LEARNINGAGENT.PY

import json

# Function to load user preferences data

def load\_user\_data():

    try:

        with open('data/user\_data.json') as f:

            return json.load(f)

    except FileNotFoundError:

        return {}

def save\_user\_feedback(user\_id, destination, liked):

    user\_data = load\_user\_data()

    # Initialize user feedback data if not present

    if user\_id not in user\_data:

        user\_data[user\_id] = {'feedback': []}

    # Check if the user has already given feedback for the destination

    feedback\_found = False

    for feedback in user\_data[user\_id]['feedback']:

        if feedback['destination'] == destination:

            # Update the existing feedback

            feedback['liked'] = liked

            feedback\_found = True

            break

    # If no existing feedback, append new feedback

    if not feedback\_found:

        user\_data[user\_id]['feedback'].append({'destination': destination, 'liked': liked})

    with open('data/user\_data.json', 'w') as f:

        json.dump(user\_data, f, indent=4)

# Function to get recommendations based on user preferences

def get\_recommendations(preferences, destinations):

    recommendations = []

    for destination in destinations:

        if (destination["type"].lower() == preferences["location\_type"] and

            destination["budget"].lower() == preferences["budget"] and

            destination["season"].lower() == preferences["season"] and

            any(activity.lower() in preferences["activities"] for activity in destination["activities"]) and

            destination["num\_days"] >= preferences["num\_days"]):

            recommendations.append(destination)

    return recommendations

def update\_recommendations(user\_id, destinations):

    user\_data = load\_user\_data()

    # Check if the user has any feedback

    if user\_id not in user\_data or 'feedback' not in user\_data[user\_id]:

        return destinations

    feedback = user\_data[user\_id]['feedback']

    liked\_destinations = [f['destination'] for f in feedback if f['liked']]

    # Print the liked destinations for debugging

    print(f"Liked Destinations: {liked\_destinations}")

    # Update the recommendations based on feedback

    for dest in destinations:

        print(f"Checking destination: {dest['name']}")

        # If the destination is liked, increase the score; if disliked, decrease the score

        if dest["name"] in liked\_destinations:

            dest["score"] = dest.get("score", 0) + 1  # Increase score for liked destinations

        else:

            dest["score"] = max(dest.get("score", 0) - 1, 0)  # Decrease score for disliked destinations

        # Print each updated destination for debugging

        print(f"Updated Destination: {dest['name']} with score {dest['score']}")

    # Sort destinations based on score

    destinations.sort(key=lambda x: x.get("score", 0), reverse=True)

    # Print sorted destinations for debugging

    print(f"Sorted Destinations: {[d['name'] for d in destinations]}")

    # Save the updated destinations back to destination.json

    with open('data/destination.json', 'w') as f:

        json.dump({"destinations": destinations}, f, indent=4)

    return destinations

DESTINATION.JSON

{

  "destinations": [

    { "name": "Maldives", "type": "Beach", "budget": "High", "season": "Summer", "activities": ["Swimming", "Hiking", "Snorkeling"], "num\_days": 7, "score": 90 },

    { "name": "Switzerland", "type": "Mountain", "budget": "Medium", "season": "Winter", "activities": ["Skiing", "Shopping"], "num\_days": 10, "score": 85 },

    { "name": "Paris", "type": "City", "budget": "Low", "season": "Spring", "activities": ["Sightseeing", "Shopping"], "num\_days": 5, "score": 75 },

    { "name": "New York", "type": "City", "budget": "High", "season": "Autumn", "activities": ["Sightseeing", "Museums"], "num\_days": 4, "score": 80 },

    { "name": "Tokyo", "type": "City", "budget": "Medium", "season": "Spring", "activities": ["Shopping", "Sightseeing", "Food tours"], "num\_days": 6, "score": 88 },

    { "name": "Hawaii", "type": "Beach", "budget": "High", "season": "Summer", "activities": ["Swimming", "Snorkeling", "Surfing"], "num\_days": 7, "score": 92 },

    { "name": "Berlin", "type": "City", "budget": "Low", "season": "Autumn", "activities": ["Museums", "Shopping"], "num\_days": 5, "score": 78 },

    { "name": "Dubai", "type": "City", "budget": "High", "season": "Winter", "activities": ["Shopping", "Sightseeing", "Beaches"], "num\_days": 5, "score": 85 },

    { "name": "Iceland", "type": "Mountain", "budget": "Medium", "season": "Winter", "activities": ["Hiking", "Sightseeing"], "num\_days": 8, "score": 87 },

