**VALLIAMMAL COLLEGE FOR WOMEN(1363)**

**DEPARTMENT OF COMPUTER SCIENCE WITH DATA SCIENCE**

**PROJECT TITLE : COOK BOOK : YOUR VIRTUAL KITCHEN ASSISTANT**

**TEAM ID :155763**

**TEAM MEMBERS: 1. AYISHA FATHIMA SH (TEAM LEADER)**

**2. NANDHINI M**

**3. PAVITHRA M**

**4. FARHANA FATHIMA M**

**5. BANU N**

* **GITHUB LINK (CODING AND DOCUMENTATION)**

[**https://github.com/Ayisha004/Cookbook\_AyishaFathima.git**](https://github.com/Ayisha004/Cookbook_AyishaFathima.git)

* + **GOOGLE DRIVE LINK (DEMO VIDEO LINK)**

[**https://drive.google.com/drive/folders/1aZIt6-DWISVGXO5yQ28HtrqQE3E4zcIa**](https://drive.google.com/drive/folders/1aZIt6-DWISVGXO5yQ28HtrqQE3E4zcIa)

**CookBook**: Your Virtual Kitchen Assistant

**(React Application)**

1. **Introduction:**

CookBook is a revolutionary web application designed to change the way you discover, organize, and create recipes. It caters to both novice and professional chefs, offering a user-friendly interface, robust features, and a vast collection of inspiring recipes.

1. **Project Overview**

**Welcome to CookBook!**

A revolutionary web app designed for cooking enthusiasts and professional chefs alike. With an intuitive interface and powerful features, CookBook transforms recipe discovery, organization, and creation. Explore dynamic search, share culinary inspiration, and connect with a passionate cooking community. Experience the future of recipe management—where every dish is an adventure!

**Project Goals and Objectives:**

The primary goal of CookBook 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 CookBooks:**

* **Recipes from the MealsDB 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.

1. **Technical Architecture:**

The user experience starts with the CookBooks web application's UI, likely built with a framework like React or Vue.js for a smooth, single-page experience. This UI interacts with an API client specifically designed for CookBooks. This client handles communication with the backend, but with a twist: it leverages Rapid API, a platform providing access to various external APIs. This suggests CookBooks might integrate external data feeds or functionalities through Rapid API, enriching the user experience without building everything from scratch.

1. **SETUP INSTRUCTION**

**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

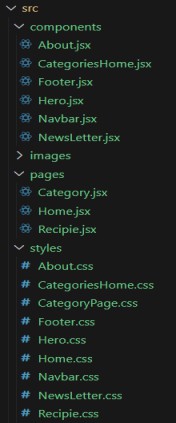
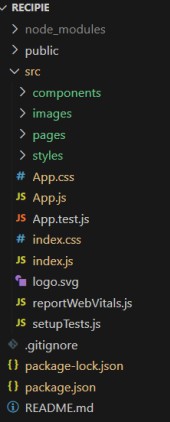
This command launches the development server, and you can access your React app at [http://localhost:3000](http://localhost:3000/) in your web browser.

* **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](http://www.sublimetext.com/download)
  + WebStorm: Download from https://[www.jetbrains.com/webstorm/download](http://www.jetbrains.com/webstorm/download)
* Install Node.js and npm
* Clone the repository:
* git clone https://github.com/Ayisha004/Cookbook\_AyishaFathima.git
* Navigate to the project directory
* cd recipe-app-react
* Install dependencies:
* npm install
* Start the development server:
* npm start
* Open http://localhost:3000 in your web browser

1. **Folder structure:**

****

In this project, we’ve split the files into 3 major folders, *Components, Pages and Styles.* In the

pages folder, we store the files that acts as pages at different url’s in the application. The components folder stores all the files, that returns the small components in the application. All the styling css files will be stored in the styles folder.

1. **Running the Application**

* Ensure dependencies are installed (npm install)
* Start the development server (npm start)
* Open http://localhost:3000 in a browser

**7. Component Documentation :**

**Project demo:**

* Running the Application
* Ensure dependencies are installed (npm install)
* Start the development server (npm start)
* Open http://localhost:3000 in a browser

The project begins with the **setup phase**, where dependencies are installed, and the folder structure is organized. Moving into the **development phase**, routing is implemented using React Router, and reusable components are created and styled. Data is then fetched from the MealsDB API to display categories, along with implementing search functionality and developing pages for recipe details and category listings. Next, the **testing and debugging phase** ensures reliability through unit testing, debugging API requests, and optimizing the user experience across devices. Finally, in the **deployment phase**, performance is optimized before deploying the project on a hosting platform like Netlify or Vercel, followed by sharing the project demo link.

**Important Code snips:**

* + **Fetching all the available categories**

Here, with the API request to Rapid API, we fetch all the available categories.

This code snippet demonstrates how to fetch data from an API and manage it within a React component. It leverages two key functionalities: state management and side

1. **State Management**

**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 (http[s://www.themealdb.com/api/json/vi/1/categories.php](http://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.

