Take-Home Challenge Instructions

General Guidelines

You are provided with a set of user needs representing different small applications. Your task is to **choose one** of these user needs and develop a web application that addresses it. The goal is to interpret these needs and translate them into a functional and user-friendly application.

This challenge is designed to assess your ability to:

- Understand and interpret user requirements.
- Design a simple, effective solution.
- Implement the solution using the specified technology stack.
- Demonstrate your coding practices and problem-solving skills.

Technology Stack

- **Frameworks**: Use either **React** or **Svelte** for building your application.
- **Styling**: You are free to use any CSS methodology or framework, such as **Tailwind CSS**, plain CSS, CSS Modules, or Styled Components.
- **Data Fetching**: Utilize public APIs that require no authentication or registration.
- **State Management**: Use the state management tools provided by your chosen framework (e.g., React's built-in state management or Svelte's stores).

Submission Requirements

- **Level 1 (50%):**
    - **Working with AI:** Please share the link to chatGPT work so that we can understand how you are approaching the problem. Even if you have not fully solved the problem, this itself is worth some points.
- **Level 2 (30 %)**
    - **Working application**: Deploy your application on a free hosting service like **CodeSandbox**, or **StackBlitz**.
- **Level 3 (20%)**
    - **Code sharing:** Share your code with appropriate notes and README
- **Deadline**: Submit your application within **one week**.

User Needs (Choose One)

1. Book Finder

User Persona:

- **Name**: Alex
- **Occupation**: College Student
- **Need**: Alex wants to search books (use your imagination of all the ways Alex would like to do it).

Task:

- Build an application for Alex. Do as good a job you can.

API to Use: **Open Library Search API**: example (know that it gives a lot of details, including the cover)::
`https://openlibrary.org/search.json?title={bookTitle}`

---

2. Weather Now

User Persona:

- **Name**: Jamie
- **Occupation**: Outdoor Enthusiast
- **Need**: Jamie wants to check the current weather conditions quickly for any city.

Task:

- Build an application for Jamie.

API to Use:

- **Open-Meteo API**.

---

3. Recipe Ideas

User Persona:

- **Name**: Taylor
- **Occupation**: Busy Professional
- **Need**: Taylor wants to help in the kitchen, when he comes home. Maybe he wants to cook based on what ingredients he has. Or, what he is in the mood for. Maybe depending on the time he has for cooking. Or, maybe what he wants or not wants. Use your imagination.

Task:

- Build an application for Taylor.

API to Use:

- **TheMealDB API**: `https://www.themealdb.com/api/json/v1/1/filter.php?i={ingredient}`
  -
    - Replace `{ingredient}` with the user's input.

---

4. Earthquake Visualizer

User Persona:

- **Name**: Casey
- **Occupation**: Geography Student
- **Need**: Casey wants to visualize recent earthquake activity around the world to understand seismic patterns.

Task:

- Build an application, with visualization for her..

API to Use:

- **USGS Earthquake API** `https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/all_day.geojson`
  -

Additional Requirements:

- Use a mapping library to display the map:
  - **Leaflet** is recommended and has integrations for both React and Svelte:
    - **React**: react-leaflet
    - **Svelte**: svelte-leaflet
- Ensure the map is interactive and user-friendly.

Additional Notes

- **Feature Selection**: Focus on the core functionality described. Additional features are welcome but not required.

- **User Interface (UI) and User Experience (UX)**: Aim for a clean and user-friendly design.
- **Responsiveness**: Ensure your application works well on both desktop and mobile devices.
- **Error Handling**: Handle errors gracefully, such as displaying a message if no results are found or if there's a network error.
- **Code Quality**: Write clean, readable code with appropriate comments.
- **Testing**: Test your application to ensure it works as intended.

We look forward to seeing how you interpret the user needs and translate them into a functional application. This is an opportunity to showcase your skills in understanding requirements, designing solutions, and implementing them effectively.

Good Luck!