    { "name": "Barcelona", "type": "City", "budget": "Medium", "season": "Spring", "activities": ["Sightseeing", "Beaches"], "num\_days": 6, "score": 82 },

    { "name": "Cairo", "type": "City", "budget": "Low", "season": "Winter", "activities": ["Sightseeing", "Pyramids"], "num\_days": 4, "score": 76 },

    { "name": "Rome", "type": "City", "budget": "Medium", "season": "Spring", "activities": ["Sightseeing", "Shopping"], "num\_days": 5, "score": 83 },

    { "name": "Machu Picchu", "type": "Mountain", "budget": "Medium", "season": "Spring", "activities": ["Hiking", "Sightseeing"], "num\_days": 9, "score": 90 },

    { "name": "Tahiti", "type": "Beach", "budget": "High", "season": "Summer", "activities": ["Snorkeling", "Swimming"], "num\_days": 7, "score": 89 },

    { "name": "Sydney", "type": "City", "budget": "Medium", "season": "Autumn", "activities": ["Sightseeing", "Beaches"], "num\_days": 6, "score": 84 },

    { "name": "London", "type": "City", "budget": "Low", "season": "Autumn", "activities": ["Sightseeing", "Museums"], "num\_days": 4, "score": 77 },

    { "name": "Bali", "type": "Beach", "budget": "Medium", "season": "Spring", "activities": ["Swimming", "Snorkeling", "Hiking"], "num\_days": 6, "score": 81 },

    { "name": "Cape Town", "type": "City", "budget": "Medium", "season": "Winter", "activities": ["Sightseeing", "Beaches"], "num\_days": 5, "score": 79 },

    { "name": "Kyoto", "type": "City", "budget": "Medium", "season": "Spring", "activities": ["Sightseeing", "Temples"], "num\_days": 5, "score": 84 },

    { "name": "Madrid", "type": "City", "budget": "Low", "season": "Spring", "activities": ["Sightseeing", "Museums"], "num\_days": 5, "score": 73 },

    { "name": "Los Angeles", "type": "City", "budget": "High", "season": "Summer", "activities": ["Sightseeing", "Shopping", "Beaches"], "num\_days": 6, "score": 86 },

    { "name": "Amsterdam", "type": "City", "budget": "Medium", "season": "Spring", "activities": ["Sightseeing", "Museums"], "num\_days": 6, "score": 80 },

    { "name": "Bangkok", "type": "City", "budget": "Low", "season": "Winter", "activities": ["Sightseeing", "Shopping", "Food tours"], "num\_days": 5, "score": 74 },

    { "name": "Vienna", "type": "City", "budget": "Medium", "season": "Winter", "activities": ["Sightseeing", "Museums"], "num\_days": 6, "score": 85 },

    { "name": "Athens", "type": "City", "budget": "Low", "season": "Spring", "activities": ["Sightseeing", "History tours"], "num\_days": 4, "score": 70 },

    { "name": "Prague", "type": "City", "budget": "Low", "season": "Autumn", "activities": ["Sightseeing", "Food tours"], "num\_days": 4, "score": 72 },

    { "name": "Zermatt", "type": "Mountain", "budget": "High", "season": "Winter", "activities": ["Skiing", "Sightseeing"], "num\_days": 8, "score": 88 },

    { "name": "Singapore", "type": "City", "budget": "High", "season": "Summer", "activities": ["Sightseeing", "Shopping"], "num\_days": 5, "score": 82 },

    { "name": "Paris", "type": "City", "budget": "Low", "season": "Winter", "activities": ["Sightseeing", "Museums"], "num\_days": 5, "score": 76 },

    { "name": "Queenstown", "type": "Mountain", "budget": "Medium", "season": "Winter", "activities": ["Skiing", "Sightseeing"], "num\_days": 8, "score": 85 },

    { "name": "Madrid", "type": "City", "budget": "Medium", "season": "Spring", "activities": ["Sightseeing", "Museums"], "num\_days": 5, "score": 80 },

    { "name": "Morocco", "type": "Desert", "budget": "Low", "season": "Autumn", "activities": ["Sightseeing", "Camel riding"], "num\_days": 4, "score": 68 },

    { "name": "Buenos Aires", "type": "City", "budget": "Medium", "season": "Winter", "activities": ["Sightseeing", "Food tours"], "num\_days": 6, "score": 80 },

    { "name": "Miami", "type": "Beach", "budget": "High", "season": "Summer", "activities": ["Beaches", "Sightseeing"], "num\_days": 6, "score": 87 },

