</h2>

FitBot is your AI-powered workout companion, designed to help you achieve your fitness goals.
Whether you're looking for exercise routines, nutritional advice, or general health tips,
FitBot is here to assist you. Start chatting below to get started!

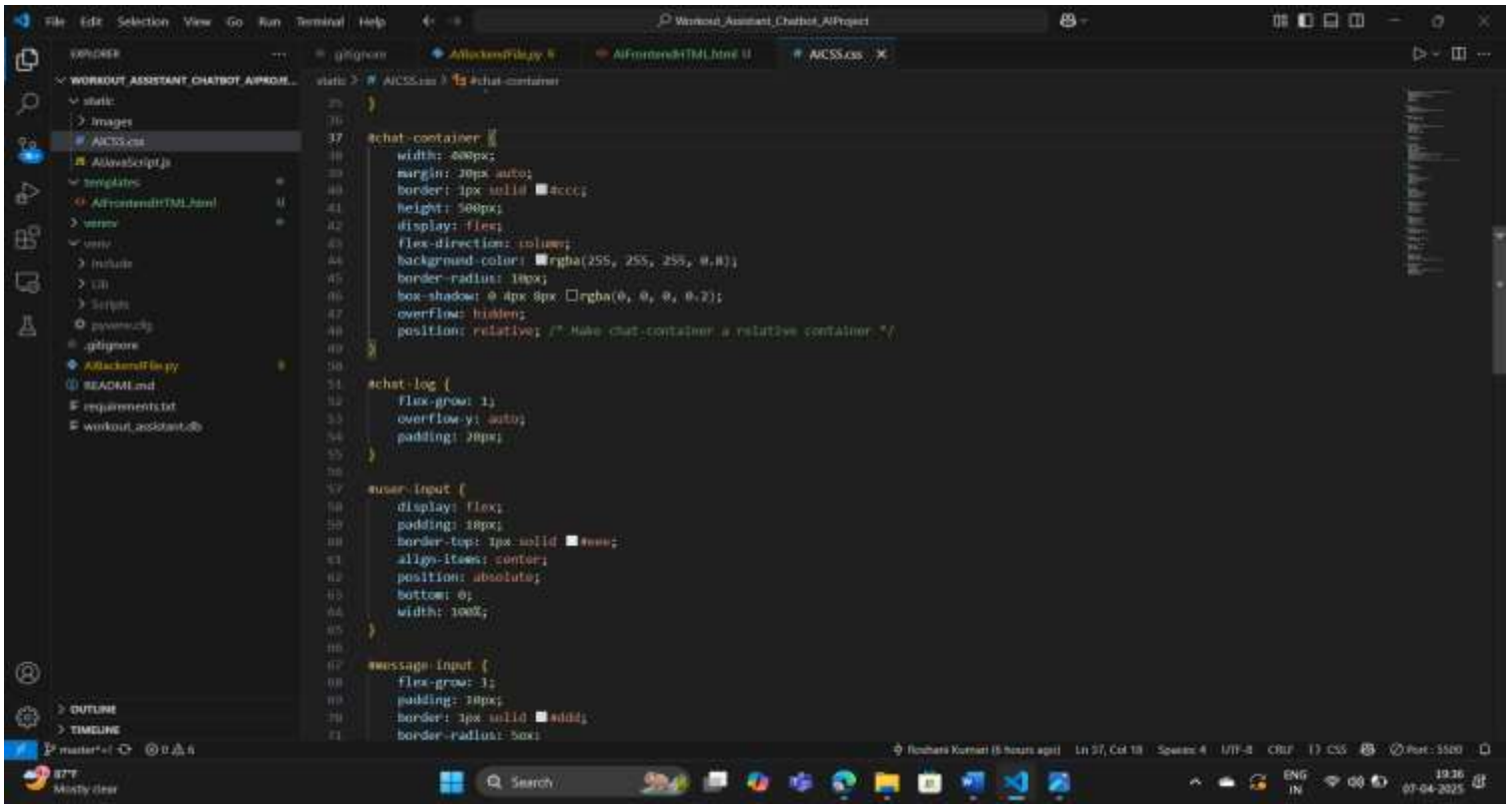
Welcome to FitBot! I'm your AI Workout Assistant. How can I help you?

