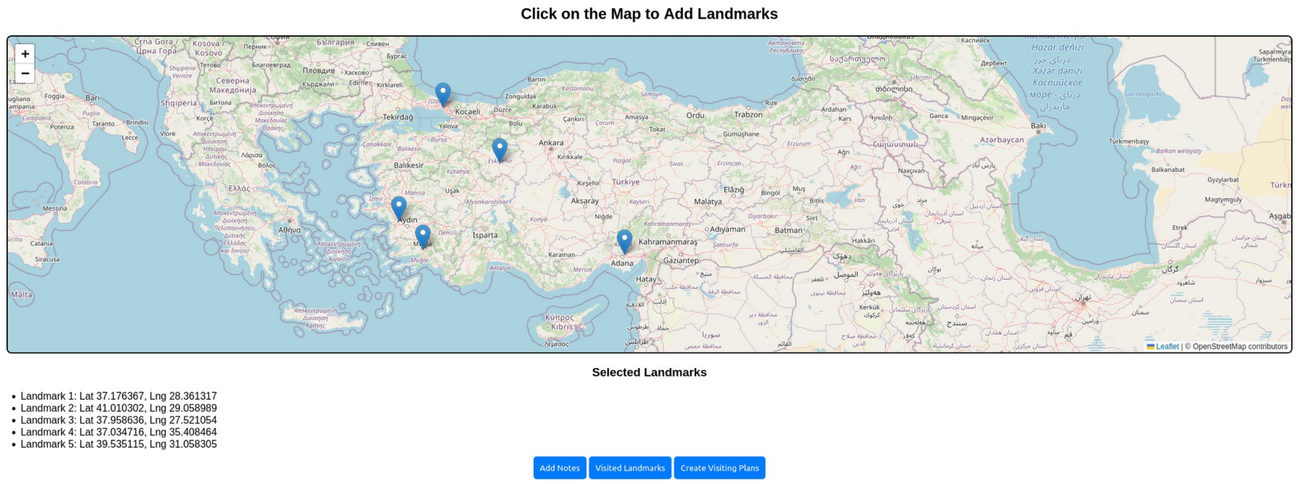
CENG-3502 : Dynamic Web Programming

Midterm Assignment: Back-End Development for Landmarks App

Objective: Create a back-end system for an HTML page that allows users to store landmarks and track visited landmark history.



Click on the Map to Add Landmarks

Selected Landmarks

- Landmark 1: Lat 37.176367, Lng 28.361317
- Landmark 2: Lat 41.010302, Lng 29.058989
- Landmark 3: Lat 37.958636, Lng 27.521054
- Landmark 4: Lat 37.034716, Lng 35.408464
- Landmark 5: Lat 39.535115, Lng 31.058305

Figure 1: Screenshot of the web page.

Description:

You will develop a back-end service along with the necessary front-end components to support the project. The application will allow users to select places on a map (as shown in Figure 1). There will be three main buttons:

1. **Add Notes:**
   Users can add notes for selected locations. These notes should be stored in the back-end.
2. **Visited Landmarks:**
   This option will retrieve and display previously visited places along with any associated notes.
3. **Create Visiting Plan:**
   Users can select multiple places and add notes for each. This will help plan future visits.

You are responsible for implementing full CRUD (Create, Read, Update, Delete) functionality in the back-end for managing the notes and place data. Please also ensure the front-end has all the required components to interact with the back-end accordingly.

Instructions:

1. **Set Up the Project:**
    - Choose a back-end framework (Node.js, Express etc.)
    - Set up a database (You can use local storage or MongoDB, PostgreSQL, MySQL).
2. **Define the Data Structure:**
    - A `Landmark` should include:
        - `id` (unique identifier)
        - `name`
        - `location` (latitude and longitude)
        - `description`
        - `category` (e.g., historical, natural, cultural)
    - A `VisitedLandmark` should include:
        - `id`
        - `landmark_id`
        - `visited_date`
        - `visitor_name`
3. **Build API Endpoints:**
    - **Landmarks:**
        - `POST /landmarks` → Add a new landmark.
        - `GET /landmarks` → Retrieve all landmarks.
        - `GET /landmarks/{id}` → Retrieve a specific landmark.
        - `PUT /landmarks/{id}` → Update a landmark.
        - `DELETE /landmarks/{id}` → Remove a landmark.
    - **Visited Landmarks:**
        - `POST /visited` → Record a visited landmark.
        - `GET /visited` → Retrieve visit history.
        - `GET /visited/{id}` → Retrieve visit history for a specific landmark.
4. **Implement Front-End Integration:**
    - Modify the given HTML page to interact with your back-end using JavaScript (AJAX or Fetch API).
    - Provide forms for adding landmarks and marking them as visited.

Bonus Tasks:

- Implement user authentication (JWT, OAuth, or sessions).
- Add search and filter functionality for landmarks.

- Deploy the application to a cloud service (Heroku, Vercel, or AWS).

Submission Guidelines:

- Upload your project to GitHub and sharWe the repository link.
- Include a README file with setup instructions.
- Submit a short report explaining your implementation.
-

Deadline: 27th April 2025

Good luck and happy coding!