    { "name": "Venice", "type": "City", "budget": "Medium", "season": "Spring", "activities": ["Sightseeing", "Shopping"], "num\_days": 5, "score": 79 },

    { "name": "Amsterdam", "type": "City", "budget": "Medium", "season": "Autumn", "activities": ["Sightseeing", "Museums"], "num\_days": 5, "score": 78 },

    { "name": "Athens", "type": "City", "budget": "Low", "season": "Spring", "activities": ["Sightseeing", "History tours"], "num\_days": 4, "score": 71 },

    { "name": "Helsinki", "type": "City", "budget": "Low", "season": "Winter", "activities": ["Sightseeing", "Shopping"], "num\_days": 5, "score": 73 },

    { "name": "Seoul", "type": "City", "budget": "Medium", "season": "Spring", "activities": ["Sightseeing", "Shopping"], "num\_days": 6, "score": 84 },

    { "name": "Oslo", "type": "City", "budget": "Medium", "season": "Winter", "activities": ["Sightseeing", "Hiking"], "num\_days": 7, "score": 86 },

    { "name": "Jakarta", "type": "City", "budget": "Low", "season": "Spring", "activities": ["Sightseeing", "Shopping"], "num\_days": 4, "score": 75 },

    { "name": "Bangkok", "type": "City", "budget": "Low", "season": "Summer", "activities": ["Sightseeing", "Shopping"], "num\_days": 4, "score": 72 },

    { "name": "Lisbon", "type": "City", "budget": "Medium", "season": "Autumn", "activities": ["Sightseeing", "Museums"], "num\_days": 5, "score": 80 },

    { "name": "Dubai", "type": "City", "budget": "High", "season": "Winter", "activities": ["Sightseeing", "Shopping"], "num\_days": 5, "score": 85 },

    { "name": "Fiji", "type": "Beach", "budget": "High", "season": "Summer", "activities": ["Swimming", "Snorkeling"], "num\_days": 7, "score": 90 },

    { "name": "Tahiti", "type": "Beach", "budget": "High", "season": "Summer", "activities": ["Swimming", "Snorkeling"], "num\_days": 7, "score": 90 },

    { "name": "Maldives", "type": "Beach", "budget": "Low", "season": "Winter", "activities": ["Snorkeling", "Relaxing"], "num\_days": 5, "score": 80 },

    { "name": "Switzerland", "type": "Mountain", "budget": "High", "season": "Spring", "activities": ["Skiing", "Snowboarding"], "num\_days": 12, "score": 95 },

    { "name": "Paris", "type": "City", "budget": "Medium", "season": "Autumn", "activities": ["Sightseeing", "Gastronomy"], "num\_days": 6, "score": 88 },

    { "name": "New York", "type": "City", "budget": "Medium", "season": "Winter", "activities": ["Broadway", "Sightseeing"], "num\_days": 7, "score": 86 },

    { "name": "Tokyo", "type": "City", "budget": "High", "season": "Winter", "activities": ["Food tours", "Sightseeing"], "num\_days": 7, "score": 89 },

    { "name": "Hawaii", "type": "Beach", "budget": "Medium", "season": "Spring", "activities": ["Relaxing", "Sightseeing"], "num\_days": 8, "score": 82 },

    { "name": "Berlin", "type": "City", "budget": "High", "season": "Spring", "activities": ["Historical tours", "Museums"], "num\_days": 6, "score": 83 },

    { "name": "Dubai", "type": "City", "budget": "Medium", "season": "Summer", "activities": ["Shopping", "Sightseeing"], "num\_days": 4, "score": 78 },

    { "name": "Iceland", "type": "Mountain", "budget": "Low", "season": "Spring", "activities": ["Sightseeing", "Hiking"], "num\_days": 9, "score": 84 },

    { "name": "Barcelona", "type": "City", "budget": "High", "season": "Summer", "activities": ["Beach activities", "Sightseeing"], "num\_days": 7, "score": 92 },

    { "name": "Cairo", "type": "City", "budget": "Medium", "season": "Autumn", "activities": ["Cultural tours", "Sightseeing"], "num\_days": 5, "score": 81 },

    { "name": "Rome", "type": "City", "budget": "High", "season": "Summer", "activities": ["Cultural tours", "Dining"], "num\_days": 6, "score": 88 },

    { "name": "Machu Picchu", "type": "Mountain", "budget": "Medium", "season": "Winter", "activities": ["Sightseeing", "Hiking"], "num\_days": 10, "score": 89 },