<input type="text" id="message-input" placeholder="Type your message...">
<button id="send-button">Send</button>

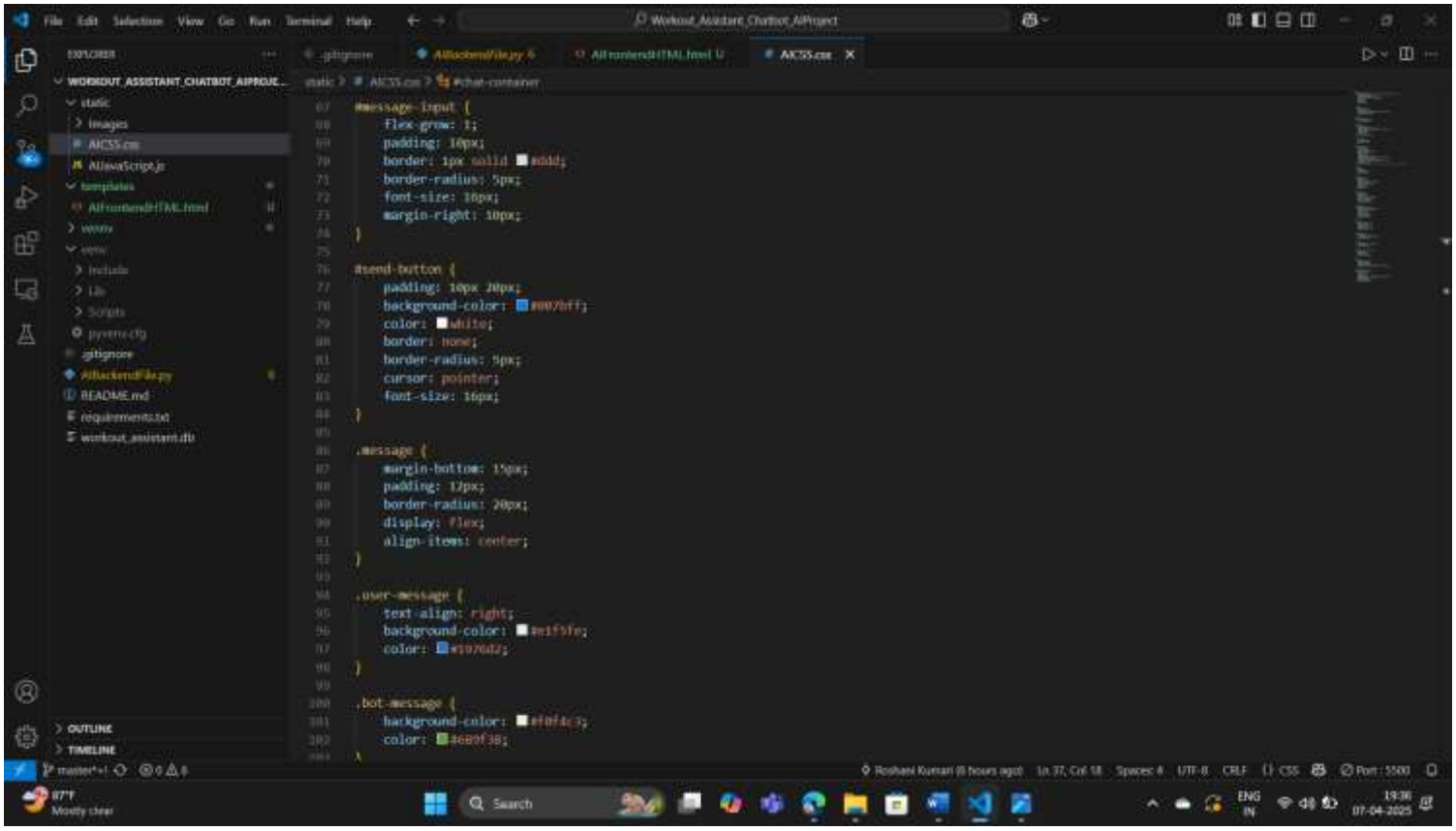
<footer>
© 2024 Workout Assistant
</footer>
<script src="/static/AIjavaScript.js"></script>
</body>
</html>
=======

