Full-stack Developer Test Instructions

1. SCSS Compilation & Documentation

- This website was created years ago. Your first task is to attempt to compile SCSS into CSS.
- Document your process in a readme file, including:
    - Any issues you encounter.
    - How you resolved them (if applicable).
- If you **successfully compile SCSS**, continue using the SCSS files for all front-end development.
- If you **cannot get SCSS to work**, create a new CSS file and use that for any front-end development.

2. User Registration & Authentication

- Using the existing website as a base, create a new page: `/register`.
- The registration form should include the following fields:
    - **Name** (required)
    - **Email** (required)
    - **Phone** (optional)
- When the user submits the form:
    - Store the record in a **database**.
    - Prompt the user to create a **password**.
    - Save the password securely in the database (use proper hashing).
    - Redirect the user to `/log-in` after successful registration.

3. User Login

- Create a `/log-in` page where users can log in using their **email** and **password**.
- Authenticate users against the stored credentials.
- If authentication is successful, grant access to **protected pages**.
- If authentication fails, display an appropriate error message.

4. Members-Only Page

- Create a `/members` page that is **only accessible to logged-in users**.
- This page should display a table of all registered users, including:
    - Name
    - Email
    - Phone
    - Actions: **Edit / Delete**

5. Editing & Deleting Users

- On the `/members` page:
    - Allow users to **edit member details**.
    - Enable users to **reset another member's password**.
    - Implement a **delete function** to remove members from the database.

6. Error Handling & User Feedback

- Implement **messaging** for various actions, including:
    - `"Error: The Email field is required"`
    - `"Your registration has been successful"`
    - `"Invalid login credentials"`
    - `"Member successfully updated/deleted"`

7. Approach & Reasoning

- Write a **brief description** explaining:
    - Your reasoning behind key decisions
    - Your review on the existing code

Submission Guidelines

- **Code Submission**: Provide a **GitHub repository** or a **ZIP file** with your project.
- **Database Setup**: Include a SQL script or migration files for setting up the database.
- **Brief Documentation**: Explain how to set up and run the project locally (this can be a readme).
- **Approach Description**: Provide a short write-up on your reasoning behind key decisions in the readme.