**Project Documentation**

CookBook:Your Virtual Kitchen Assistant

**1. Introduction**

* **Project Title:** CookBook:Your Virtual Kitchen Assistant
* **Team id:**NM2025TMID37988
* **Team leader:** 
  1. **Name:**S.Subhalakshmi
  2. **Email id:**sg.subhalakshmi@gmail.com
* **Team Members:**
  1. **Name :**G.Divya

**Email id:**divs9327@gmail.com

* 1. **Name :**P.Divya

**Email id:**divyadiv0608@gmail.com

* 1. **Name :**S.Sirajun nisha

**Email id:**sirajunnisha2007@gamil.com

**2. Project Overview**

**1.Purpose:**

The project is a recipe browsing web application where users can search for meals, explore categories, and view detailed recipes with ingredients and cooking instructions.

**2.Features:**

* Homepage with a hero section and categories.
* Search bar to find recipes by name.
* Browse popular recipe categories.
* View trending Indian dishes.
* Category-specific pages with meal listings.
* Recipe detail page with ingredients, instructions, and YouTube video tutorials.
* Newsletter subscription feature.
* Footer with quick navigation links.

**3. Architecture**

**Component Structure:**

* Hero.jsx → Displays homepage banner with text and image.
* CategoriesHome.jsx → Shows popular categories and trending dishes.
* Category.jsx (page) → Lists meals under a selected category.
* Recipie.jsx (page) → Displays detailed recipe info (image, procedure, ingredients, video).
* Navbar.jsx → Navigation bar with search functionality.
* Footer.jsx → Footer with category links.
* NewsLetter.jsx → Newsletter subscription form.Home.jsx (page) → Combines Hero, CategoriesHome, and Newsletter.
* About.jsx → About page (basic).

**1.State Management:**

Local component state using React.useState and useEffect.

No global state management (like Redux or Context API) is used.

**2.Routing:**

react-router-dom is used for navigation.

**3.Routes include:**

/ → Homepage

/category/:id → Category page

/recipie/:id → Recipe detail page

/contact → Contact page

/about → About page

**4. Setup Instructions**

**Prerequisites:**

Node.js (latest stable version)

npm (comes with Node.js)

**5.Installation:**

**1. Clone the repository**

git clone - https://github.com/sirajunnisha2007-commits/my-first.git

cd project-folder

**2**. **Install dependencies:**

npm install

**3.** **Start development server:**

npm start

**5. Folder Structure**

src/

│── components/ # Reusable UI components (Hero, Navbar, Footer, Newsletter, etc.)

│── pages/ # Page components (Home, Category, Recipie, About)

│── styles/ # CSS files for styling

│── images/ # Image assets

│── App.js # Main app entry point

│── index.js # React DOM rendering

**Utilities:**

axios is used for making API calls to TheMealDB API.

No custom hooks or utility functions added yet.

**6. Running the Application**

Frontend Command:

npm start

Runs the app locally at http://localhost:3000

**7. Component Documentation**

* Navbar.jsx → Contains site title, search bar, and navigation links. Accepts no props.
* Hero.jsx → Static component showing banner text and an image.
* CategoriesHome.jsx → Fetches and displays categories and trending dishes. Uses API calls.
* Category.jsx → Fetches recipes based on category. Uses useParams.
* Recipie.jsx → Fetches detailed recipe by ID, shows ingredients, instructions, and video.
* Footer.jsx → Displays quick category links and copyright.
* NewsLetter.jsx → Allows user to subscribe with an email.

**8. State Management**

* Global State: Not used.
* Local State:
* Used in components with useState. Example:
* Navbar.jsx → stores search input.
* CategoriesHome.jsx → stores categories and items.
* Recipie.jsx → stores selected recipe.
* NewsLetter.jsx → stores email and subscription status.

**9. User Interface**

**1.Homepage:**

Hero section with image + text.

Popular categories grid.

Trending dishes display.

**2.Category Page:**

Grid of meals filtered by category.

**3.Recipe Page:**

Recipe image, procedure, ingredients list, and YouTube video

**4.Newsletter Section:**

Email input with subscription success message.

