**Project Documentation format**

**CookBook: Your Virtual Kitchen Assistant**

**1. Introduction**

* **Project Title:** CookBook
* **Team Members:**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Names and Roles** | **EmailID** |
| **1** | **Sareeshma Rehan M -** (Leader) | rehansareeshma@gmail.com |
| **2** | **Vedhavalli K** | krishnamurthi21.05.2004@gmail.com |
| **3** | **Prathiba M** | prathibamagesh@gmail.com |
| **4** | **Samadharshini N** | samadharashini08@gmail.com |

**2. Project Overview**

**Purpose**

CookBook is a web application designed to help users explore, search, and categorize recipes. It provides an intuitive interface for browsing different cuisines, viewing recipe details, and subscribing to newsletters.

**Features**

* Recipe listing with filtering options
* Recipe details page
* Category-based browsing
* Newsletter subscription
* Responsive UI with a modern design

**3. Architecture**

**Component Structure**

The application follows a modular React component structure:

* **Navbar.jsx** - Handles site navigation.
* **Hero.jsx** - Displays homepage banner.
* **CategoriesHome.jsx** - Shows recipe categories.
* **Footer.jsx** - Contains footer information.
* **Recipe.jsx** - Displays individual recipe details.
* **Category.jsx** - Lists recipes under a specific category.

**State Management**

* The application uses **React useState** for local state management.
* Future enhancements may include **Context API** or **Redux** for better global state handling.

**Routing**

* **React Router** (react-router-dom) is used for navigation.
* Routes include:
  + / - Home Page
  + /category/:id - Recipe categories
  + /recipe/:id - Recipe details

**4. Setup Instructions**

**Prerequisites**

* Node.js (v14+ recommended)
* npm or yarn

**Installation**

1. Clone the repository:
2. git clone https://github.com/SareeshmaRehan/Recipe.git
3. cd Recipe
4. Install dependencies:
5. npm install
6. Start the development server:
7. npm start

**5. Folder Structure**

CookBook/

├── public/

│ ├── index.html

│ ├── favicon.ico

├── src/

│ ├── components/

│ │ ├── Navbar.jsx

│ │ ├── Footer.jsx

│ │ ├── Hero.jsx

│ │ ├── CategoriesHome.jsx

│ ├── pages/

│ │ ├── Home.jsx

│ │ ├── Recipe.jsx

│ │ ├── Category.jsx

│ ├── styles/

│ │ ├── Navbar.css

│ │ ├── Home.css

│ ├── App.js

│ ├── index.js

├── package.json

└── README.md

**6. Running the Application**

* To start the project locally:
* npm start
* The application runs on **localhost:3000** by default.

**7. Component Documentation**

**Key Components**

* **Navbar.jsx** - Displays navigation links.
* **Hero.jsx** - Main banner section.
* **CategoriesHome.jsx** - Displays different food categories.
* **Recipe.jsx** - Renders recipe details.

**Reusable Components**

* **Button.jsx** - Custom button component.
* **Card.jsx** - Used for recipe and category previews.

**8. State Management**

* **Global State:** Currently, no global state management is used.
* **Local State:** React useState is used within individual components.

**9. User Interface**

* The UI includes:
  + A responsive navbar.
  + Recipe category sections.
  + Individual recipe detail pages.

**10. Styling**

**CSS Frameworks/Libraries**

* **CSS Modules** for scoped styles.
* Future enhancement: **Styled Components** for dynamic styling.

**Theming**

* Currently, the app follows a predefined theme with CSS variables.
* Dark mode support can be added in the future.

