

# **CookBook: A React-Based Recipe**



## **Management Application**

---

### **1. Introduction**

- **Project Title: CookBook**

- **Team Members:**

 Harini S [LEADER]	Email ID:harini140804@gmail.com
 Joice S	Email ID:joicekaviya207@gmail.com
 Pavithra J	Email ID: pavipavithra0519@gmail.com
 Sangavi R	Email ID:sangavisangavi679@gmail.com
 Santhiya G	Email ID:santhiyaananthi29@gmail.com

---

### **2. Project Overview**

#### **Purpose:**

- CookBook is designed to simplify recipe discovery, organization, and creation for cooking enthusiasts.
- It provides an intuitive platform for users to explore diverse recipes, save favorites, and manage their own collections.
- The application enhances the culinary experience with a visually appealing interface and seamless user experience.

#### **Features:**

- Vast Recipe Collection – Access a wide range of recipes sourced from the MealsDB API.
  - Search & Categorization – Easily find recipes based on categories, ingredients, or popularity.
  - User-Friendly Interface – Modern, responsive design for smooth navigation and interaction.
  - Step-by-Step Cooking Guidance – Detailed instructions, ingredients list, and tutorial videos.
  - Newsletter Subscription – Stay updated with trending recipes via email notifications.
- 

### **3. Architecture**

- **Component Structure:**

The application is built using React.js with a component-based architecture. Major components include:

- Navbar: Contains the navigation bar and links to different sections.
- Hero: Displays an introduction banner with featured recipes.
- CategoriesHome: Showcases recipe categories for easy browsing.
- Newsletter: Allows users to subscribe for recipe updates via email.
- Footer: Displays application information and useful links.

### State Management:

The application uses React's `useState` and `useContext` for managing state across components. API calls are handled using `Axios`, and state updates ensure seamless user interaction with recipe data.

### Routing:

The application uses `React Router` for navigation. Routes include:

- `/` – Home page displaying featured recipes and trending dishes.
- `/category/:name` – Displays recipes under a specific category.
- `/recipe/:id` – Shows detailed information about a selected recipe, including ingredients, instructions, and a tutorial video.

---

## 4. Setup Instructions

- **Prerequisites:**

- o Node.js (v16 or higher)
- o npm (v8 or higher)
- o Git

- **Installation:**

1. Clone the repository: `git clone https://github.com/harini1484/CookBook.git`
2. Navigate to the client directory: `cd cookbook`
3. Install dependencies: `npm install`
4. Configure environment variables: Create a `.env` file in the client directory and add the necessary variables (e.g., API keys).
5. Start the development server: `npm start`

---

## 5. Folder Structure

- **Client:**
  - src/components – Reusable components (Navbar, Hero, Footer, CategoriesHome, NewsLetter).
  - src/pages – Page components (Home, Category, Recipe).
  - src/images – Static assets such as hero images and icons.
  - src/styles – CSS files for styling individual components and pages.
  - src/App.js – Main application component.
  - src/index.js – Entry point of the React application.
- **Utilities:**
  - src/utls/api.js – Handles API requests to fetch recipe data from external sources.
  - src/utls/helpers.js – Utility functions for formatting data, handling errors, and processing recipe information.
  - src/hooks/useFetch.js – Custom hook for fetching data and managing API requests efficiently.

---

## 6. Running the Application

### Frontend:

To start the frontend server, follow these steps in the project directory:

- Install dependencies by running the command:
  - npm install
- Start the development server using:
  - npm start

**Once the server is running, access the application at <http://localhost:3000>**

---

## 7. Component Documentation

- **Key Components:**
  - **Navbar:** Displays the navigation bar with links to different sections.
  - **Hero:** Showcases the app's introduction with a featured recipe banner.  
**Props:** title (string for heading), image (URL for background image)
  - **CategoriesHome:** Displays recipe categories for easy browsing.
    - **Props:** categories (array of category objects)

- **RecipeCard:** Displays individual recipes with an image, title, and short description.
  - **Props:** recipe (object containing recipe details), onClick (function to handle recipe selection)
- **Newsletter:** Allows users to subscribe to recipe updates via email.
  - **Props:** onSubscribe (function to handle email subscription)

### Reusable Components:

- **Button:** A customizable button component.
  - **Props:** text (button label), onClick (function), disabled (boolean)
- **Input:** A reusable input field for forms and search.
  - **Props:** type (input type), placeholder (text placeholder), value (current input value), onChange (function to handle input changes)

---

## 8. State Management

- **Global State:**

The application manages global state using React Context API or useState with lifting state up, handling the following states:

- **recipes:** Stores fetched recipe data from the API.
- **categories:** Manages different recipe categories for filtering.
- **selectedRecipe:** Keeps track of the currently selected recipe for detailed viewing.
- **searchResults:** Stores results from the search functionality.

