

## Technical Test: React Native Developer Position

Dear Candidate,

Thank you for participating in the technical test. This test aims to evaluate your skills and proficiency in React Native development and related technologies. We appreciate your time and effort in completing this assessment.

### Instructions:

- Please read through each task carefully before proceeding.
- Complete each task to the best of your ability, adhering to any provided guidelines or requirements.
- You may use any necessary resources, such as documentation or online references, to aid in completing the tasks.
- Ensure that your code is well-commented and organized for clarity.
- Submit your completed test via email to [hr@bazaraf.com](mailto:hr@bazaraf.com) by **20<sup>th</sup> of August 2024**.
- If you encounter any technical issues or have questions regarding the tasks, feel free to reach out for assistance.

**Note:** This test is designed to assess your technical skills and problem-solving abilities. There is no strict time limit, but we recommend managing your time effectively to complete all tasks within a reasonable timeframe.

### TASK: Recipe Finder

#### Objective:

Develop a Recipe Finder react native app with Expo or React Native CLI, where users can search for recipes, view recipe details, save their favorite recipes, and filter recipes by different categories (e.g., cuisine, meal type). The application will also fetch recipes from an external API.

#### Requirements:

##### 1. Recipe Search Component:

- A search bar where users can type in ingredients or recipe names.
- On submitting the search, display a list of matching recipes.
- Integrate with a public API like the Spoonacular API (or use a mock API if needed).

## **2. Recipe List Component:**

- Display a list of recipes based on the search or filter criteria, with each recipe showing an image, title, and a short description.
- Include pagination or infinite scroll to handle large datasets.
- Each recipe should have a "Save to Favorites" button.

## **3. Recipe Details Component:**

- Clicking on a recipe should navigate to a detailed view of the recipe.
- The detailed view should include the recipe title, image, ingredients list, and step-by-step instructions.
- Include a button to go back to the search results.

## **4. Favorites Component:**

- A separate section where users can view recipes they've saved as favorites.
- Allow users to remove recipes from their favorites list.

## **5. Filter Component:**

- Filters to narrow down recipes by categories such as cuisine type (e.g., Italian, Chinese), meal type (e.g., breakfast, lunch, dinner), and dietary preferences (e.g., vegetarian, vegan).
- The filter should update the recipe list dynamically based on selected criteria.

## **6. API Integration:**

- On initial load, fetch a list of popular recipes from an API.
- Handle API loading states, errors, and display appropriate messages if no results are found.

## **7. State Management:**

- Use React's built-in state management (useState, useEffect) for managing the application's state.
- Consider using useContext or a state management library (like Redux) if needed for managing global state, particularly for favorites.

## **8. Routing:**

- Use React Router to manage navigation between the search results, recipe details, and favorites sections.
- Ensure that the URL reflects the current view, allowing users to bookmark or share links to specific recipes.

## **9. Styling:**

- Implement a responsive and visually appealing UI with clean CSS. You can allow the use of CSS frameworks like Tailwind CSS or Material-UI.
- Ensure that the application is mobile-friendly.

## **Testing:**

- Write unit tests for key components using Jest and React Testing Library.
- Ensure that tests cover search functionality, API data fetching, state management, and favorites handling.

## **Bonus Features (Optional):**

- Implement a feature to allow users to rate recipes or leave comments.
- Add animations to enhance the user experience (e.g., transitions when switching between views).
- Allow users to create and manage their own recipes in addition to searching for existing ones.

## **Evaluation Criteria:**

- Code Quality: Clean, maintainable, and modular code with good comments.
- State Management: Effective use of React state and context, with consideration for scalability.
- API Handling: Proper handling of API requests, responses, loading states, and errors.
- UI/UX: A polished, user-friendly, and responsive interface.

- Routing: Seamless and intuitive navigation using React Router.
- Testing: Well-written and comprehensive unit tests that cover critical features.

**Bonus:**

Implementation of optional features demonstrates a higher level of skill and creativity.

**Note:** *Consideration will be given to both functionality, code quality, and the inclusion of comprehensive tests.*

**Conclusion:**

Thank you for completing the technical test for the React Native Developer position. We appreciate your dedication and effort in demonstrating your skills and capabilities. Your submission will be carefully reviewed by our hiring team, and we will be in touch regarding the next steps in the hiring process.

If you have any further questions or require additional information, please don't hesitate to contact us.

Best regards,

**Khalid Fareed**

**HR Representative**

**BazarAf**

[Hr@bazaraf.com](mailto:Hr@bazaraf.com)