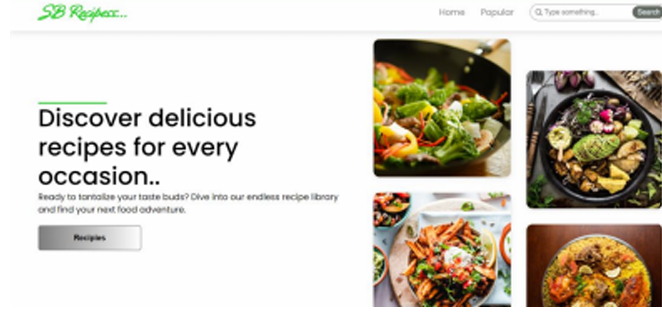
**Frontend Development with React.js**

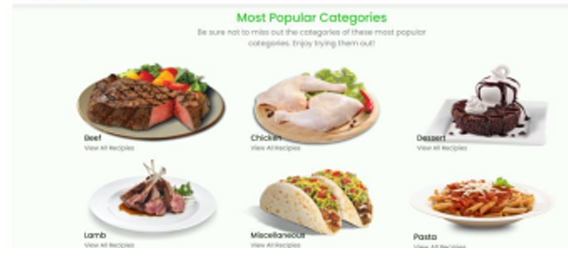
**Project Documentation format**

1. **Introduction**
   * **Project Title**: [Virtual Kitchen Assistant]
   * **Team Members**: Deepak S(212201327), Hariharan G(212201343), Mutharasan K(212201373), JafarSathik A(212201351), Chinmaya Pradhan R (212201432).
2. **Project Overview**
   * **Purpose**: Welcome to the forefront of culinary exploration with CookBook! Our cutting-edge web application is meticulously crafted to transcend the boundaries of culinary experiences, catering to the tastes of both passionate cooking enthusiasts, and seasoned professional chefs.

* + Features: 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. **Architecture**
   * **Component Structure**: 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.
   * **State Management**: The method is chained to the axios.get call to handle a successful response from the API. Inside the 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.
   * **Routing**: Setup the clear routing paths to access various files in the application.
   * Develop the Navbar and Hero components
2. **Setup Instructions**
   * **Prerequisites**: 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.
   * **Installation**: To build Cook Book, we'll need a developer's toolkit. We'll use React.js for the interactive interface, React Router Dom for seamless navigation, and Axios to fetch news data. For visual design, we'll choose either Bootstrap or Tailwind CSS for pre-built styles and icons. Open the project folder to install necessary tools, In this project, we use:
   * React JS
   * React Router Dom
   * React Icons
   * Bootstrap/tailwind CSS
   * Axios
3. **Folder Structure**
   * **Client**: 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
   * **Utilities**: 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 [].
4. **Running the Application**
   * Provide commands to start the frontend server locally.
     + **Frontend**: npm start in the client directory.
5. **Component Documentation**
   * **Key Components**: Document major components are Hero Components.
   * **Reusable Components**: The reusable components are Hero Components and Recipe Components.
6. **State Management**
   * **Global State**: The flow of this application is the user can search the required0 recipe and the user can watch the video of that recipe
   * **Local State**: The Hero components are manage the visual snips of this Application.
7. **User Interface**





1. **Styling**

* **CSS Frameworks/Libraries**: we have used CSS for styling the Application

1. **Testing**

* **Testing Strategy**: Testing like text generating, libraries, image generating, You tube video link generating.
* **Code Coverage**: Explain any tools or techniques used for ensuring adequate test coverage.

1. **Screenshots or Demo**

* Demo link: <https://drive.google.com/file/d/1khMJkccySgKyqRaEZgCpgDACHi572Llj/view?usp=sharing>

1. **Known Issues**

* Document any known bugs or issues that users or developers should be aware of.

1. **Future Enhancements**

* Outline potential future features or improvements, such as new components, animations, or enhanced styling.