**Frontend Development with React.js**

**Project Documentation**

**Introduction**

**Project Title:** CookBook: Your Virtual Kitchen Assistant

**Team id**:150895

**Team Members:**

* A.Arul lishi \_ stu\_arullishi\_a@cttewc.edu.in
* S.Inbarathi – stu\_inbarathi\_s@cttewc.edu.in
* S.Hannah Hephzibah – stu\_hannahhephzibah\_s@cttewc.edu.in
* Zeenath Fareen R – stu\_zeenathfareen\_r@cttewc.edu.in
* Aysha Siddiqa M – stu\_ayshasiddiqa\_m@cttewc.edu.in

**Project Overview**

**Purpose:** The primary purpose of CookBook is to provide a user-friendly platform for both novice and professional chefs to discover, organize, and create recipes. The project aims to enhance the culinary experience through an intuitive interface and robust features.

**Features:**

* **Recipes from the MealsDB API:** Access a vast library of international recipes.
* **Visual recipe browsing:** Explore categories through curated image galleries.
* **Intuitive and user-friendly design:** Clean interface with seamless navigation.
* **Search feature:** Easily find dishes using dynamic search options.

**Architecture**

**Component Structure:**

* **Hero Component:** Brief introduction and button to explore recipes.
* **Popular Categories Component:** Displays all popular recipe categories.
* **Trending Dishes Component:** Highlights trending dishes.
* **Newsletter Component:** Includes email input to subscribe for newsletters.
* **Category Dishes Page:** Lists all dishes under a selected category.
* **Recipe Page:** Displays ingredients, instructions, and a demo video.

**State Management:**

* Used React's **useState** hook to manage component-level states.
* Employed **useEffect** to fetch data from APIs and handle side effects.

**Routing:**

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

**Setup Instructions**

**Prerequisites:**

* **Node.js and npm:** Download from [https://nodejs.org/en/download](https://nodejs.org/en/download/)/
* **React.js:** Installed using npx create-react-app my-react-app
* **React Router Dom:** For routing
* **Axios:** To fetch data from MealsDB API
* **Bootstrap/Tailwind CSS:** For styling and UI components

**Installation:**

1. **Clone the repository:** Download the code from the drive link:
2. **Navigate into the cloned directory:**
3. cd recipe-app-react
4. **Install dependencies:**
5. npm install
6. **Start the development server:**
7. npm start
8. **Access the app:** Open your browser and go to http://localhost:3000

**Folder Structure**

* **Components:** Contains reusable UI components (Navbar, Hero, Categories, etc.)
* **Pages:** Holds page-specific components (Home, Category, Recipe)
* **Styles:** Contains CSS files for styling

**Running the Application**

* **Start the frontend server locally:**
* npm start

**Component Documentation**

**Key Components:**

* **Hero Component:** Introduction section with a call-to-action button.
* **Popular Categories Component:** Displays recipe categories.
* **Trending Dishes Component:** Highlights popular recipes.
* **Recipe Page:** Detailed view of a selected recipe.

**Reusable Components:**

* **Navbar:** For app navigation.
* **Footer:** Contains additional links and contact info.

**State Management**

**Global State:** Not used; relies on component-level state managed with **useState**.

**Local State:** Each component maintains its state using **useState** and updates data with **useEffect**.

**User Interface**

* **Homepage:** Displays trending recipes, popular categories, and a search bar.
* **Category Page:** Lists all recipes under a category.
* **Recipe Page:** Includes recipe details, ingredients, and a tutorial video.

**Styling**

* Used **Bootstrap/Tailwind CSS** for pre-built styles.
* Custom CSS for specific styling needs.

