# **CookBook: Your Virtual Kitchen Assistant**

## **Project Documentation**

### **Frontend Development with React.js**

CookBook is built using React.js, a popular JavaScript library for building user interfaces. React.js was chosen for its component-based architecture, efficient state management, and seamless routing capabilities, which enhance the overall user experience.

### **Introduction**

#### **Project Name: CookBook - Your Personal Culinary Guide**

### **Team Members and their E-Mail ID’s :**

* Riswana .S - stu\_riswana\_s@cttewc.edu.in
* Ashika . M - stu\_ashika\_m@cttewc.edu.in
* Messina .V - stu\_messina\_v@cttewc.edu.in
* Mercy .B - stu\_mercy\_b5@cttewc.edu.in

### **Project Overview**

CookBook is a digital cooking assistant designed for both beginners and experts. It provides an intuitive platform for discovering, saving, and creating recipes while ensuring a seamless user experience.

**Features:**

* Global Recipe Access: Browse an extensive recipe collection from the MealsDB API.
* Visual Browsing: Explore dishes through well-organized image galleries.
* User-Friendly Interface: Navigate easily with a clean and structured design.
* Smart Search: Locate recipes quickly with dynamic filtering options.

### **Architecture & Component Breakdown**

#### **Component Structure:**

* **Hero Section:** An introduction with an explore button.
* **Popular Categories:** Displays recipe categories.
* **Trending Dishes:** Highlights the most popular recipes.
* **Newsletter Section:** Users can subscribe for updates.
* **Category Page:** Displays recipes categorized for convenience.
* **Recipe Details Page:** Provides detailed instructions, ingredient lists, and video tutorials.

#### **State Management:**

* React's useState Hook: Manages component-level states.
* React's useEffect Hook: Fetches API data and handles side effects.

#### **Routing:**

* Implemented using React Router Dom for smooth navigation between pages.

### **Setup Guide**

#### **Prerequisites:**

* Node.js and npm: Install from the official Node.js site.
* React.js: Create a project using npx create-react-app cookbook-app.
* React Router Dom: Enables smooth page navigation.
* Axios: Manages API requests.
* Bootstrap/Tailwind CSS: Enhances design and layout.

#### **Installation Steps:**

1. **Clone the repository**:

[[Project Code]](https://drive.google.com/drive/folders/1u8PnV_mE0mwKkH_CvuNpliZtRLJZMqrO?usp=sharing)

2**. Navigate into the project directory:**

cd recipe-app-react

3. **Install dependencies**:

npm install

4. **Run the development server:**

npm start

5. **Open in Browser**: Go to http://localhost:3000

### **Folder Structure**

* **components/** → Contains reusable UI elements (Navbar, Hero, Categories, etc.).
* **pages/** → Holds page-specific components (Home, Category, Recipe).
* **styles/** → Stores CSS files for styling.

### **Running the Application**

**To start the application, run:**

npm start

### **Component Documentation**

#### **Main Components:**

* **Hero Section** → Introduces the application with a call-to-action.
* **Popular Categories** → Displays various recipe categories.
* **Trending Dishes** → Highlights top-rated recipes.
* **Recipe Page** → Provides full recipe details and instructions.

#### **Reusable Components:**

* **Navbar** → Handles site navigation.
* **Footer** → Displays additional resources and contact information.

### **State Management Approach**

* No Global State Management (Redux is not used).
* Each Component Maintains Its Own State using useState & useEffect.

### **User Interface Overview**

* **Homepage:** Features trending recipes, category listings, and a search bar.
* **Category Page:** Displays all recipes within a selected category.
* **Recipe Page:** Provides recipe details, ingredient lists, and an instructional video.

### **Styling & UI**

* Bootstrap & Tailwind CSS: Used for responsive and consistent design.
* Custom CSS: Applied for additional styling where needed.

### **Testing & Validation**

#### **Testing Strategy:**

* **Manual UI Testing:** Ensures responsiveness and smooth navigation.
* **API Data Testing:** Validates data retrieval from MealsDB.
* **Component Testing:** Uses React Testing Library for key UI elements.

### **Demo**

Demo Link: [[View Demo Link]](https://drive.google.com/file/d/1T3M89LZ6r6pH0fxHHVJGlqZPSXw3tDRg/view?usp=drivesdk)

### **Known Issues & Future Plans**

#### **Current Issues:**

* API responses may experience occasional delays.
* UI may experience slight lag during high API usage.

#### **Planned Enhancements:**

* User Authentication: Enable users to save their favorite recipes.
* Redux Integration: Implement global state management for better efficiency.
* UI Improvements: Introduce enhanced animations and visual transitions.