

Job Description: React Native Developer

Overview:

We are looking for a talented React Native Developer to join our dynamic team. As part of the application process, candidates will be required to complete a small assignment that involves developing a React Native mobile application and integrating it with a backend system. This assignment will help us assess your skills and fit for our team.

Objective:

Develop a simple chat application using React Native that integrates with the provided backend API for real-time messaging.

Requirements:

1. API Integration:

You will be provided with a custom API containing all the necessary endpoints for interacting with the backend chat service.

Your task is to integrate this API into your React Native application, using them to manage user registration, room creation, and message handling.

2. Core Features to implement:

- **Set Username Screen:**
Allow the user to set their username through the provided API.
- **Rooms List Screen:**
Display a list of available chat rooms. Users can choose to join an existing room or create a new one.
- **Create Room Screen:**
Allow users to create a new room.
- **Chat Screen:**
After selecting or creating a room, users should be able to see the previous 10 messages in that room and send new messages.
 - Messages should appear in real-time using a WebSocket connection.
 - A text input should allow the user to send messages, and they should be sent instantly to other users in the room.

3. WebSocket Integration:

- **WebSocket Connection:**
Set up a WebSocket connection to handle real-time communication in the chat rooms using our API.

- **Real Time Communication:**

Users should be able to send messages through the WebSocket connection, and new messages should appear instantly for all users in the room.

4. Error Handling & Edge Cases:

- Ensure that the app handles API errors gracefully, such as failed requests or issues with the WebSocket connection.
- Handle edge cases like no rooms available or network errors.

5. User Flow:

- **Login Screen:**

- The user enters their username.
- A unique user ID is assigned upon signup.

- **Choose Room:**

- The user can either create a new room or join an existing room.
- Create Room: A unique room ID is generated for the room.
- Join Room: The user selects a room, and the WebSocket connection is established. The last 10 messages of the room are loaded.

- **Real-time Messaging:**

- When the user sends a message, it is sent via WebSocket. After the message is sent successfully an event is received.
- A unique message ID is generated for each new message.

- **Join/Leave Room:**

- When the user joins or leaves a room, an event is broadcasted to all the connected clients to that room.
- Your application should show the same i.e.,
`<username> has joined/left the room`

Preferred Skills:

- Experience with cloud services (e.g., AWS, Google Cloud).
- Knowledge of databases (e.g., PostgreSQL, MongoDB).
- Familiarity with version control systems (e.g., GitHub).

Evaluation Criteria:

Code quality, API/WebSocket integration, UI design, and app accessibility.

Submission Guidelines:

1. GitHub Repository:

- Create a GitHub repository containing all relevant code for your React Native app.
- Ensure your repository includes a comprehensive README file with setup instructions.

2. APK File:

- Compile the application and provide an APK file.
- Make sure all the API's and WebSocket work properly in your application.

3. Video Explanation: (Mandatory)

- Record a short video explaining the key features of your app.
- Upload the video to YouTube or Loom and include the link in your submission.

Resources:

- **API Endpoint:** <https://chat-api-k4vi.onrender.com/>
- **Documentation:**
 - **Swagger UI:** <https://chat-api-k4vi.onrender.com/docs>
 - **Redocly UI:** <https://chat-api-k4vi.onrender.com/redoc>

WebSocket Overview:

- **Join Room:** `ws://chat-api-k4vi.onrender.com/ws/{roomId}/{username}`

- **Message Payload Object:**

```
{  
  "event": "message",  
  "content": "your-message-content"  
}
```