raftlabs

Sr Full Stack Developer- Assessment

Project Overview:

You will build an **Order Management** feature for a food delivery app. The goal is to create a simple, functional part of the app that allows users to place orders, view the menu, and track the order status.

Feature Requirements:

1. **Menu Display:**
   - A list of food items (e.g., pizza, burgers, etc.) displayed on the UI.
   - Each item has a name, description, price, and image.

2. **Order Placement:**
   - Users can add items to their cart.
   - Users can specify the quantity of each item in the cart.
   - Users can proceed to checkout, where they enter their delivery details (name, address, phone number).

3. **Order Status:**
   - Once the order is placed, show the status (e.g., “Order Received”, “Preparing”, “Out for Delivery”).
   - Real-time updates of the status (this can be simulated in the back-end).

4. **Back-End:**

- REST API to handle order placement, menu retrieval, and updating order status.
- Store order details and menu items in-memory or a simple database.

5. **Test-Driven Development (TDD):**
    - Write tests for the API endpoints and key UI components.
    - Tests should cover CRUD operations for orders, input validation, and order status updates.

6. **User Interface:**
    - A simple UI that allows users to interact with the menu, add items to the cart, and place orders.
    - Use React with Vite or Next.js for the front-end.

7. **Real-Time Updates (Optional but Encouraged):**
    - Implement a way to simulate real-time updates of the order status.

Deliverables:

1. **Code Repository:**
    - A public GitHub repository with the full code for the order management feature.
    - The code should be well-structured, with a focus on readability and maintainability.

2. **Hosted Application:**

    Host the app on a platform like Vercel or Netlify.

3. **Loom Video:**
    - A short video (12-15 minutes) explaining your process.

        - Walk through your code and explain how you structured it.
        - Discuss the architecture and design choices.
        - Explain how you used AI tools during development (e.g., for code generation, testing, etc.).
        - Mention any challenges you faced and how you solved them.

Evaluation Criteria:

1. **Problem-Solving Approach:**
    - How do you break down the requirements into smaller tasks?
    - What steps did you take to ensure scalability and maintainability in the feature?

2. **Code Quality:**
    - Is your code clean and maintainable?
    - Are the tests thorough and well-written?

3. **UI/UX:**
    - Is the UI simple but functional?
    - Is the user experience intuitive and smooth?

4. **Back-End:**
    - Are your API endpoints well-structured and secure?
    - Do you handle edge cases and input validation?

5. **Use of AI:**
    - How effectively did you use AI tools in your development? Did you leverage AI for code generation, testing, debugging, etc.?