

Final Practical Evaluation Test – React Frontend Development

Project: Recipe Explorer App

Objective:

The goal of this final practical evaluation is to assess students' ability to create a **React-based application** that fetches data from an API, displays a list of items, implements routing for detail pages, and ensures a user-friendly interface. Students will demonstrate their proficiency in:

- ☒ Fetching data from an API using `fetch`
 - ☒ Managing state using `useState` and `useEffect`
 - ☒ Implementing **React Router** for navigation between pages
 - ☒ Displaying a **list of recipes** and navigating to a **detailed recipe view**
 - ☒ Applying **Bootstrap for styling**
 - ☒ Ensuring **code structure, naming convention and readability**
-

Problem Statement

You are required to **build a React application** that displays a list of recipes by fetching data from <https://dummyjson.com/recipes> and allows users to view the details of a selected recipe.

Project Requirements

1 Setup the Project

1. Create a new React project using Vite or Create React App:

```
npm create vite@latest recipe-explorer  
cd recipe-explorer  
npm install
```

2. Install necessary dependencies:

```
npm install react-router-dom bootstrap
```

Features to Implement

◇ 1. Home Page (`RecipeList.js`) – List View

- Fetch recipes from the API:
 - ◇ **API Endpoint:**

```
GET https://dummyjson.com/recipes
```

- Display the list of recipes in a **Bootstrap Card format**.
- Show **Recipe Name**, **Cuisine**, **Prep Time**, and **Thumbnail Image**.
- Each card should have a **View Details** button that navigates to the **Recipe Detail Page** (`/recipe/:id`).

◇ 2. Recipe Detail Page (`RecipeDetails.js`)

- Fetch the selected recipe details based on the `id` from the route.
- Display complete details including:
 - **Recipe Name**
 - **Cuisine**
 - **Ingredients** (list format)
 - **Instructions** (paragraph format)
 - **Cooking Time**
 - **Servings**
 - **Recipe Image**

◇ API Endpoint:

```
GET https://dummyjson.com/recipes/:id
```

◇ 3. Navigation & Routing

- Use `react-router-dom` to configure routing:
 - `/` → Shows the Recipe List
 - `/recipe/:id` → Shows the Recipe Details
- Implement a **Back to Home** button in the detail view to navigate back to the list.

UI & Functional Requirements

- ✓ **Use Bootstrap for styling** (cards, buttons, images, layout).
- ✓ **Handle API errors** gracefully and display a message if data cannot be fetched.

 **Good Luck for the exam!** 😊