    { "name": "Tahiti", "type": "Beach", "budget": "Medium", "season": "Autumn", "activities": ["Swimming", "Snorkeling"], "num\_days": 6, "score": 85 },

    { "name": "Sydney", "type": "City", "budget": "Low", "season": "Spring", "activities": ["Sightseeing", "Beaches"], "num\_days": 5, "score": 80 },

    { "name": "London", "type": "City", "budget": "Medium", "season": "Summer", "activities": ["Theatre", "Shopping"], "num\_days": 6, "score": 87 },

    { "name": "Bali", "type": "Beach", "budget": "High", "season": "Winter", "activities": ["Relaxing", "Sightseeing"], "num\_days": 7, "score": 91 },

    { "name": "Cape Town", "type": "City", "budget": "Low", "season": "Summer", "activities": ["Beaches", "Wine tours"], "num\_days": 5, "score": 75 },

    { "name": "Kyoto", "type": "City", "budget": "High", "season": "Autumn", "activities": ["Cultural tours", "Temples"], "num\_days": 7, "score": 92 },

    { "name": "Madrid", "type": "City", "budget": "Low", "season": "Autumn", "activities": ["Sightseeing", "Food tours"], "num\_days": 5, "score": 79 },

    { "name": "Los Angeles", "type": "City", "budget": "Medium", "season": "Winter", "activities": ["Sightseeing", "Beaches"], "num\_days": 7, "score": 85 },

    { "name": "Amsterdam", "type": "City", "budget": "Low", "season": "Spring", "activities": ["Biking", "Museums"], "num\_days": 6, "score": 77 },

    { "name": "Bangkok", "type": "City", "budget": "High", "season": "Winter", "activities": ["Shopping", "Food tours"], "num\_days": 5, "score": 88 },

    { "name": "Vienna", "type": "City", "budget": "Medium", "season": "Spring", "activities": ["Opera", "Sightseeing"], "num\_days": 6, "score": 82 },

    { "name": "Athens", "type": "City", "budget": "Medium", "season": "Spring", "activities": ["Archaeological tours", "Museums"], "num\_days": 6, "score": 84 },

    { "name": "Prague", "type": "City", "budget": "Low", "season": "Spring", "activities": ["Sightseeing", "Food tours"], "num\_days": 4, "score": 75 },

    { "name": "Zermatt", "type": "Mountain", "budget": "High", "season": "Winter", "activities": ["Skiing", "Snowboarding"], "num\_days": 8, "score": 90 },

    { "name": "Singapore", "type": "City", "budget": "Medium", "season": "Summer", "activities": ["Shopping", "Sightseeing"], "num\_days": 6, "score": 85 },

    { "name": "Paris", "type": "City", "budget": "High", "season": "Spring", "activities": ["Wine tours", "Cultural tours"], "num\_days": 6, "score": 90 },

    { "name": "Queenstown", "type": "Mountain", "budget": "Medium", "season": "Spring", "activities": ["Skiing", "Sightseeing"], "num\_days": 9, "score": 86 },

    { "name": "Madrid", "type": "City", "budget": "High", "season": "Summer", "activities": ["Food tours", "Shopping"], "num\_days": 5, "score": 88 },

    { "name": "Morocco", "type": "Desert", "budget": "Medium", "season": "Winter", "activities": ["Camel riding", "Sightseeing"], "num\_days": 6, "score": 84 },

    { "name": "Buenos Aires", "type": "City", "budget": "Low", "season": "Spring", "activities": ["Sightseeing", "Food tours"], "num\_days": 6, "score": 78 },

    { "name": "Vancouver", "type": "City", "budget": "Low", "season": "Summer", "activities": ["Sightseeing", "Hiking"], "num\_days": 6, "score": 83 },

    { "name": "Bangkok", "type": "City", "budget": "Low", "season": "Winter", "activities": ["Shopping", "Sightseeing"], "num\_days": 5, "score": 78 },

    { "name": "Seoul", "type": "City", "budget": "Medium", "season": "Autumn", "activities": ["Sightseeing", "Food tours"], "num\_days": 6, "score": 84 },

    { "name": "Marrakech", "type": "City", "budget": "Low", "season": "Spring", "activities": ["Cultural tours", "Shopping"], "num\_days": 5, "score": 80 },

    { "name": "Buenos Aires", "type": "City", "budget": "Medium", "season": "Summer", "activities": ["Sightseeing", "Food tours"], "num\_days": 6, "score": 82 },