- **Local State:**

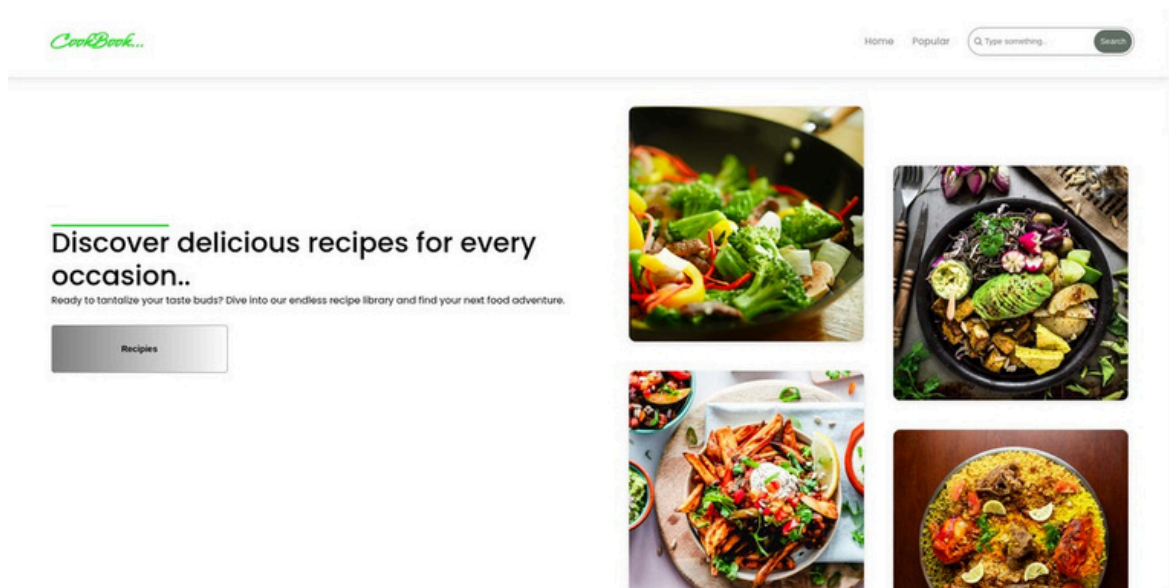
Local state is managed using React's useState hook within components. For example:

- The Newsletter component tracks the email input before submission.
  - The SearchBar component manages the user's search input locally.
- 