**11. Testing**

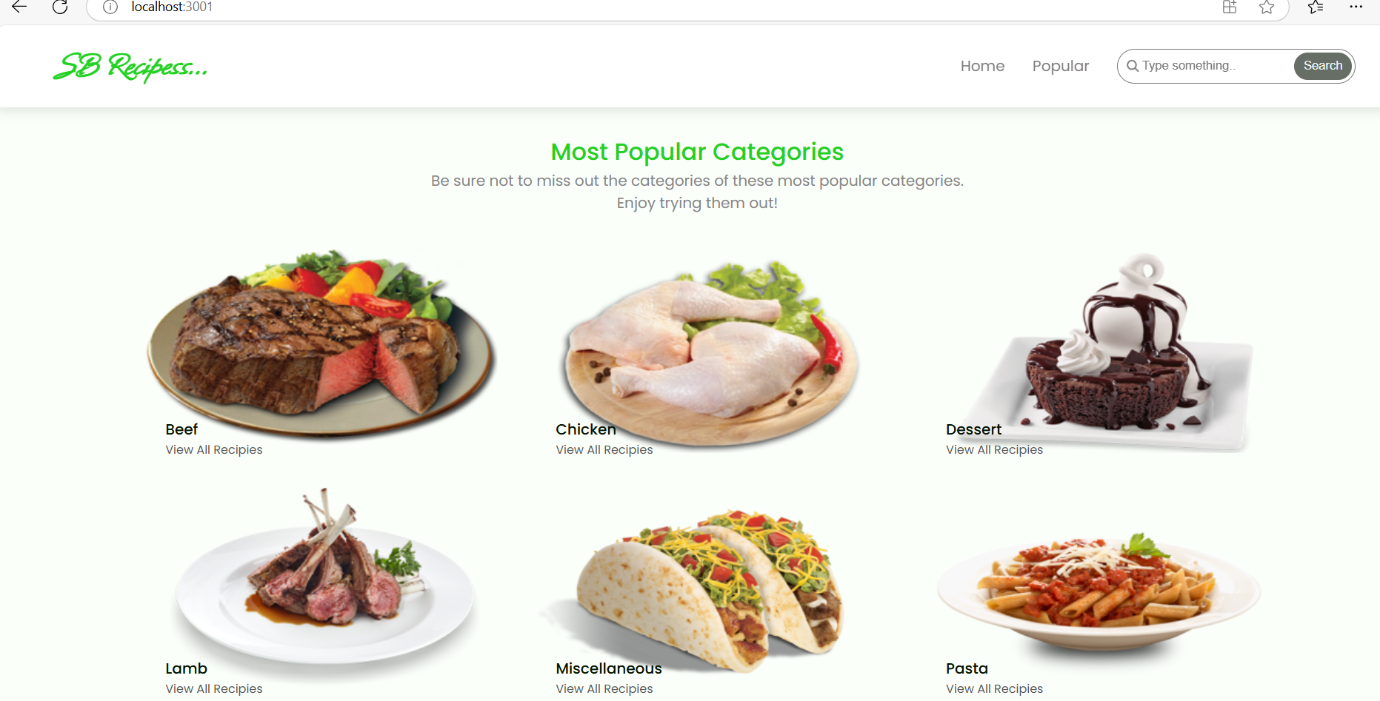
**Testing Strategy**

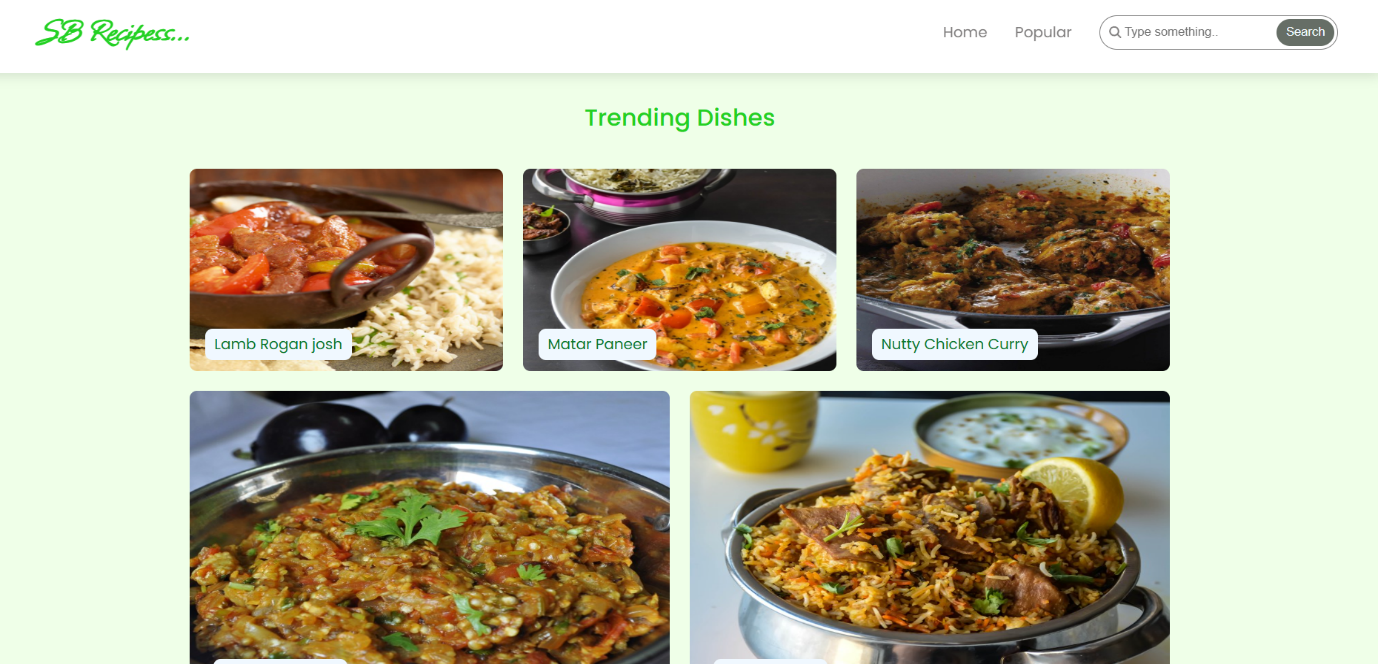
* Uses **Jest** and **React Testing Library** for unit tests.

