# **Frontend development with React.js**

## **Project Documentation**

**Introduction**

**Project Title** : CookBook – Your Kitchen Virtual Assistant  
  
CookBook is a modern web application designed to simplify, organize, and enhance the culinary journey of its users. It acts as a virtual assistant for cooking enthusiasts and professional chefs alike, providing an intuitive and interactive platform for recipe discovery, organization, and personalized guidance.

**Team Members :**

Team ID : NM2025TMID41074

Team Size : 4

Team Leader : CHINNARASU E , Mail id : chinnarasu.official@gmail.com

Team member : BALA K , Mail id : balakamaraj1028@gmail.com

Team member : CHANDRU M , Mail id : chandrumeganathan@gmail.com

Team member : DHINAKARAN V , Mail id : vasudhinakaran@gmail.com

**Project Overview**

**Purpose:**  
The primary purpose of CookBook is to create a user-friendly platform where users can search, save, and manage recipes, while also gaining inspiration from curated content. The project demonstrates the application of modern frontend technologies to build an engaging, responsive, and efficient web interface.  
  
**Goals:**

→ To provide a seamless and intuitive recipe management experience.  
→ To demonstrate effective use of React.js for building scalable frontend applications.  
→ To establish a foundation for integrating advanced features such as personalization and user community engagement.  
  
**Features:**  
→ Recipe discovery with search functionality.  
→ Categorization of recipes by cuisine, meal type, or ingredients.  
→ Recipe detail pages with step-by-step instructions.  
→ User-friendly navigation and responsive UI design.  
→ Interactive and dynamic components built with React.js.

**Architecture**

**Component Structure:**  
→ App Component: Root component managing overall structure and routing.  
→ Navbar Component: Provides easy navigation across sections.  
→ Home Component: Showcases featured recipes and entry point to browsing.  
→ RecipeList Component: Displays a collection of recipes dynamically.  
→ RecipeDetail Component: Presents in-depth recipe instructions.  
→ Search Component: Handles keyword-based recipe search.  
→ Footer Component: Concludes the user interface with links and credits.

**State Management:**  
Local state is primarily managed using React ’s use State and use Effect hooks. For recipe data and search functionality, props are passed between components to maintain consistency.  
  
**Routing:**React Router is used for navigation between different pages such as Home, Recipe List, and Recipe Details. This ensures smooth client-side transitions without page reloads.

**Setup Instructions**

**Prerequisites:**

Here are the key prerequisites for developing a frontend application using React.js:

* **Node.js and npm**:

Node.js is a powerful JavaScript runtime environment that allows you to run JavaScript code on the local environment. It provides a scalable and efficient platform for building network applications.

Install Node.js and npm on your development machine, as they are required to run JavaScript on the server-side.

* Download: <https://nodejs.org/en/download/>
* Installation instructions: <https://nodejs.org/en/download/package-manager/>

* **React.js**:

React.js is a popular JavaScript library for building user interfaces. It enables developers to create interactive and reusable UI components, making it easier to build dynamic and responsive web applications.

Install React.js, a JavaScript library for building user interfaces.

* Create a new React app:

npx create-react-app my-react-app

Replace my-react-app with your preferred project name.

* Navigate to the project directory:

cd my-react-app

* Running the React App:

With the React app created, you can now start the development server and see your React application in action.

* Start the development server:

npm start

This command launches the development server, and you can access your React app at [http://localhost:3000](about:blank) in your web browser.

* **HTML, CSS, and JavaScript**: Basic knowledge of HTML for creating the structure of your app, CSS for styling, and JavaScript for client-side interactivity is essential.

* **Version Control**: Use Git for version control, enabling collaboration and tracking changes throughout the development process. Platforms like GitHub or Bitbucket can host your repository.

 • Git: Download and installation instructions can be found at: <https://git-scm.com/downloads>

* **Development Environment**: Choose a code editor or Integrated Development Environment (IDE) that suits your preferences, such as Visual Studio Code, Sublime Text, or WebStorm.