    { "name": "San Francisco", "type": "City", "budget": "Medium", "season": "Spring", "activities": ["Sightseeing", "Food tours"], "num\_days": 7, "score": 85 },

    { "name": "Oslo", "type": "City", "budget": "Medium", "season": "Winter", "activities": ["Sightseeing", "Museums"], "num\_days": 6, "score": 82 },

    { "name": "Lisbon", "type": "City", "budget": "Medium", "season": "Spring", "activities": ["Sightseeing", "Cultural tours"], "num\_days": 6, "score": 88 },

    { "name": "Hong Kong", "type": "City", "budget": "High", "season": "Autumn", "activities": ["Shopping", "Sightseeing"], "num\_days": 7, "score": 91 },

    { "name": "Kuala Lumpur", "type": "City", "budget": "Low", "season": "Summer", "activities": ["Sightseeing", "Food tours"], "num\_days": 5, "score": 79 },

    { "name": "Copenhagen", "type": "City", "budget": "Medium", "season": "Autumn", "activities": ["Sightseeing", "Cultural tours"], "num\_days": 6, "score": 83 },

    { "name": "Sydney", "type": "City", "budget": "Low", "season": "Summer", "activities": ["Sightseeing", "Beaches"], "num\_days": 7, "score": 85 },

    { "name": "Rio de Janeiro", "type": "City", "budget": "Medium", "season": "Winter", "activities": ["Sightseeing", "Beaches"], "num\_days": 6, "score": 84 },

    { "name": "Melbourne", "type": "City", "budget": "Medium", "season": "Spring", "activities": ["Cultural tours", "Shopping"], "num\_days": 5, "score": 80 },

    { "name": "Rome", "type": "City", "budget": "High", "season": "Winter", "activities": ["Sightseeing", "Cultural tours"], "num\_days": 5, "score": 85 },

    { "name": "Toronto", "type": "City", "budget": "Medium", "season": "Autumn", "activities": ["Sightseeing", "Cultural tours"], "num\_days": 6, "score": 84 },

    { "name": "Barcelona", "type": "City", "budget": "Low", "season": "Autumn", "activities": ["Sightseeing", "Food tours"], "num\_days": 5, "score": 83 },

    { "name": "Dubai", "type": "City", "budget": "High", "season": "Summer", "activities": ["Shopping", "Sightseeing"], "num\_days": 6, "score": 87 },

    { "name": "London", "type": "City", "budget": "High", "season": "Winter", "activities": ["Shopping", "Sightseeing"], "num\_days": 6, "score": 89 },

    { "name": "Madrid", "type": "City", "budget": "Medium", "season": "Spring", "activities": ["Food tours", "Shopping"], "num\_days": 6, "score": 84 },

    { "name": "Berlin", "type": "City", "budget": "Low", "season": "Winter", "activities": ["Sightseeing", "Museums"], "num\_days": 5, "score": 80 }

  ]

}

USER\_DATA.JSON

{}

STYLES.CSS

/\* Global Styles \*/

body {

    font-family: 'Arial', sans-serif;

    background-color: #f4f6f9;

    color: #333;

    margin: 0;

    padding: 0;

}

/\* Navigation Bar \*/

nav {

    background-color: #3b9b94;

    color: white;

    padding: 1em;

}

nav a {

    color: white;

    margin-right: 15px;

    text-decoration: none;

    font-weight: bold;

}

nav a:hover {

    color: #ffcccb;

    transition: color 0.3s ease;

}

/\* Buttons \*/

button {

    background-color: #ff5722;

    color: white;

    border: none;

    padding: 10px 20px;

    cursor: pointer;

    border-radius: 5px;

    font-weight: bold;

    transition: background-color 0.3s ease, transform 0.2s ease;

}

button:hover {

    background-color: #e64a19;

    transform: translateY(-2px);

}

/\* Hero Section \*/

.hero {

    background-image: url('https://www.example.com/hero-image.jpg');

    background-size: cover;

    padding: 80px 20px;

    color: white;

    text-align: center;

    background-attachment: fixed;

}

.hero h1 {

    font-size: 3em;

    margin-bottom: 20px;

}

.hero p {

    font-size: 1.5em;

}

/\* Recommendations \*/

.recommendations {

    display: flex;

    flex-wrap: wrap;

    gap: 20px;

    justify-content: center;

    padding: 20px;

}

.recommendation-card {

    background-color: white;

    border-radius: 8px;

    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

    width: 250px;

    overflow: hidden;

    text-align: center;

    transition: transform 0.3s ease, box-shadow 0.3s ease;