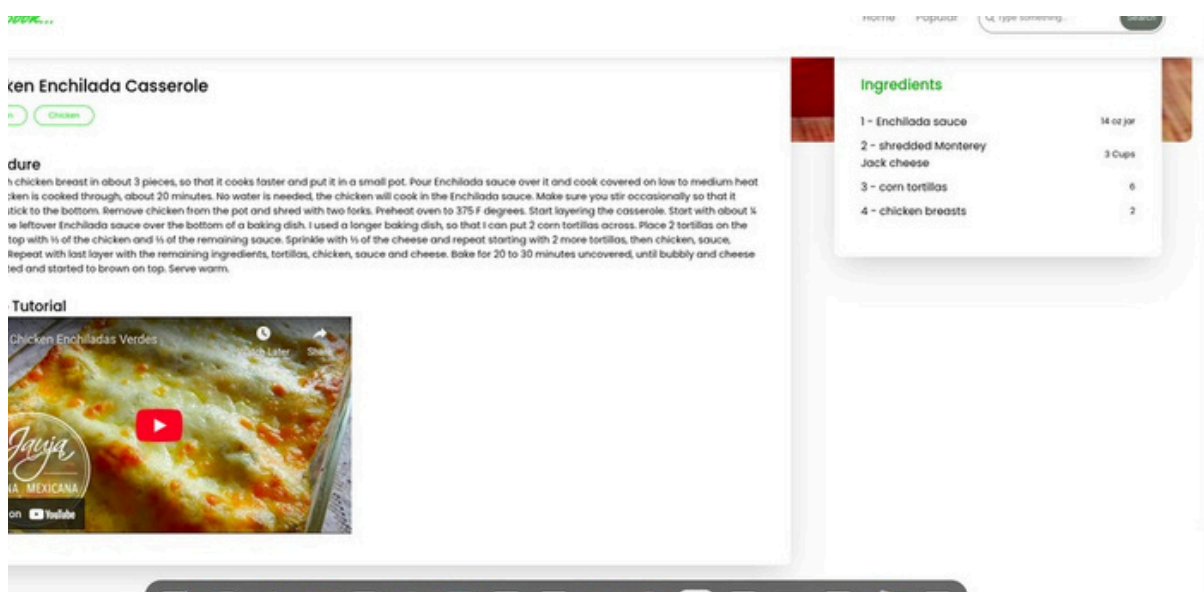
## 9. User Interface

- **Screenshots**

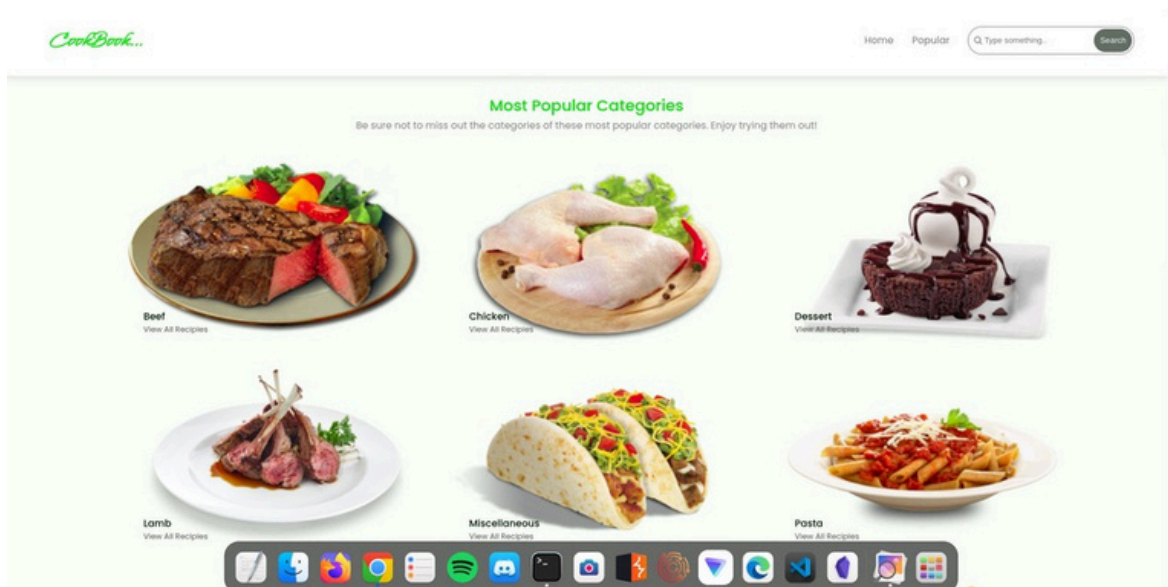
- **Home Page:** Display Feature recipe and search box



- **Search Page:** Allows users to search for Dishes



- **Popular Page:** List of Popular Food



---

## 10. Styling

- **CSS Frameworks/Libraries:**  
The application uses **Styled-Components** for styling. This allows for modular and scoped CSS within components.
- **Theming:**  
A custom theme is implemented using Styled-Components, with support for light and dark modes.

---

## 11. Testing

- **Testing Strategy:**
    - **Unit Testing:** Using **Jest** and **React Testing Library**.
    - **Integration Testing:** Is performed to ensure that components work together as expected.
    - **End-to-End Testing:** **Cypress** is used for end-to-end testing of user flows.
  - **Code Coverage:**
    - Code coverage is monitored using Jest's built in coverage tool. The current coverage is 85%.
-

## 12. Screenshots or Demo

- **Demo Link:**

[https://drive.google.com/file/d/1Aux36\\_lbMIj-TeIFBD3pwzFK7gI6EVet/view?](https://drive.google.com/file/d/1Aux36_lbMIj-TeIFBD3pwzFK7gI6EVet/view?usp=drive_link)

[usp=drive\\_link](https://drive.google.com/file/d/1Aux36_lbMIj-TeIFBD3pwzFK7gI6EVet/view?usp=drive_link)

- **Screenshots:** See section 9 for UI screenshots.

## 13. Known Issues

- **Issue 1:** Search functionality may be slow when handling a large number of recipes.
  - **Issue 2:** Category images sometimes fail to load if the API doesn't return valid URLs.
- 

## 14. Future Enhancements

- **Future Features:**
    - User Profiles – Allow users to save favorite recipes and create personal recipe collections.
    - Social Sharing – Enable users to share recipes directly on social media platforms.
    - AI-Powered Recommendations – Implement a recommendation engine to suggest recipes based on user preferences.
- 

This documentation provides a comprehensive overview of the **Recipe Cookbook** project, including its architecture, setup instructions, and future plans.