• Visual Studio Code: Download from <https://code.visualstudio.com/download>

• Sublime Text: Download from <https://www.sublimetext.com/download>

• WebStorm: Download from [https://www.jetbrains.com/webstorm/download](https://www.jetbrains.com/webstorm/download%20)

To get the Application project from drive:

Follow below steps:

**Install Dependencies:**

• Navigate into the cloned repository directory and install libraries:

cd fitness-app-react

npm install

* **Start the Development Server**:

• To start the development server, execute the following command:

npm start

**Access the App:**

• Open your web browser and navigate to [http://localhost:3000](http://localhost:3000/).

• You should see the application's homepage, indicating that the installation and setup were successful.

 You have successfully installed and set up the application on your local machine. You can now proceed with further customization, development, and testing as needed.

**Installation Steps:**  
1. Clone the repository:  
 git clone <repository-url>  
2. Navigate into the project directory:  
 cd cookbook-frontend  
3. Install dependencies:  
 npm install  
4. Start the development server:  
 npm start

**Folder Structure**

**Client Structure:**  
→ /src – Source code folder  
→ /components – Reusable UI components (Navbar, Footer, Search, etc.)  
→ /pages – Page-level components (Home, RecipeList, RecipeDetail)  
→ /assets – Images, icons, and styling resources  
→ /App.js – Root component  
→ /index.js – Entry point

**Utilities:**  
→ Custom hooks for handling data fetching and state updates.  
 → Utility functions for filtering and categorizing recipes.

**Running the Application**

**•To run the frontend locally:**

**• Start the Development Server**:

• To start the development server, execute the following command:

npm start

**Access the App:**

• Open your web browser and navigate to [http://localhost:3000](http://localhost:3000/).

• You should see the application's homepage, indicating that the installation and setup were successful.

**Component Documentation**

**Key Components:**  
→ Navbar: Provides site-wide navigation links.  
→ Home: Displays featured recipes and categories.  
→ RecipeList: Dynamically renders multiple recipe cards.  
→ RecipeDetail: Shows complete recipe instructions and ingredients.  
→ Search: Allows users to search for recipes by keywords.  
  
**Reusable Components:**  
 → Card Component: Used for displaying individual recipes in a consistent format.  
 → Button Component: Reusable styled buttons for navigation and actions.

**State Management**

**Global State:**  
Global state is lightweight, handled at the App level and passed down via props to child components.  
  
**Local State:**  
Individual components, such as RecipeDetail and Search, maintain their own states using hooks (useState, useEffect).

**User Interface**

**The user interface is clean, minimal, and intuitive. It includes:**  
→ A navigation bar with quick access to Home, Recipes, and Search.  
→ A recipe browsing page displaying cards with thumbnails and titles.  
→ Recipe detail pages with structured ingredient lists and instructions.  
→ Fully responsive design for desktop, tablet, and mobile.

**Styling**

**CSS Frameworks/Libraries:**  
→ Pure CSS and CSS modules for modular styling.  
  
**Theming:**  
→ A consistent color palette and typography are applied throughout to ensure professional aesthetics. The design emphasizes readability and modern simplicity.