**5.Footer:**

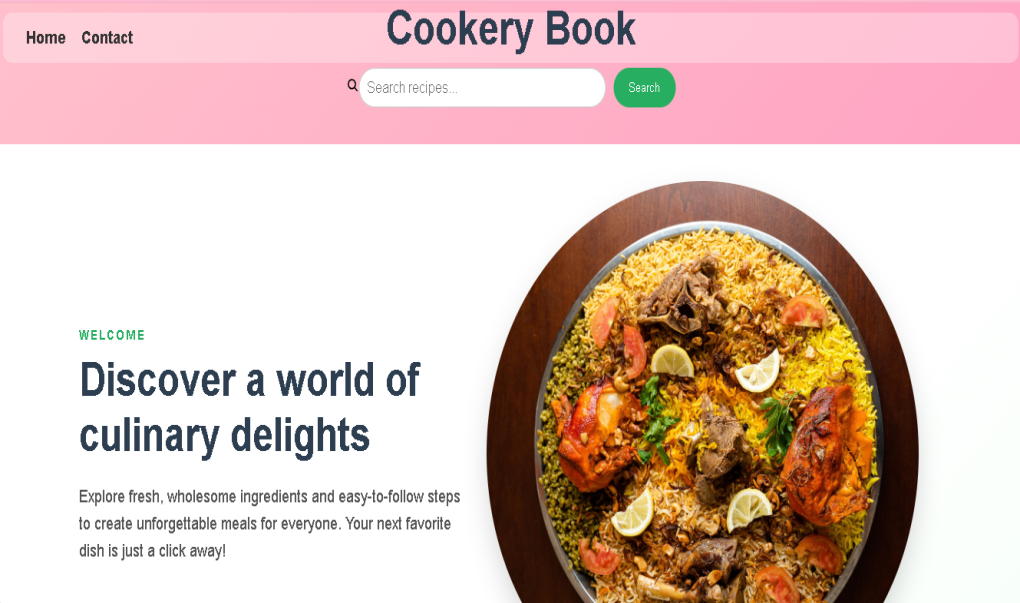
Quick links to categories.

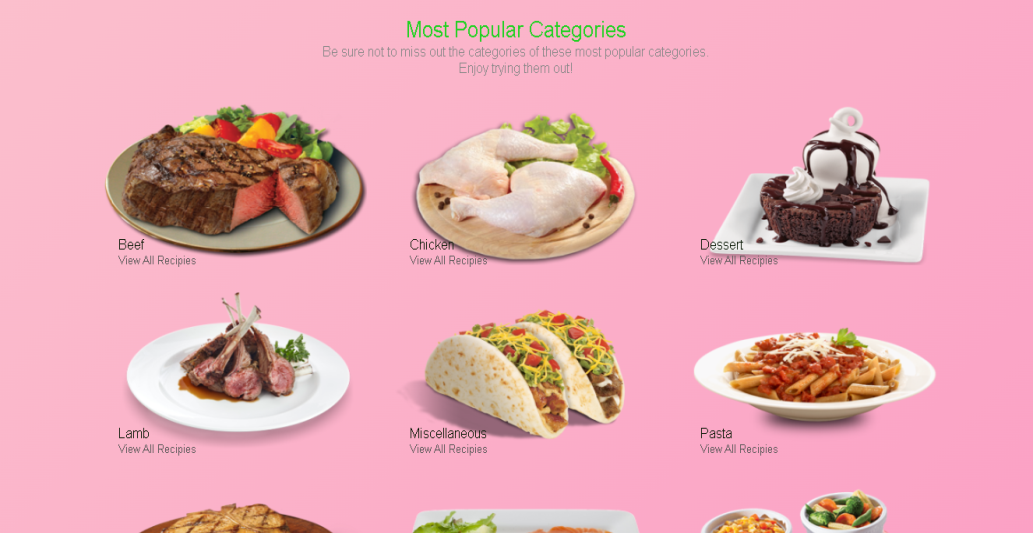
**10. Styling**

CSS Frameworks/Libraries: Plain CSS (modular styles inside styles/ folder).

Theming: No theme provider; custom color palette is hardcoded in CSS (green and neutral tones).

**11. Screenshots**







**12. Known Issues**

* API calls may fail if TheMealDB API is down.
* Some recipes may have missing YouTube links or ingredients.
* No error boundary for broken routes

**13. Future Enhancements**

* Add favorites / saved recipes feature.
* Improve search results (show multiple results instead of first one).
* Add loading spinners while fetching data.
* Implement global state management (Context API or Redux).
* Improve mobile responsiveness with better CSS.

|  |  |
| --- | --- |
|  |  |