**Cook Book**

**Introduction:**

Welcome to Cook Book: Your Virtual Kitchen Assistant!

Cooking made simple, fun, and effortless! Cook Book is your all-in-one digital companion, helping you discover, organize, and perfect your favourite recipes. Whether you're a seasoned chef or a beginner in the kitchen, our platform provides step-by-step guidance, smart ingredient substitutions, meal planning tips, and more.

Why Choose Cook Book?

* Explore Thousands of Recipes – Find delicious recipes for every occasion.
* Smart Cooking Assistant – Get personalized recommendations based on your preferences.
* Ingredient Substitutions – Missing an ingredient? We’ve got you covered.
* Meal Planning & Grocery Lists – Stay organized with our smart planner.
* Interactive Cooking Mode – Follow along with easy step-by-step instructions.

Start your culinary journey today with CookBook and bring the joy of cooking to your home

**Description:**

Your Virtual Cooking App is your ultimate digital cookbook, designed to make cooking fun, easy, and accessible for everyone. Whether you're a beginner or an experienced chef, this app offers step-by-step guidance, interactive tutorials, and a vast collection of recipes from around the world. With a user-friendly interface, you can explore personalized recipe recommendations, smart shopping lists, and ingredient substitutions to enhance your cooking experience.

From quick and easy meals to gourmet dishes, healthy options, and international cuisines, the app caters to all tastes and dietary preferences. Enjoy features like video tutorials with expert chefs, voice-guided cooking assistance, and adjustable serving sizes. Smart tools such as AI-powered meal planning, pantry inventory tracking, and nutritional breakdowns make it easier than ever to organize your kitchen and maintain a healthy diet.

Join a vibrant community of food lovers where you can share your creations, follow fellow home chefs, and even upload your own recipes. Discover essential cooking tips, time-saving meal prep strategies, and handy ingredient substitutes to elevate your skills. With seamless sync across devices, regular updates with new recipes, and exciting cooking challenges, *Your Virtual Cooking App* transforms your kitchen into a creative culinary space. Download today and start your cooking journey.

**Project Goals and Objectives:**

The primary goal of Cook Book is to provide a user-friendly platform that caters to individuals passionate about cooking, baking, and exploring new culinary horizons. Our objectives include:

• **User-Friendly Experience:** Create an interface that is easy to navigate, ensuring users can effortlessly discover, save, and share their favourite recipes.

• **Comprehensive Recipe Management:** Offer robust features for organizing and managing recipes, including advanced search options.

• **Technology Stack:** Leverage modern web development technologies, including React.js, to ensure an efficient, and enjoyable user experience.

**Features of Cook Books:**

* **Recipes from the Meals DB API:** Access a vast library of international recipes spanning diverse cuisines and dietary needs.
* **Visual recipe browsing:** Explore recipe categories and discover new dishes through curated image galleries.
* **Intuitive and user-friendly design:** Navigate the app effortlessly with a clean, modern interface and clear navigation.
* **Search feature:** various dishes can be accessed easily through the search feature.

**PRE-REQUISITES:**

**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 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. ✓ 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>

**✓ Get the code:**

• Download the code from the drive link given below: https://drive.google.com/drive/folders/1u8PnV\_mE0mwKkH\_CvuNpliZtRLJZMqrO?usp=sharing Install Dependencies:

• Navigate into the cloned repository directory and install libraries: cd recipe-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>.

• You should see the recipe app'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.

**State Management with useState Hook:**

The code utilizes the useState hook to create a state variable named categories. This variable acts as a container to hold the fetched data, which in this case is a list of meal categories. Initially, the categories state variable is set to an empty array [ ].

**Fetching Data with useEffect Hook:**

The useEffect hook is employed to execute a side effect, in this instance, fetching data from an API. The hook takes a callback function (fetchCategories in this case) and an optional dependency array. The callback function is invoked after the component renders and whenever the dependencies in the array change. Here, the dependency array is left empty [], signifying that the data fetching should occur only once after the component mounts.

**Fetching Data with fetchCategories Function:**

An asynchronous function named fetchCategories is defined to handle the API interaction. This function utilizes the axios.get method to make a GET request to a specified API endpoint (https://www.themealdb.com/api/json/vi/1/categories.php in this example). This particular endpoint presumably returns a JSON response containing a list of meal categories.

**Processing API Response:**

The .then method is chained to the axios.get call to handle a successful response from the API. Inside the .then block, the code retrieves the categories data from the response and updates the React component's state using the setCategories function. This function, associated with the useState hook, allows for modification of the categories state variable. By calling setCategories(response.data.categories), the component's state is updated with the fetched list of meal categories.

● It leverages the useState hook to establish a state variable named categories. This variable acts as a container to hold the fetched data, which is initially set to an empty array [].

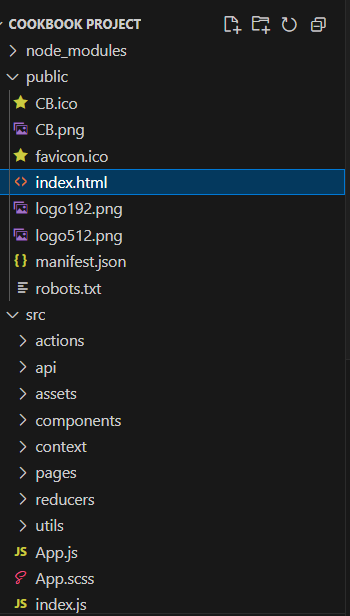
● The useEffect hook comes into play to execute a side effect, in this instance, fetching data from an API endpoint. The hook takes a callback function (fetchCategories in this case) and an optional dependency array. The callback function is invoked after the component renders and whenever the dependencies in the array change. Here, the dependency array is left empty [], signifying that the data fetching should occur only once after the component mounts.

