CausalFunnel

Coding Assignment: **Build a Simple Quiz Application**

Objective

Thank you for showing interest in the Software Engineer Intern position at CausalFunnel. As a next step, to evaluate your skills in software development, we would like you to create a quiz application that meets the criteria detailed below. This will help us understand your technical skills,attention to detail, and the quality of your work. You can build it using any tech stack like, Reactjs, Angular, Next.js, Vue.js, php etc

Requirements:

1. **Quiz Layout & Flow**:

- The application should have a start page where the user should submit their email address.
- Then the application should display 15 questions to the user.
- A timer should be displayed on the top of the page, counting down from 30 minutes. The quiz should auto-submit when the timer reaches zero.

2. **Navigation**:

- Users should be able to navigate to a specific question.
- An overview panel or similar element should show all questions indicating:
- Which questions the user has visited.
- Which questions have been attempted.

3. **End of Quiz**:

- After the quiz or when the timer ends, users should be directed to a report page.
- This report should display each question with the user's answer and the correct answer side by side or in a format that is easy to compare.

4. **Data Source**:

- Fetch the quiz questions from https://opentdb.com/api.php?amount=15 API.
- From this API use the question parameter as the question to be displayed.
- Choices to be shown to the user for each question is a concatenated array of correct_answer and incorrect_answers parameters.
- Correct answer for every question is provided in correct_answer parameter.

Judgment Criteria:

1.**Functionalities**: Ensure all the requirements above are implemented and working correctly.
2. **Bug-free Code**: Your application should run without any errors or unexpected behavior.
3. **Code Quality**: The code should be clean, well-organized, and follow best practices such as reusability. Comments should be used judiciously to explain complex or non-intuitive parts of your code.

Submission:

1. Host the application on a platform of your choice (e.g., Netlify, Vercel, GitHub Pages).
2. Share the repository link containing your source code.
3. Include a README in your repository detailing:

- Brief overview of your application which may include your approach to the problem and information regarding the components that you build.
- Any setup or installation instructions.
- Assumptions made, if any.
- Challenges faced and how you overcame them.

Bonus (Optional):

- The application should adapt to different device sizes and be compatible with the latest versions of major browsers (e.g., Chrome, Firefox, Safari, and Edge).
- Add smooth transitions or animations when navigating between questions.
- Implement any additional features or improvements that you believe would enhance the user experience.