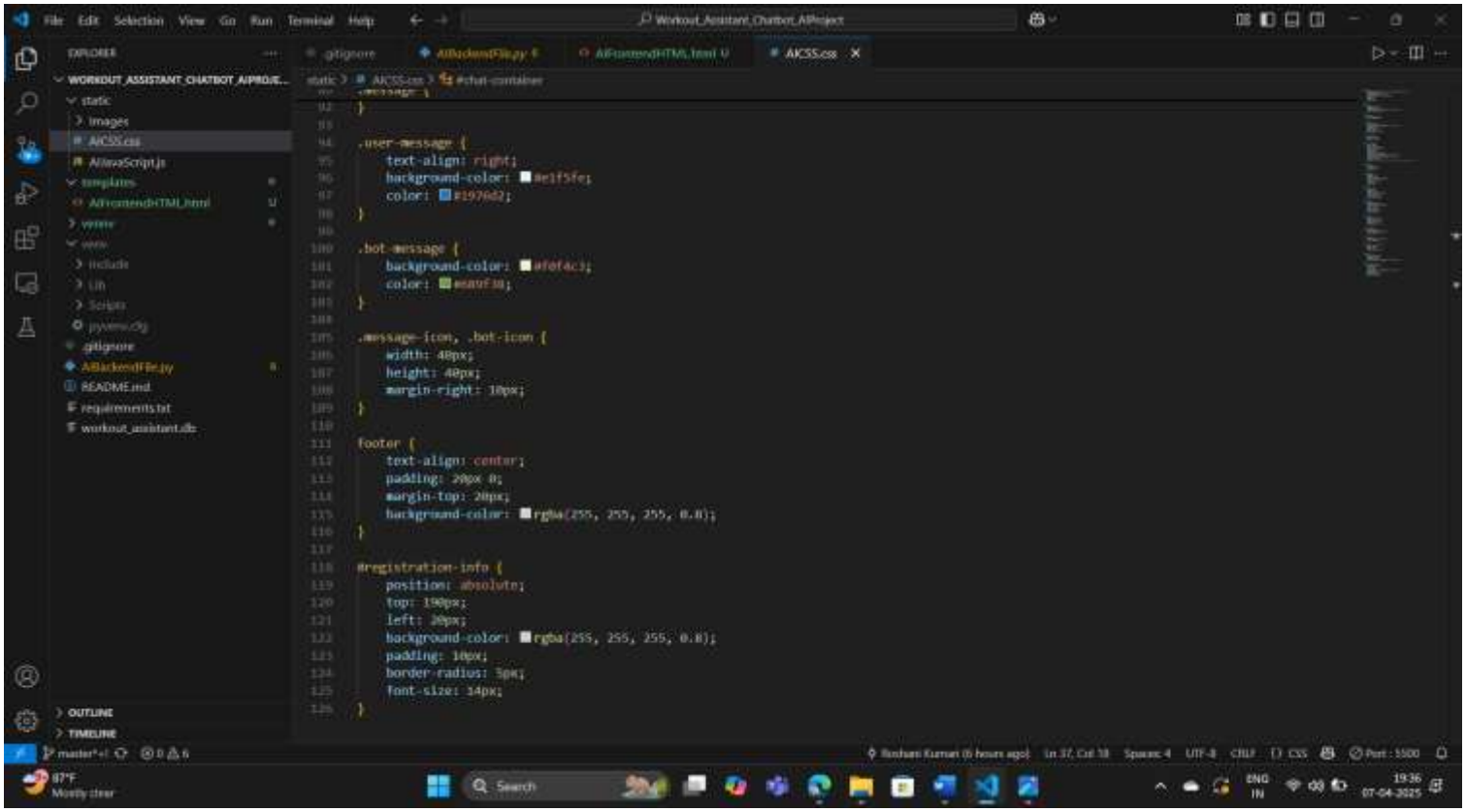


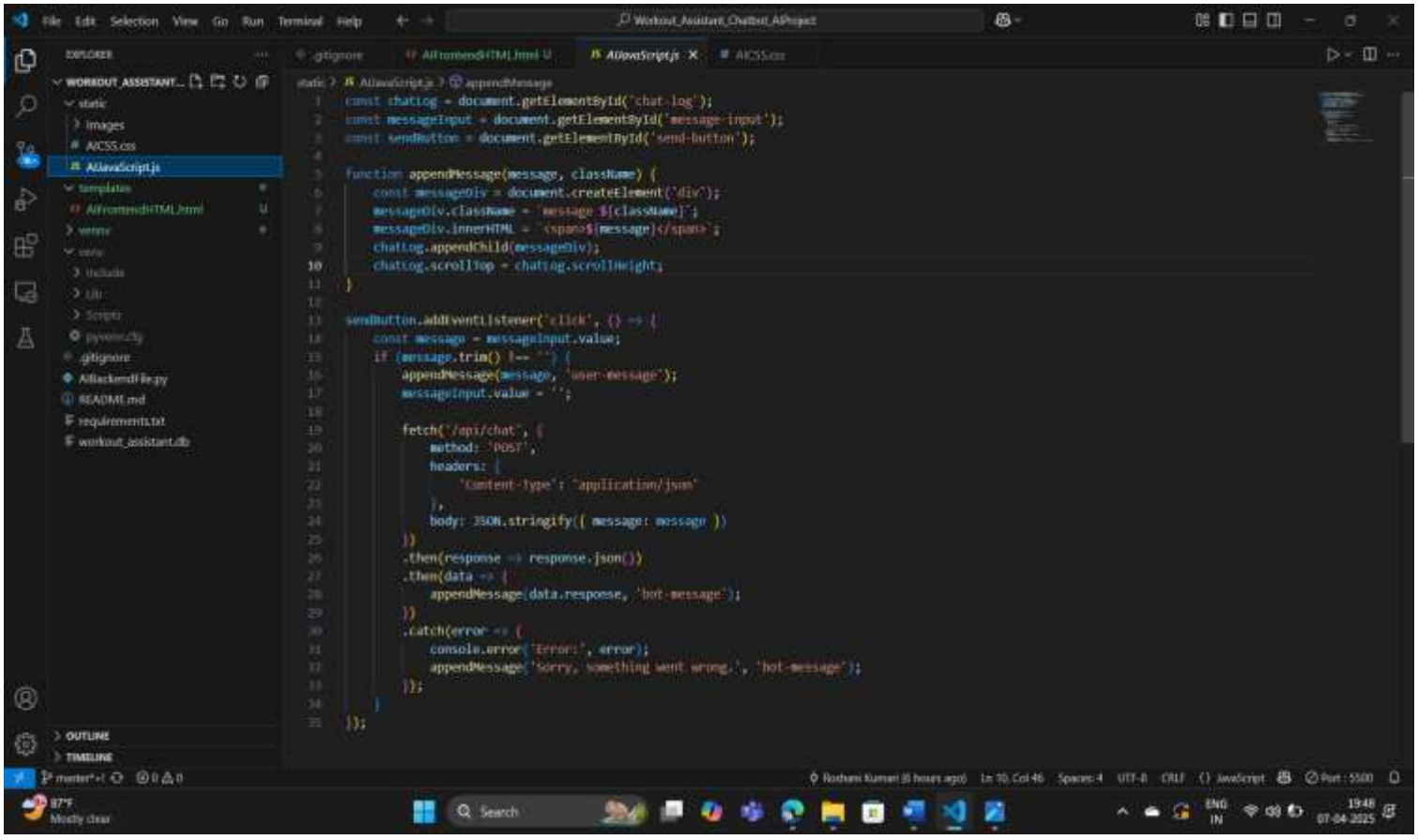
```css
body {
    background-image: url('/static/images/background.jpg');
    background-size: cover;
    background-repeat: no-repeat;
    font-family: 'Arial', sans-serif;
    margin: 0;
    padding: 0;
    color: #333;
}

header {
    text-align: center;
    padding: 30px 0;
    background-color: rgba(255, 255, 255, 0.8);
    margin-bottom: 20px;
}

header h1 {
    margin-bottom: 10px;
    color: #000000;
}

#introduction {
    text-align: center;
    background-color: rgba(255, 255, 255, 0.8);
    padding: 30px;
    margin: 20px auto;
    width: 60%;
    border-radius: 10px;
}

.intro-content {
    max-width: 800px;
    margin: 0 auto;
}

#chat-container {
```