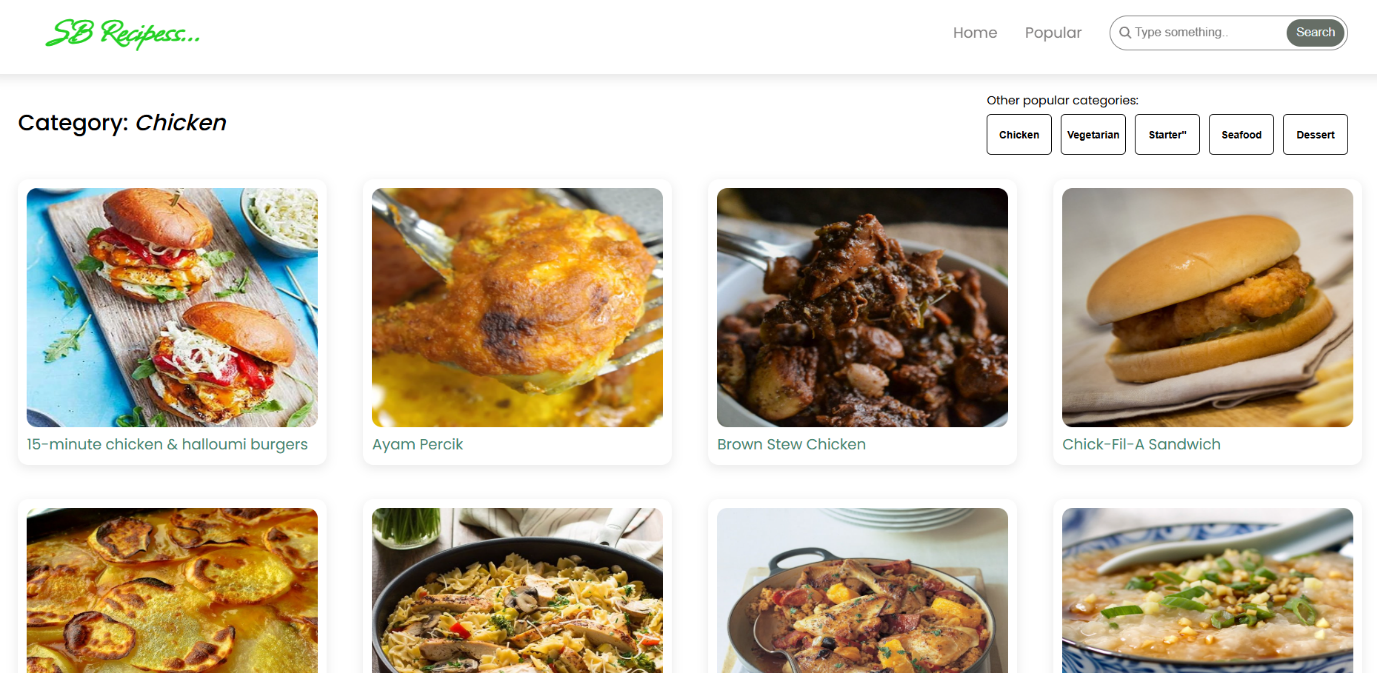
**Code Coverage**

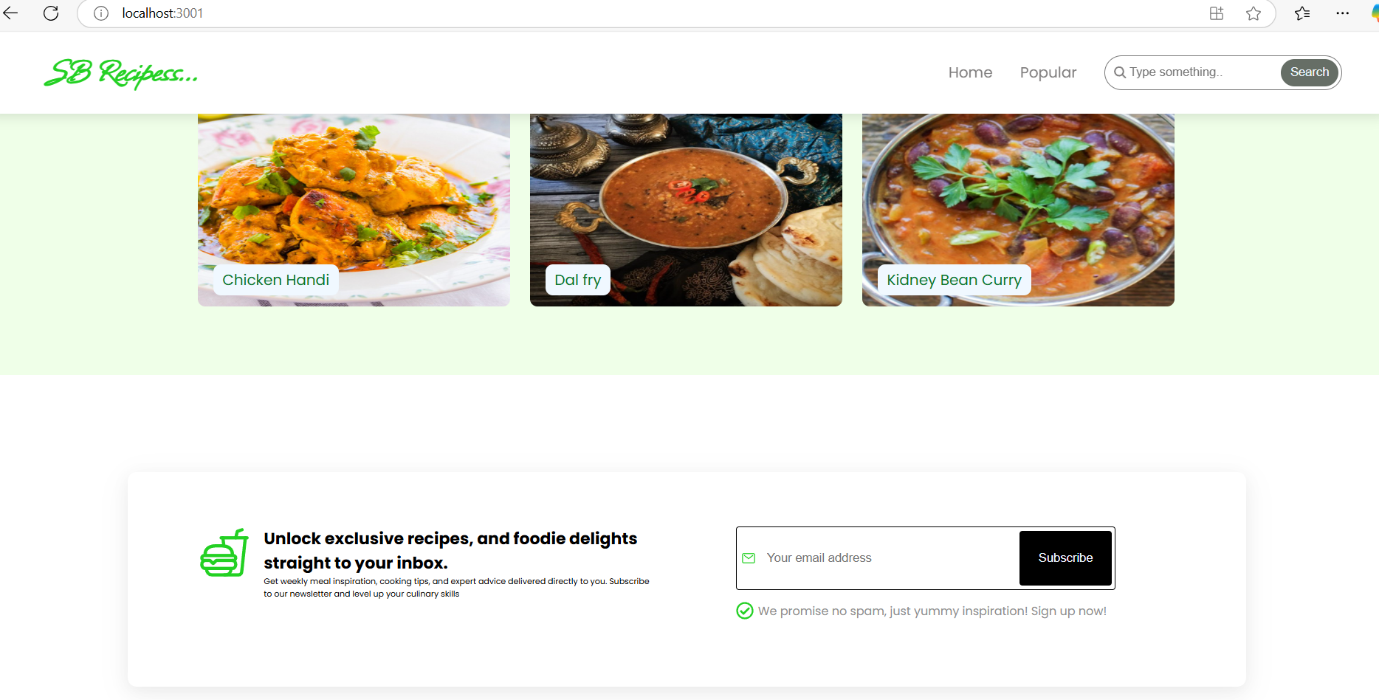
* Run tests with:
* npm test

**12. Screenshots**









**13. Known Issues**

* No global state management yet.
* API integration is not yet implemented.

**14. Future Enhancements**

* Add user authentication.
* Implement API for fetching recipes dynamically.
* Improve UI animations.