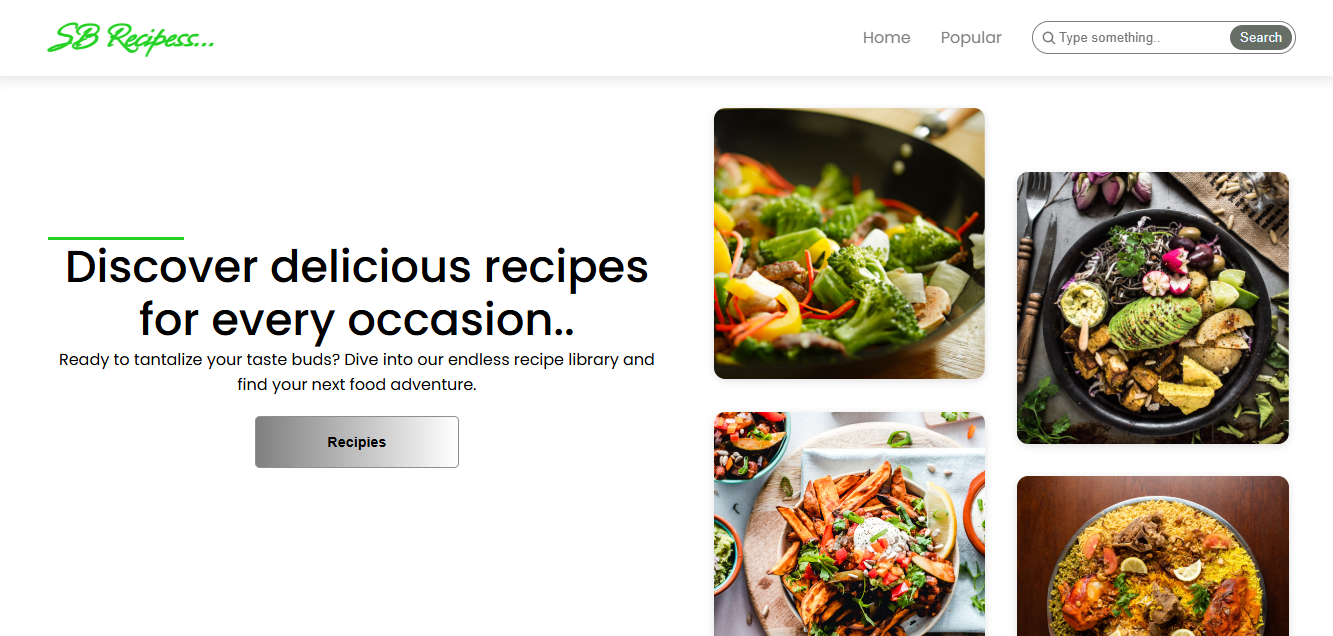
**Testing**

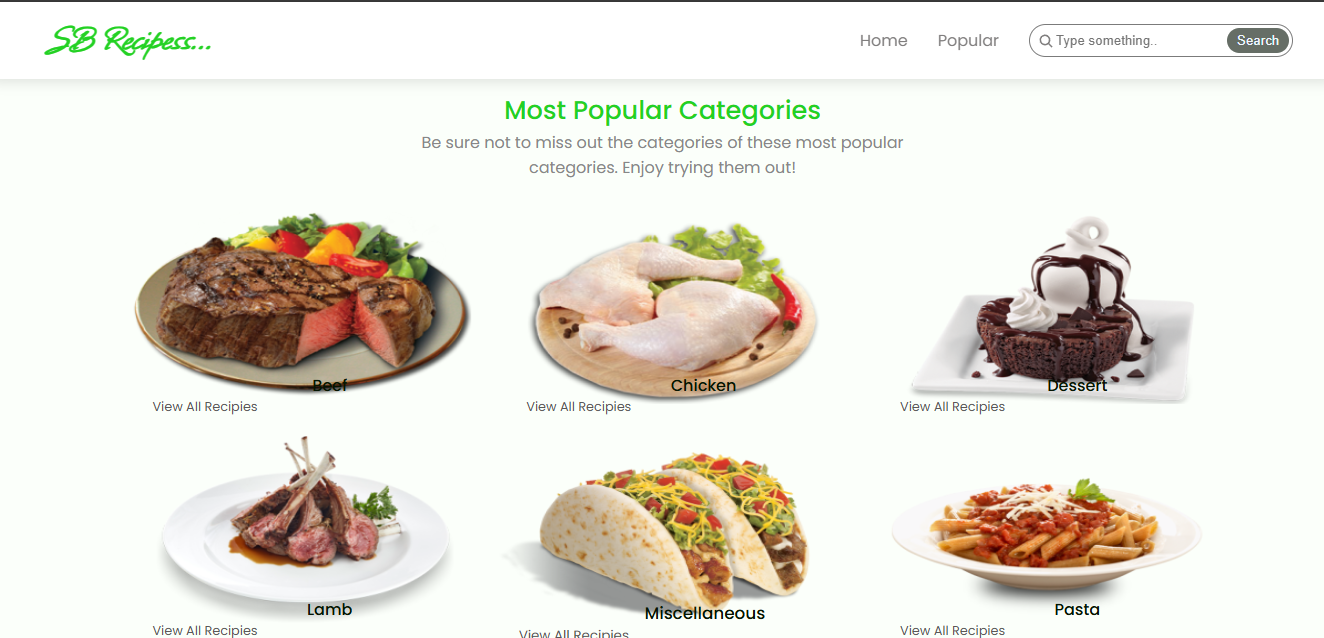
**Testing Strategy:**

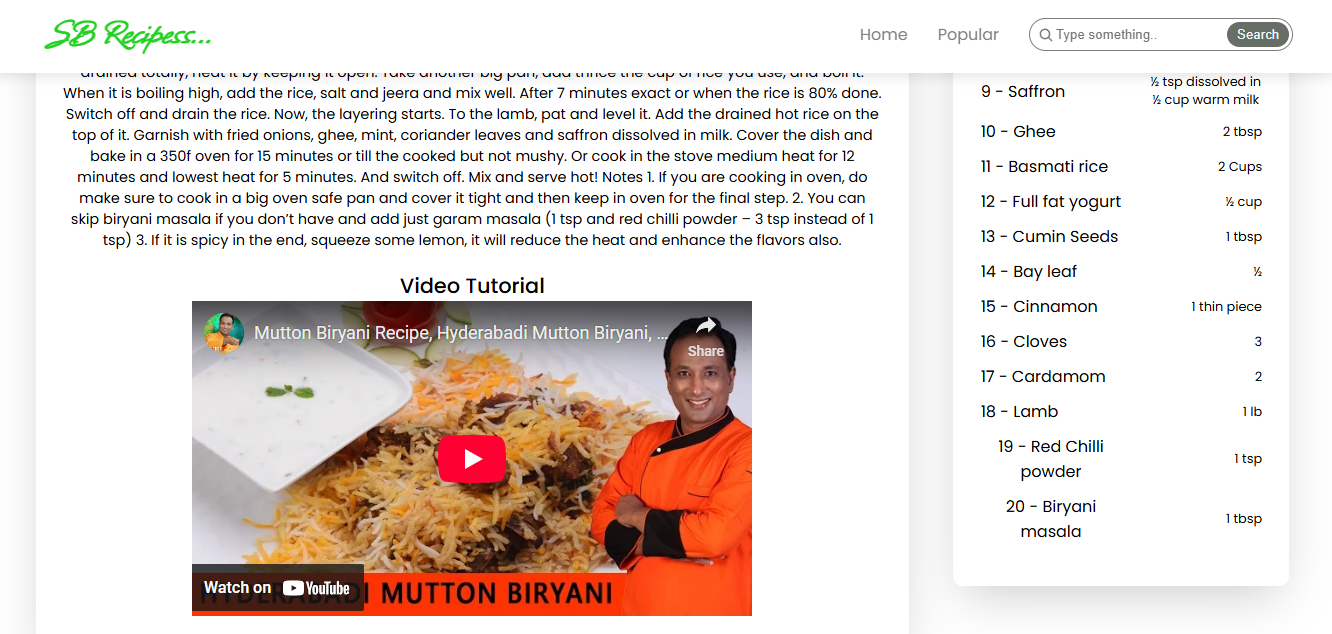
* Manual testing for UI responsiveness and API data fetching.
* Basic unit tests for key components using React Testing Library.

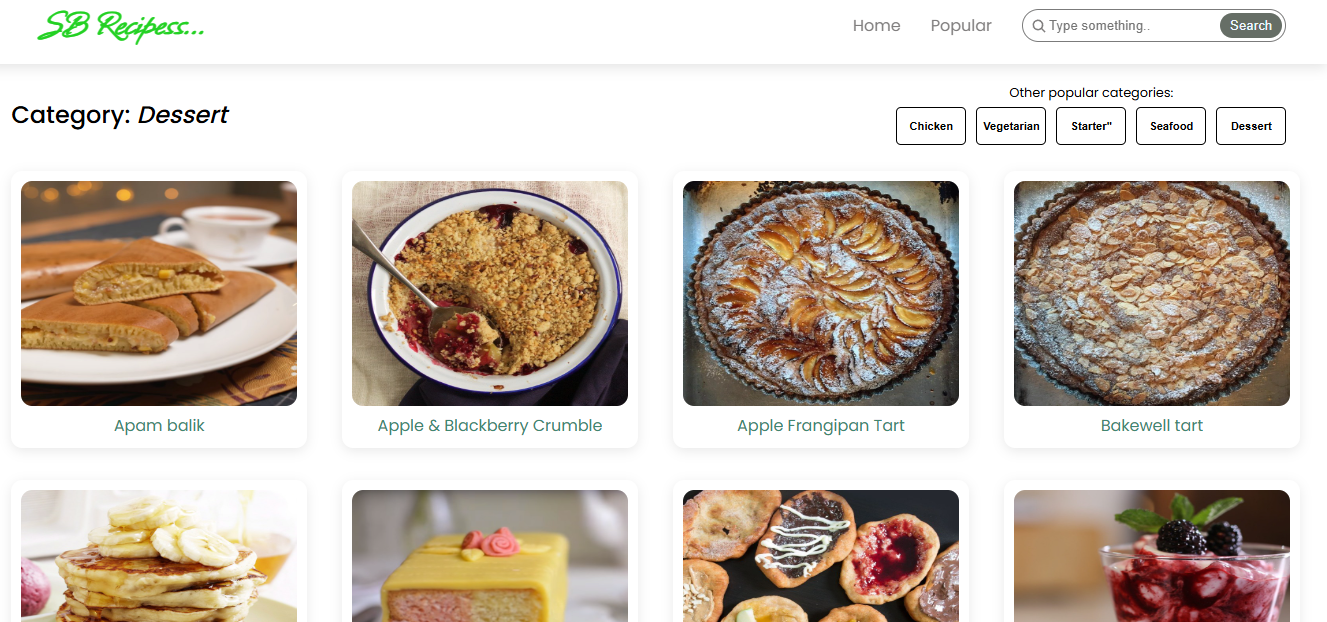
**Screenshots & Demo video**

**Project Demo Link:** <https://drive.google.com/file/d/1uO9GGRWiegSWxmh9ovgBt7DvAO30Y9_o/view?usp=sharing>









**Known Issues**

* Occasionally slow API responses from MealsDB.
* UI may slightly lag during heavy API calls.

**Future Enhancements**

* **Add user authentication:** Allow users to save their favorite recipes.
* **Implement Redux:** Introduce global state management.
* **Enhance UI animations:** Add smooth transitions and effects.