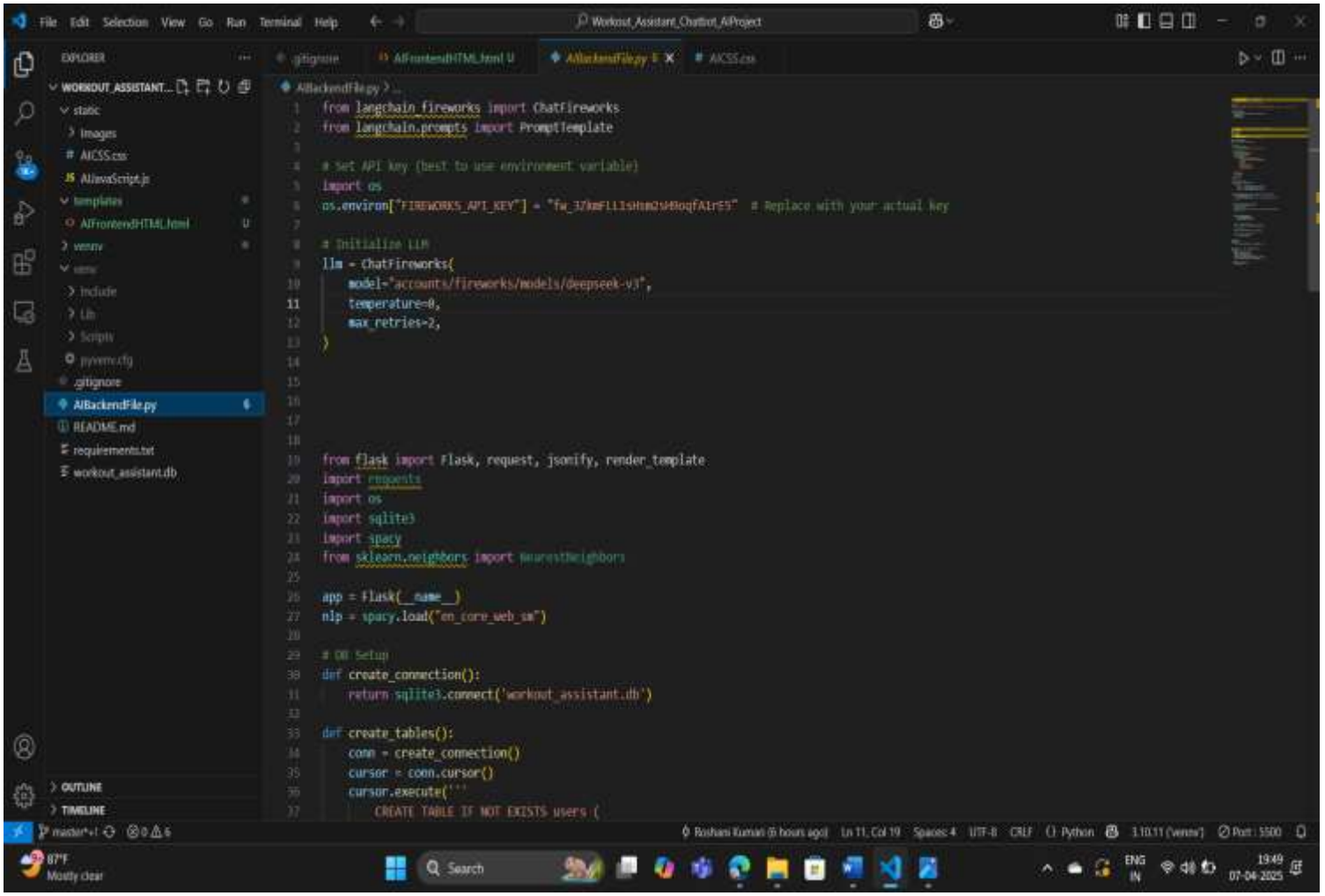


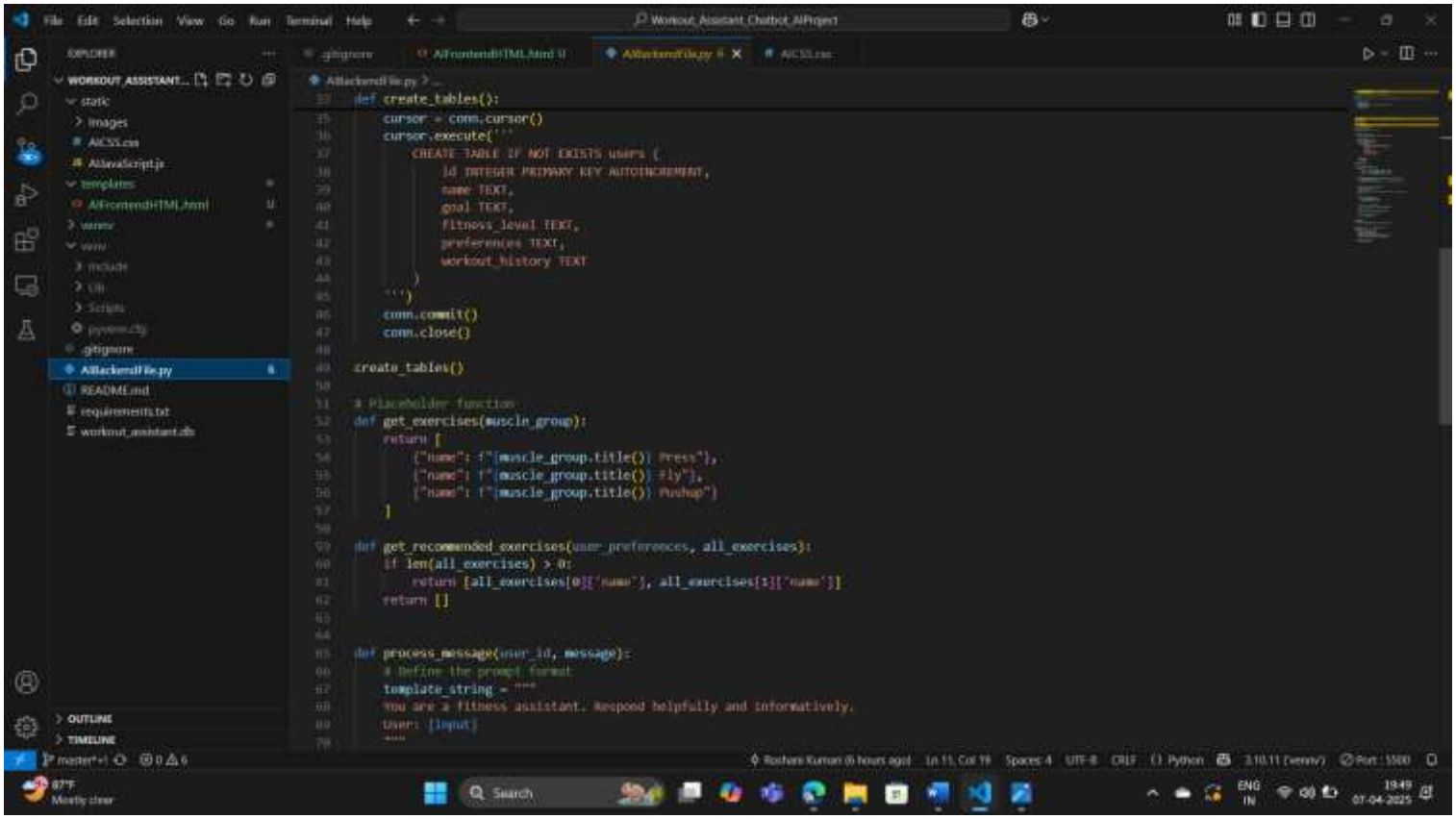
```css
}

#chat-container {
    width: 400px;
    margin: 20px auto;
    border: 1px solid #ccc;
    height: 500px;
    display: flex;
    flex-direction: column;
    background-color: rgba(255, 255, 255, 0.8);
    border-radius: 10px;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
    overflow: hidden;
    position: relative; /* Make chat-container a relative container */
}

#chat-log {
    flex-grow: 1;
    overflow-y: auto;
    padding: 20px;
}

#user-input {
    display: flex;
    padding: 10px;
    border-top: 1px solid #eee;
    align-items: center;
    position: absolute;
    bottom: 0;
    width: 100%;
}

#message-input {
    flex-grow: 1;
    padding: 10px;
    border: 1px solid #ddd;
    border-radius: 5px;
```


#message-input {
flex-grow: 1;
padding: 10px;
border: 1px solid #ddd;
border-radius: 5px;
font-size: 16px;
margin-right: 10px;
}
#send-button {
padding: 10px 20px;
background-color: #007bff;
color: white;
border: none;
border-radius: 5px;
cursor: pointer;
font-size: 16px;
}
.message {
margin-bottom: 15px;
padding: 12px;
border-radius: 20px;
display: flex;
align-items: center;
}
.user-message {
text-align: right;
background-color: #e1f5fe;
color: #1976d2;
}
.bot-message {
background-color: #f0f4c3;
color: #689f38;