## Fetching the food items under a particular category

Now, with the API request, we fetch all the available food items under the certain category.

A computer screen with text and images

AI-generated content may be incorrect.

This React code snippet manages data fetching from an API. The component uses the **useState** hook to store fetched categories, initially set as an empty array. The **useEffect** hook triggers data fetching once after the component mounts. The **fetchCategories** function, an asynchronous function, retrieves meal categories from the API using **axios.get**. Upon a successful response, the data updates the state via **setCategories**. An error handling mechanism with a **.catch** block logs any issues to the console, ensuring smooth debugging

* + **Fetching Recipe details**

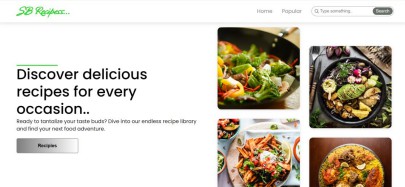
With the recipe id, we fetch the details of a certain recipe.

A computer screen with text and images

AI-generated content may be incorrect.

This React code manages fetching recipe data from an API and storing it within a state variable

1. **User Interface snips**
   * **Hero components**

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

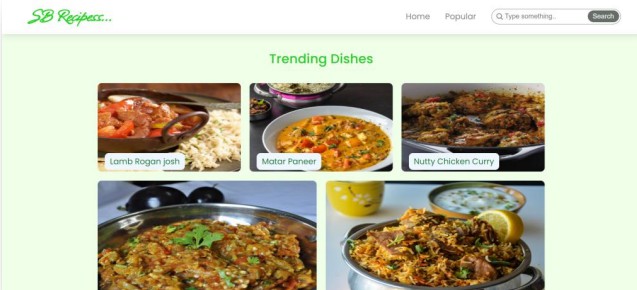
## Popular categories

This component contains all the popular categories of recipes..



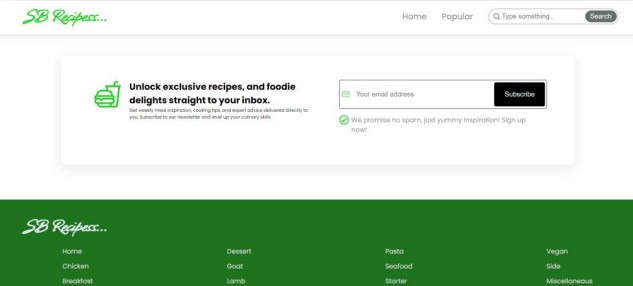
## Trending Dishes

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



## News Letter

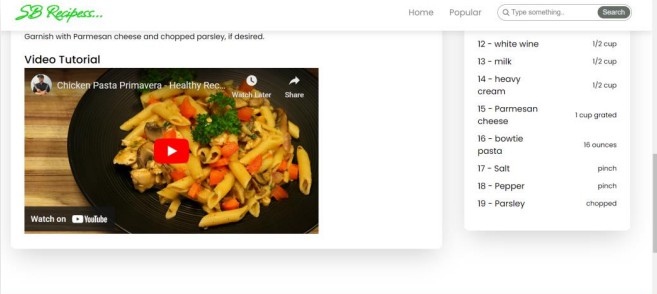
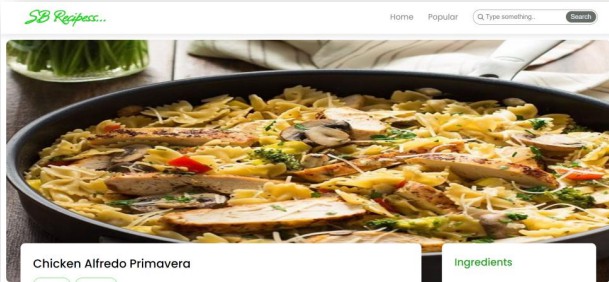
The news letter component provides an email input to subscribe for the recipe newsletters.



## Category dishes page

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

## Recipe page

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

1. **Future Enhancements**

* User Authentication: Allow users to create accounts and save favorite recipes
* Meal Planning Feature: Enable users to plan meals for the week and generate shopping lists
* AI-Based Recipe Suggestions: Suggest recipes based on user preferences and available ingredients
* Community & Sharing: Users can share their own recipes and interact with others
* Dark Mode: Provide a dark mode option for a better user experience
* Mobile App Version: Develop a mobile version for iOS and Android

1. **Conclusion**

CookBook is a powerful and user-friendly web application designed to transform the way users explore and manage recipes. With seamless API integration, intuitive UI, and advanced features such as video tutorials and search filters, CookBook offers an engaging experience for food enthusiasts. Future enhancements, including user authentication, AI-based suggestions, and a mobile version, will further elevate its capabilities. By leveraging modern web technologies, CookBook provides an interactive and dynamic platform, making recipe discovery and meal planning effortless.