Full Stack Web Development Challenge

Task 1: Create a webpage that displays weather information for a user-specified city using HTML, CSS, and Bootstrap. The page should include:

- A navigation bar with a “Weather App” title.
- A form with an input field for the city name and a “Get Weather” button.
- An area to display the weather information, including temperature, weather conditions, and an icon representing the weather.

Task 2: Implement JavaScript functionality to handle the following:

- When the “Get Weather” button is clicked, make an API call to a weather API (e.g., OpenWeatherMap) using JavaScript’s `fetch` function.
- Parse the API response to extract the temperature, weather conditions, and weather icon.
- Update the webpage with the retrieved information dynamically.

Task 3: Enhance the webpage with Local Storage and Cookies features:

- Save the last searched city in both local storage and a cookie whenever a user searches for weather.
- When the page loads, check for the presence of the saved city in local storage or the cookie and display the weather information for that city if available.

Task 4: Incorporate a SQL Database aspect:

- Create a simple SQLite database schema for storing user information, including name, email, and saved cities.
- Use JavaScript to handle database operations like adding new users and saving cities associated with each user.
-

Time duration (3.5 hours)

Task 1: Create Weather Webpage (HTML, CSS, Bootstrap)

- Estimated Time: 1 hour

Task 2: Implement Weather API Call (JavaScript)

- Estimated Time: 1 hour

Task 3: Local Storage and Cookies (JavaScript)

- Estimated Time: 30 minutes

Task 4: SQL Database Operations (SQL, JavaScript)

- Estimated Time: 1 hour

Coding Test Instructions

Test Overview:
This coding test is designed to assess your skills in various aspects of web development, including HTML, CSS, Bootstrap, JavaScript, Local Storage, Cookies, SQL Database, Web APIs, and problem-solving abilities. You are required to complete a series of tasks that cover these topics.

Test Duration:
The test is expected to take around 3 to 4 hours to complete. However, feel free to work at your own pace within the given time frame.

Environment:

- You will be taking the test remotely on your personal computer.
- Use your preferred code editor and web browser for development.
- You are required to share your screen using a live screen sharing tool (e.g., Zoom, Microsoft Teams, Google Meet) to ensure transparency and prevent cheating.

Instructions:

1. You are not allowed to use any AI tools, automated code generators, or any form of assistance from AI applications.
2. Do not directly copy or use full problem solutions from the internet. It’s acceptable to use documentation, references, and online resources for learning, but the code you write must be your own.
3. For each task, follow the provided instructions and complete the task according to the specified requirements.
4. Organize your code neatly and use appropriate variable names and comments for clarity.
5. Submit your solutions in a zip file, each task in a separate folder, including all necessary files and a brief README explaining your approach to solving the task.

Submission:

1. Once you have completed all the tasks or reached the end of the allotted time, compress the entire project folder into a zip file.
2. Rename the zip file with your name (e.g., “John_Doe_Coding_Test.zip”).
3. Upload the zip file to the designated submission link or email it to the provided email address.

Important:

- Plagiarism and cheating are strictly prohibited and will result in immediate disqualification.
- The purpose of this test is to assess your skills and knowledge. Please refrain from seeking external help during the test.

Good luck! If you have any questions or need assistance, don’t hesitate to ask the proctor.