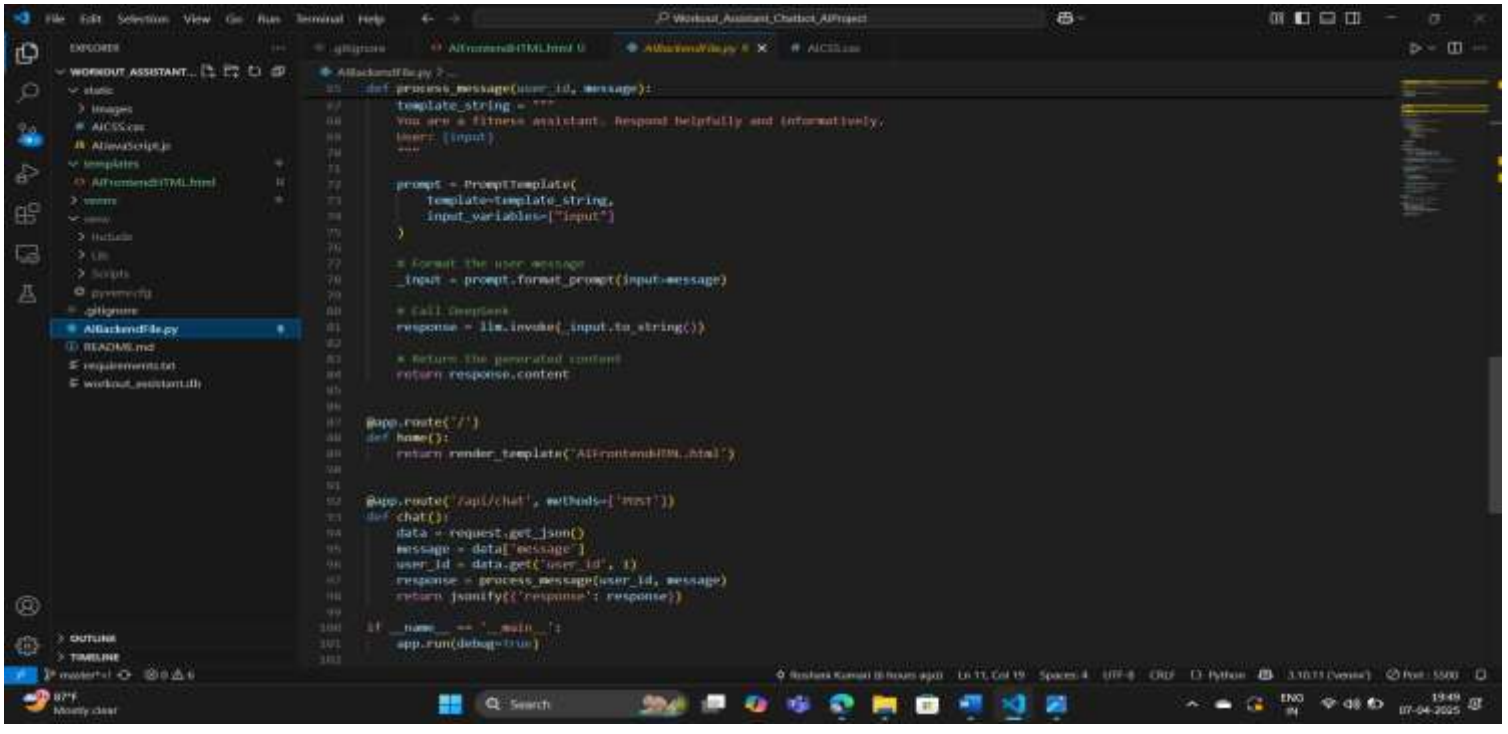

File Edit Selection View Go Run Terminal Help
Workout_Assistant_Chatbot_AIProject
EXPLORER
WORKOUT_ASSISTANT_CHATBOT_AIPROJE...
static
Images
AICSS.css
AIJavaScript.js
templates
AIFrontendHTML.html
venv
include
Lib
Scripts
pyvenv.cfg
.gitignore
AIBackendFile.py
README.md
requirements.txt
workout_assistant.db
OUTLINE
TIMELINE
.gitignore
AIBackendFile.py
AIFrontendHTML.html
AICSS.css
static > AICSS.css > #chat-container
.message-i
}
.user-message {
text-align: right;
background-color: #e1f5fe;
color: #1976d2;
}
.bot-message {
background-color: #f0f4c3;
color: #827717;
}
.message-icon, .bot-icon {
width: 40px;
height: 40px;
margin-right: 10px;
}
footer {
text-align: center;
padding: 20px 0;
margin-top: 20px;
background-color: rgba(255, 255, 255, 0.8);
}
#registration-info {
position: absolute;
top: 150px;
left: 20px;
background-color: rgba(255, 255, 255, 0.8);
padding: 10px;
border-radius: 5px;
font-size: 14px;
}
87°F
Mostly clear
Search
ENG
IN
19:36
07-04-2025


File Edit Selection View Go Run Terminal Help
Workout_Assistant_Chatbot_AIProject
EXPLORER
WORKOUT_ASSISTANT...
static
Images
AICSS.css
AIJavaScript.js
templates
AIFrontendHTML.html
venv
env
Include
Lib
Scripts
pyvenv.cfg
.gitignore
AIBackendFile.py
README.md
requirements.txt
workout_assistant.db
.gitignore
AIFrontendHTML.html U
AIJavaScript.js
AICSS.css
static > AIJavaScript.js > appendMessage
const chatLog = document.getElementById('chat-log');
const messageInput = document.getElementById('message-input');
const sendButton = document.getElementById('send-button');
function appendMessage(message, className) {
const messageDiv = document.createElement('div');
messageDiv.className = `message ${className}`;
messageDiv.innerHTML = `<span>${message}</span>`;
chatLog.appendChild(messageDiv);
chatLog.scrollTop = chatLog.scrollHeight;
}
sendButton.addEventListener('click', () => {
const message = messageInput.value;
if (message.trim() !== '') {
appendMessage(message, 'user-message');
messageInput.value = '';
fetch('/api/chat', {
method: 'POST',
headers: {
'Content-Type': 'application/json'
},
body: JSON.stringify({ message: message })
})
.then(response => response.json())
.then(data => {
appendMessage(data.response, 'bot-message');
})
.catch(error => {
console.error('Error:', error);
appendMessage('Sorry, something went wrong.', 'bot-message');
});
}
});
OUTLINE
TIMELINE
master*
Rashami Kumari (6 hours ago)
Ln 10, Col 46
Spaces: 4
UTF-8
CRLF
JavaScript
Port : 5500
87°F
Mostly clear
Search
ENG
IN
19:48
07-04-2025