}

.recommendation-card img {

    width: 100%;

    height: 150px;

    object-fit: cover;

    border-top-left-radius: 8px;

    border-top-right-radius: 8px;

}

.recommendation-card h3 {

    color: #3b9b94;

    margin-top: 15px;

    font-size: 1.5em;

}

.recommendation-card p {

    color: #555;

    padding: 0 15px;

    font-size: 0.9em;

}

.recommendation-card:hover {

    transform: scale(1.05);

    box-shadow: 0 8px 16px rgba(0, 0, 0, 0.2);

}

/\* Footer \*/

footer {

    background-color: #3b9b94;

    color: white;

    text-align: center;

    padding: 1em 0;

    font-size: 0.9em;

}

/\* Media Queries for Responsiveness \*/

@media (max-width: 768px) {

    .hero h1 {

        font-size: 2.5em;

    }

    .hero p {

        font-size: 1.2em;

    }

    .recommendation-card {

        width: 100%;

    }

}

.weather-container {

    background-color: #e0f7fa;

    padding: 20px;

    border-radius: 8px;

    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

    max-width: 400px;

    margin: auto;

    text-align: center;

}

.weather-container h2 {

    color: #3b9b94;

}

.weather-container p {

    font-size: 1.1em;

    color: #333;

}

.weather-display {

    background-color: #e0f7fa;

    padding: 15px;

    border-radius: 8px;

    margin-top: 20px;

}

.weather-display h4 {

    color: #3b9b94;

}

INDEX.HTML

{% extends "layout.html" %}

{% block content %}

<div class="row">

    <div class="col-12 col-md-6">

        <h2>Plan Your Next Holiday</h2>

        <form method="POST">

            <div class="mb-3">

                <label for="location\_type" class="form-label">Location Type</label>

                <select class="form-select" id="location\_type" name="location\_type" required>

                    <option value="Beach">Beach</option>

                    <option value="Mountain">Mountain</option>

                    <option value="City">City</option>

                </select>

            </div>

            <div class="mb-3">

                <label for="budget" class="form-label">Budget</label>

                <select class="form-select" id="budget" name="budget" required>

                    <option value="Low">Low</option>

                    <option value="Medium">Medium</option>

                    <option value="High">High</option>

                </select>

            </div>

            <div class="mb-3">

                <label for="season" class="form-label">Season</label>

                <select class="form-select" id="season" name="season" required>

                    <option value="Summer">Summer</option>

                    <option value="Winter">Winter</option>

                    <option value="Spring">Spring</option>

                </select>

            </div>

            <div class="form-group">

                <label for="num\_days">Number of Days</label>

                <input type="number" class="form-control" id="num\_days" name="num\_days" min="1" required>

            </div><br>

            <div class="mb-3">

                <label for="activities" class="form-label">Activities</label><br>

                <input type="checkbox" id="Swimming" name="activities" value="Swimming"> Swimming<br>

                <input type="checkbox" id="Hiking" name="activities" value="Hiking"> Hiking<br>

                <input type="checkbox" id="Skiing" name="activities" value="Skiing"> Skiing<br>

                <input type="checkbox" id="Shopping" name="activities" value="Shopping"> Shopping<br>

                <input type="checkbox" id="Sightseeing" name="activities" value="Sightseeing"> Sightseeing<br>

            </div><br>

            <button type="submit" class="btn btn-primary">Submit</button>

        </form>

    </div>

    <div class="col-12 col-md-6">

        <h3>Weather Info</h3>

        <form action="/weather" method="get">

            <input type="text" name="city" class="form-control" placeholder="Enter city" required>

            <button type="submit" class="btn btn-info mt-2">Get Weather</button>

        </form>

        {% if weather %}

        <div class="mt-3 weather-display">

            <h4>Weather in {{ weather.name }}</h4>

            <p><strong>Temperature:</strong> {{ weather.main.temp }}°C</p>

            <p><strong>Description:</strong> {{ weather.weather[0].description | capitalize }}</p>

        </div>

        {% endif %}

    </div>

</div>

{% endblock %}

LAYOUT.HTML

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Holiday Planner</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet">

    <link href="{{ url\_for('static', filename='css/styles.css') }}" rel="stylesheet">

