# **Recipe Finder**

## **Objective**

Develop a Recipe Finder application using React, Tailwind CSS, or Material-UI for styling. The application will fetch data from a provided recipe API and display it interactively with filtering and search options.

## **Requirements**

### **Frontend**

1. **Library/Framework**: React.
2. **Styling**: Tailwind CSS or Material-UI for a responsive and interactive design.
3. **Functionalities**:
   1. **Homepage**:
      1. Display a list of recipes with their names, images, cooking time, and difficulty levels.
      2. Include a search bar to filter recipes by name or ingredients.
      3. Pagination to manage the list of recipes.
   2. **Recipe Detail Page**:
      1. Display full recipe details, including name, image, ingredients, steps, cooking time, and difficulty level.
      2. Provide a button to mark a recipe as a favorite.
   3. **Favorite Recipes Page**:
      1. List all recipes marked as favorites.
      2. Allow users to remove recipes from the favorites list.
   4. **Responsive Design**:
      1. Ensure the app is mobile-friendly.

### **Backend API (Pre-provided)**

You do not need to create a backend. Instead, they will consume a pre-provided API such as the <https://www.postman.com/spoonacular-api/spoonacular-api/collection/rqqne3j/spoonacular-api>

#### **API Endpoints:**

* **Get Recipes**: GET /recipes - Fetch a list of recipes.
* **Search Recipes**: GET /recipes/search - Search for recipes by name or ingredients.
* **Get Recipe Details**: GET /recipes/:id - Fetch detailed information about a specific recipe.

Provide sample API responses or mock data if the actual API is not accessible.

### **Optional Advanced Features**

1. **Sorting**:
   1. Add sorting options (e.g., by cooking time or difficulty level).
2. **Advanced Search**:
   1. Enable multi-ingredient search.
3. **Animations**:
   1. Add subtle animations for transitions and interactions.

## **Deliverables**

* **Frontend**:
  + A responsive React application with the required pages (home, recipe detail, and favorites).
* **API Integration**:
  + Fetch and display data accurately using the provided API.
* **Deployment**:
  + Hosted application on a platform like Vercel or Netlify.
* **Code**:
  + Share the GitHub repository with clear instructions for setup and usage in the README file.