File Edit Selection View Go Run Terminal Help
Workout_Assistant_Chatbot_AIProject
EXPLORER
WORKOUT_ASSISTANT...
static
Images
AICSS.css
AIJavaScript.js
templates
AIFrontendHTML.html
venv
include
Lib
Scripts
pyvenv.cfg
.gitignore
AIBackendFile.py
README.md
requirements.txt
workout_assistant.db
OUTLINE
TIMELINE
.gitignore
AIFrontendHTML.html U
AIBackendFile.py
AICSS.css
AIBackendFile.py > ...
from langchain_fireworks import ChatFireworks
from langchain.prompts import PromptTemplate
# Set API key (best to use environment variable)
import os
os.environ["FIREWORKS_API_KEY"] = "fw_3ZkmFL11sHnm2sW4oqfA1rE5" # Replace with your actual key
# Initialize LLM
llm = ChatFireworks(
model="accounts/fireworks/models/deepseek-v3",
temperature=0,
max_retries=2,
)
from flask import Flask, request, jsonify, render_template
import requests
import os
import sqlite3
import spacy
from sklearn.neighbors import NearestNeighbors
app = Flask(__name__)
nlp = spacy.load("en_core_web_sm")
# DB Setup
def create_connection():
return sqlite3.connect('workout_assistant.db')
def create_tables():
conn = create_connection()
cursor = conn.cursor()
cursor.execute('''
CREATE TABLE IF NOT EXISTS users (
Roshan Kumar (6 hours ago) Ln 11, Col 19 Spaces: 4 UTF-8 CRLF Python 3.10.11 ('venv') Port : 5500
master*↓↑ 0 6
87°F
Mostly clear
Search
ENG
IN
19:49
07-04-2025



```python
def create_tables():
    cursor = conn.cursor()
    cursor.execute('''
        CREATE TABLE IF NOT EXISTS users (
            id INTEGER PRIMARY KEY AUTOINCREMENT,
            name TEXT,
            goal TEXT,
            fitness_level TEXT,
            preferences TEXT,
            workout_history TEXT
        )
    ''')
    conn.commit()
    conn.close()

create_tables()

# Placeholder function
def get_exercises(muscle_group):
    return [
        {"name": f"{muscle_group.title()} Press"},
        {"name": f"{muscle_group.title()} Fly"},
        {"name": f"{muscle_group.title()} Pushup"}
    ]

def get_recommended_exercises(user_preferences, all_exercises):
    if len(all_exercises) > 0:
        return [all_exercises[0]['name'], all_exercises[1]['name']]
    return []


def process_message(user_id, message):
    # Define the prompt format
    template_string = """
    You are a fitness assistant. Respond helpfully and informatively.
    User: {input}
    """
```



```python
def process_message(user_id, message):
    template_string = """
    You are a fitness assistant. Respond helpfully and informatively.
    User: {input}
    """

    prompt = PromptTemplate(
        template=template_string,
        input_variables=["input"]
    )

    # Format the user message
    _input = prompt.format_prompt(input=message)

    # Call DeepSeek
    response = llm.invoke(_input.to_string())

    # Return the generated content
    return response.content


@app.route('/')
def home():
    return render_template('AIFrontendHTML.html')


@app.route('/api/chat', methods=['POST'])
def chat():
    data = request.get_json()
    message = data['message']
    user_id = data.get('user_id', 1)
    response = process_message(user_id, message)
    return jsonify({'response': response})

if __name__ == '__main__':
    app.run(debug=True)
```

Project Link On Github

Link: https://github.com/kripalnathtiwari/AI-Based-Time-Managment-S

Conclusion

This project successfully developed a personalized fitness chatbot, 'Workout Assistant', utilizing advanced AI models to provide tailored workout and nutritional advice.

By creating an accessible and conversational interface, the chatbot aims to simplify fitness guidance and enhance user engagement.

The implementation demonstrates the potential of AI in delivering personalized health support, paving the way for future advancements in AI-driven fitness applications.