</head>

<body>

    <div class="container">

        <nav class="navbar navbar-expand-lg navbar-light bg-light">

            <a class="navbar-brand" href="/">Holiday Planner</a>

            <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">

                <span class="navbar-toggler-icon"></span>

            </button>

            <div class="collapse navbar-collapse" id="navbarNav">

                <ul class="navbar-nav ml-auto">

                    {% if current\_user.is\_authenticated %}

                        <li class="nav-item"><a class="nav-link" href="/logout">Logout</a></li>

                    {% else %}

                        <li class="nav-item"><a class="nav-link" href="/login">Login</a></li>

                    {% endif %}

                </ul>

            </div>

        </nav>

        {% block content %}{% endblock %}

    </div>

    <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js"></script>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.min.js"></script>

</body>

</html>

LOGIN.HTML

{% extends "layout.html" %}

{% block content %}

<div class="d-flex justify-content-center align-items-center" style="height: 80vh;">

    <div class="card p-4" style="max-width: 400px; width: 100%;">

        <h2 class="text-center mb-4">Login</h2>

        {% if error %}

            <div class="alert alert-danger" role="alert">

                {{ error }}

            </div>

        {% endif %}

        <form method="POST" action="/login">

            <div class="mb-3">

                <label for="username" class="form-label">Username</label>

                <input type="text" class="form-control" id="username" name="username" required>

            </div>

            <div class="mb-3">

                <label for="password" class="form-label">Password</label>

                <input type="password" class="form-control" id="password" name="password" required>

            </div>

            <button type="submit" class="btn btn-primary w-100">Login</button>

        </form>

    </div>

</div>

{% endblock %}

RESULTS.HTML

{% extends "layout.html" %}

{% block content %}

<h1 class="text-center mb-4">Recommendations</h1>

<!-- Display selected dates and number of days -->

{% if preferences %}

    <p><strong>Number of Days:</strong> {{ preferences.num\_days }}</p>

{% endif %}

{% if recommendations %}

    <div class="row">

        {% for destination in recommendations %}

            <div class="col-md-4">

                <div class="card mb-4">

                    <div class="card-body">

                        <h5 class="card-title">{{ destination }}</h5>

                        <form action="/feedback/{{ destination }}" method="post">

                            <button type="submit" name="feedback" value="liked" class="btn btn-success">Like</button>

                            <button type="submit" name="feedback" value="disliked" class="btn btn-danger">Dislike</button>

                        </form>

                    </div>

                </div>

            </div>

        {% endfor %}

    </div>

{% else %}

    <p>No matching destinations found.</p>

{% endif %}

<div class="text-center mt-4">

    <a href="{{ url\_for('home') }}" class="btn btn-primary">Back to Home</a>

</div>

{% endblock %}

WEATHER.HTML

{% extends "layout.html" %}

{% block content %}

<div class="weather-container">

    {% if weather %}

        <h2>Weather in {{ weather.name }}</h2>

        <p><strong>Temperature:</strong> {{ weather.main.temp }}°C</p>

        <p><strong>Description:</strong> {{ weather.weather[0].description | capitalize }}</p>

        <p><strong>Humidity:</strong> {{ weather.main.humidity }}%</p>

        <p><strong>Wind Speed:</strong> {{ weather.wind.speed }} m/s</p>

    {% else %}

        <p>Weather data is currently unavailable. Please try again later.</p>

    {% endif %}

</div>

{% endblock %}

APP.PY

from datetime import datetime

import json

import requests

from flask import Flask, render\_template, request, redirect, url\_for

from flask\_login import LoginManager, UserMixin, login\_user, login\_required, logout\_user, current\_user

app = Flask(\_\_name\_\_)

app.secret\_key = 'dlMwYFRPmi4Rrdfh5rVUuUdPv2e5Bckw'

login\_manager = LoginManager()

login\_manager.init\_app(app)

# User model

class User(UserMixin):

    def \_\_init\_\_(self, id):

        self.id = id

# Hardcoded user credentials (replace with real authentication in production)

users = {'admin': {'password': 'password123'}}

# User loader

@login\_manager.user\_loader

def load\_user(user\_id):

    return User(user\_id)

# Login route

@app.route("/login", methods=["GET", "POST"])

def login():

    if request.method == "POST":

        username = request.form['username']

        password = request.form['password']

        if username in users and users[username]['password'] == password:

            user = User(username)

            login\_user(user)

            return redirect(url\_for('home'))

        else:

            return render\_template("login.html", error="Invalid credentials")

    return render\_template("login.html")

# Home route (only accessible if logged in)

@app.route("/", methods=["GET", "POST"])