**Testing**

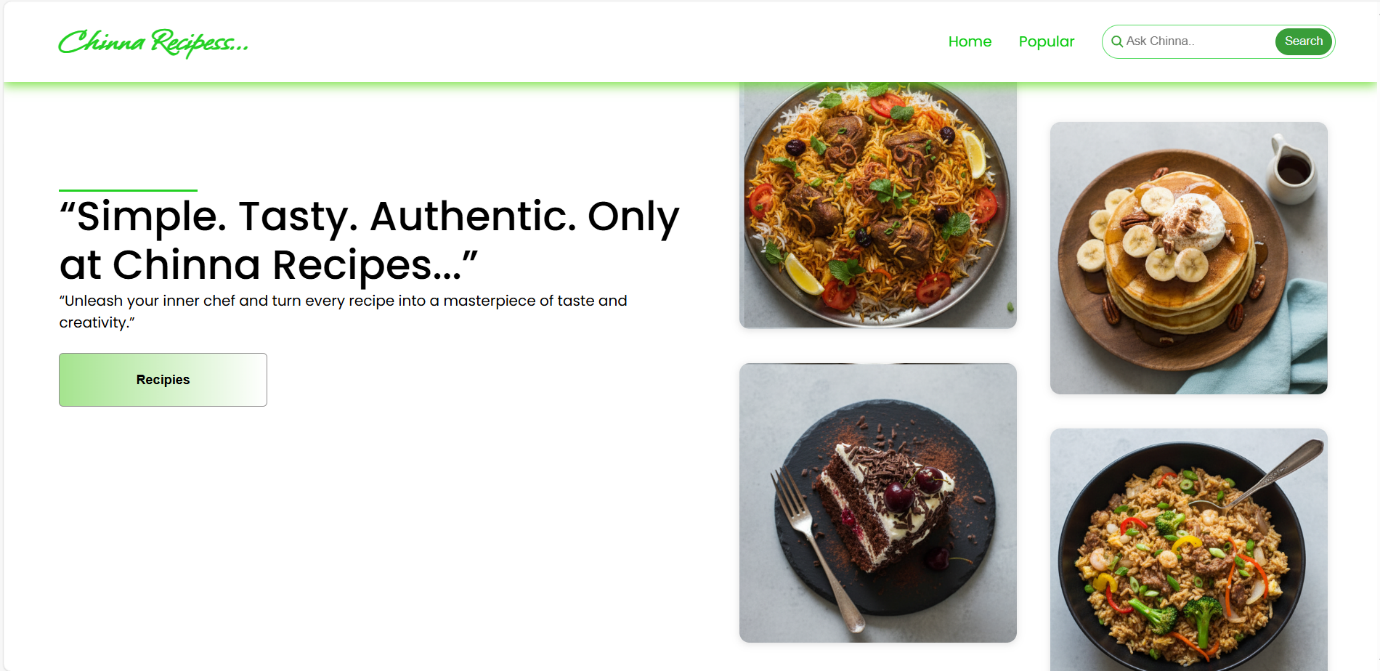
**Testing Strategy:**  
- Manual testing of navigation, search, and recipe detail functionality.  
- Component-level testing using React Developer Tools.

**Code Coverage:**  
Basic coverage ensured through manual validation and debugging tools. Automated testing (e.g., Jest, React Testing Library) can be integrated as a future enhancement.

**Screenshots**

**Hero components**

The hero component of the application provides a brief description about our application and a button to view more recipes.



It features a clean design with a bold tagline, recipe search, and visually appealing food images to engage users.

**Popular categories**

This component contains all the popular categories of recipes..

A screenshot of a menu

AI-generated content may be incorrect.

**Trending Dishes**

This component contains some of the trending dishes in this application.

A screenshot of a food menu

AI-generated content may be incorrect.This section of the website highlights **“Trending Dishes”**, showcasing a curated collection of popular recipes. It attracts users to explore and try out the most-loved recipes of the moment.

**News Letter**

The News letter component provides an email input to subscribe for the recipe newsletters.

A screenshot of a computer

AI-generated content may be incorrect.

**Category dishes page**

The category page contains the list of dishes under a certain category.