● The fetchCategories function is an asynchronous function responsible for handling the API interaction. This function utilizes the axios.get method to make a GET request to a predetermined API endpoint (https://www.themealdb.com/api/json/vi/1/categories.php in this example). This particular endpoint presumably returns a JSON response containing a list of meal categories.

● The code snippet employs the .then method, which is chained to the axios.get call, to handle a successful response from the API. Inside the .then block, the code retrieves the categories data from the response and updates the React component's state using the setCategories function. This function, associated with the useState hook, allows for modification of the categories state variable. By calling setCategories(response.data.categories), the component's state is updated with the fetched list of meal categories.

● An optional error handling mechanism is incorporated using the .catch block. This block is designed to manage any errors that might arise during the API request. If an error occurs, the .catch block logs the error details to the console using the console.error method. This rudimentary error handling mechanism provides a way to identify and address potential issues during the data fetching process.

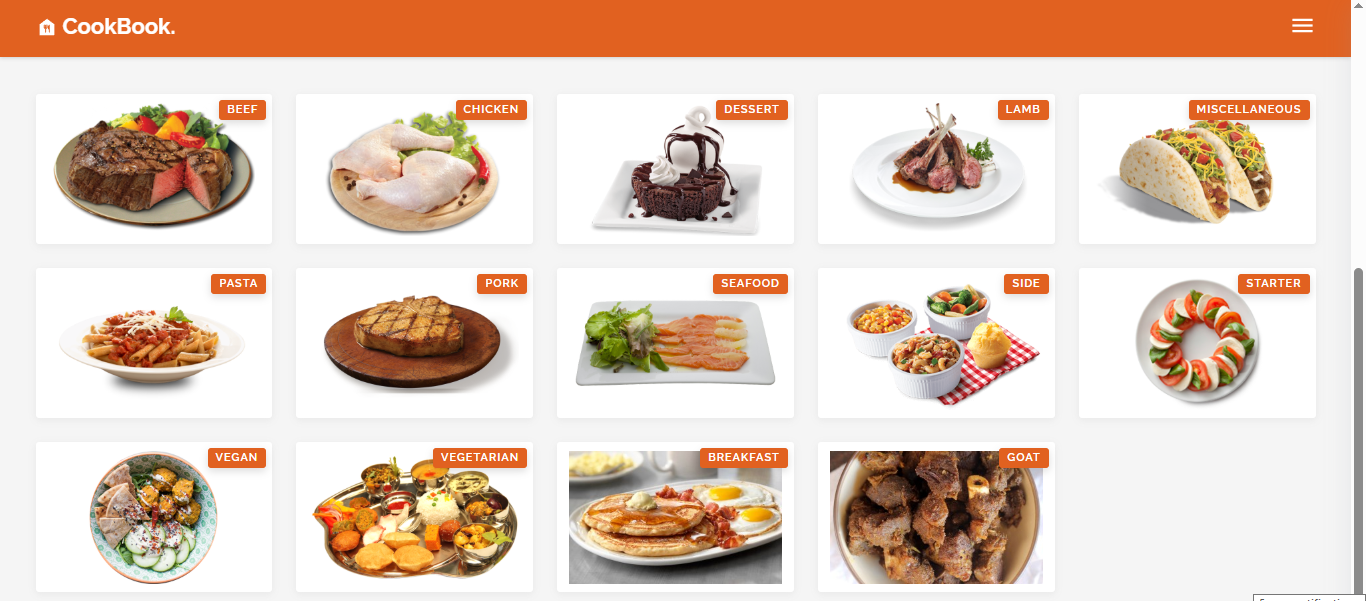
**Program Structure :**



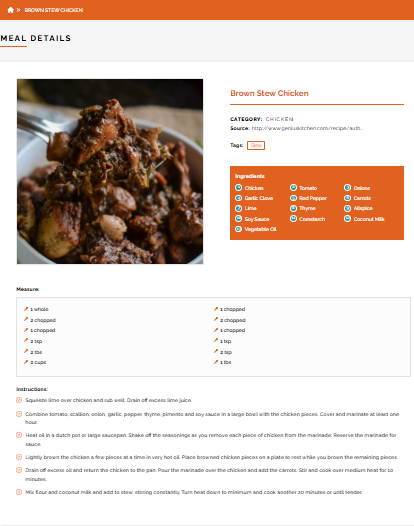
**Project Interface :**



**Categories Section :**



**Ingredient and Instruction Page :**



**Conclusion:**

By combining HTML, CSS, JavaScript, and React.js, this Recipe App is a powerful and engaging tool for users who love cooking and exploring new recipes. Its intuitive design, interactive features, and smooth performance ensure that users can find, save, and share recipes effortlessly.

**Team Details :**

**Team Leader :** G.Deepak

**Team Members :**

1. Adhitya.M
2. Aravindan.M
3. Dhayanithy maran.S
4. GIridharan.M