@login\_required

def home():

    if request.method == "POST":

        preferences = {

            "location\_type": request.form.get("location\_type").lower(),

            "budget": request.form.get("budget").lower(),

            "season": request.form.get("season").lower(),

            "activities": [activity.lower() for activity in request.form.getlist("activities")],

            "num\_days": int(request.form.get("num\_days"))

        }

        # Load test data from JSON

        with open('data/destination.json') as f:

            destinations = json.load(f)["destinations"]

        # Filter destinations based on user preferences and number of days

        recommendations = [

            d for d in destinations if

            d["type"].lower() == preferences["location\_type"] and

            d["budget"].lower() == preferences["budget"] and

            d["season"].lower() == preferences["season"] and

            any(activity.lower() in [a.lower() for a in d["activities"]] for activity in preferences["activities"]) and

            d["num\_days"] >= preferences["num\_days"]  # Check if destination's days are >= user's days

        ]

        # Only return the city names and the preferences

        if recommendations:

            city\_names = [d["name"] for d in recommendations]

            return render\_template("results.html", recommendations=city\_names, preferences=preferences)

        else:

            # If no matching destinations, inform the user

            return render\_template("results.html", recommendations=[], preferences=preferences, error="No destinations match your criteria for the number of days.")

    return render\_template("index.html")

# Logout route

@app.route("/logout")

@login\_required

def logout():

    logout\_user()

    return redirect(url\_for('login'))

# Weather API route with error handling

def get\_weather(city):

    api\_key = "0152a10cc622469182929c0f5d489c6d"

    url = f"http://api.openweathermap.org/data/2.5/weather?q={city}&appid={api\_key}&units=metric"

    try:

        response = requests.get(url)

        response.raise\_for\_status()  # Raise an exception for HTTP errors

        return response.json()

    except requests.RequestException:

        return {"error": "Could not retrieve weather data"}

@app.route("/weather")

@login\_required

def weather():

    city = request.args.get('city', 'London')

    weather\_data = get\_weather(city)

    return render\_template('weather.html', weather=weather\_data)

# Learning agent: Save feedback and improve recommendations

def load\_user\_data():

    try:

        with open('data/user\_data.json') as f:

            return json.load(f)

    except FileNotFoundError:

        return {}

def save\_user\_feedback(user\_id, feedback):

    user\_data = load\_user\_data()

    if user\_id not in user\_data:

        user\_data[user\_id] = {'feedback': []}

    user\_data[user\_id]['feedback'].append(feedback)

    with open('data/user\_data.json', 'w') as f:

        json.dump(user\_data, f)

def update\_recommendations(user\_id, destinations):

    user\_data = load\_user\_data()

    if user\_id not in user\_data:

        user\_data[user\_id] = {'feedback': []}

    # Simple feedback logic to improve recommendations

    feedback = user\_data[user\_id]['feedback']

    liked\_destinations = [f['destination'] for f in feedback if f['liked']]

    for dest in destinations:

        if dest["name"] in liked\_destinations:

            dest["score"] += 1

        else:

            dest["score"] = max(dest.get("score", 0) - 1, 0)

    destinations.sort(key=lambda x: x.get("score", 0), reverse=True)

    return destinations

# Feedback route to store and update recommendations

@app.route("/feedback/<destination>", methods=["POST"])

@login\_required

def feedback(destination):

    feedback = request.form['feedback']  # 'liked' or 'disliked'

    # Save user feedback

    user\_id = current\_user.id

    feedback\_data = {'destination': destination, 'liked': feedback == 'liked'}

    save\_user\_feedback(user\_id, feedback\_data)

    # Update recommendations based on feedback

    with open('data/destination.json') as f:

        destinations = json.load(f)["destinations"]

    updated\_recommendations = update\_recommendations(user\_id, destinations)

    return redirect(url\_for('home'))

if \_\_name\_\_ == "\_\_main\_\_":

    app.run(debug=True)

import json

@app.route('/update\_score', methods=['POST'])

def update\_score():

    # Assuming the destination is sent as part of the form data

    destination\_name = request.form.get('destination\_name')

    action = request.form.get('action')  # "like" or "dislike"

    # Load the current data from the destination.json file

    with open('destination.json', 'r') as file:

        data = json.load(file)

    # Update the score based on action

    for destination in data['destinations']:

        if destination['name'] == destination\_name:

            if action == 'like':

                destination['score'] += 1

            elif action == 'dislike':

                destination['score'] -= 1

    # Write the updated data back to the destination.json file

    with open('destination.json', 'w') as file:

        json.dump(data, file, indent=4)

    return redirect(url\_for('results'))