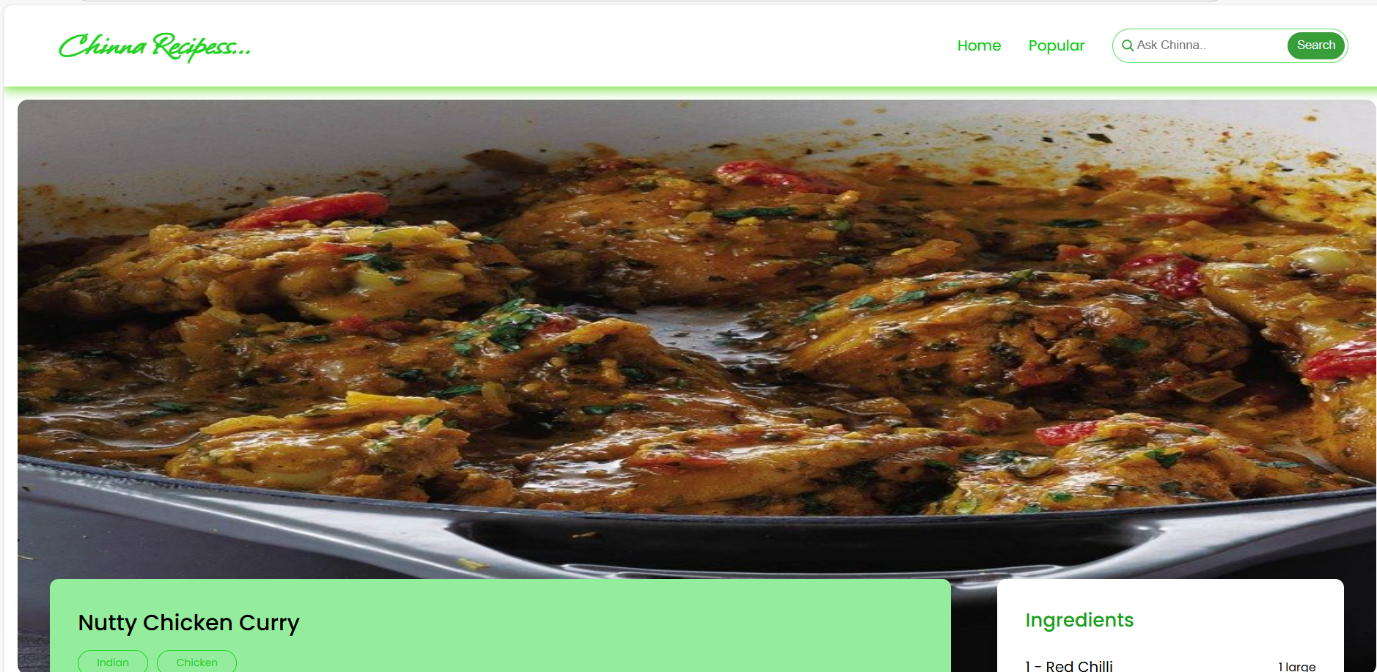
A screenshot of a food menu

AI-generated content may be incorrect.

This page showcases the **Chicken category**, featuring diverse and flavourful dishes from quick bites to hearty meals. Clean visuals and curated recipes make it easy for users to explore and try new favourites.

**Recipe page**

The images provided below shows the recipe page, that includes images, recipe instructions, ingredients and even a tutorial video.

****

**A screenshot of a computer

AI-generated content may be incorrect.**

This page highlights **Nutty Chicken Curry**, combining rich flavours with easy-to-follow steps.  
It features a well-structured **procedure**, a clear **ingredients list**, and a supportive **video tutorial**.  
Together, these elements make cooking simple, engaging, and enjoyable for all users.

**Demo Video**

Demo Video Link : https://drive.google.com/file/d/17zRwg\_tIlMo2OMPyIeVA2bmnNxLD9XXi/view?usp=sharing

**Known Issues**

→ Limited recipe dataset used for demonstration purposes.  
→ No persistent backend storage implemented (recipes reset on refresh).  
→ Search functionality limited to keyword matching.

**Future Enhancement**

* Secure user authentication for personalized accounts and saved favourites.
* Advanced filtering and recommendation system for tailored recipe suggestions.
* Dark mode and theme customization to improve accessibility and comfort.
* Community-driven features like recipe sharing, reviews, and ratings.
* Option to create personalized recipe collections and cooking history.
* Future integration of AI-driven meal planning and dietary suggestions.