

# COOKBOOK: YOUR VIRTUAL KITCHEN ASSISTANT

# INTRODUCTION

- **PROJECT TITLE:** COOKBOOK: YOUR VIRTUAL KITCHEN ASSISTANT
- **TEAM ID:** NM2025TMID39879
- **TEAM LEADER:** NIRANCHANA R & niranchanarajamani@gmail.com
- **TEAM MEMBERS:**
  - VIDYA S & svidhyavidhya507@gmail.com
  - ARCHANA DEVI A & achuarchana51504@gmail.com
  - DHARSHINI A & ddharshiniarulselvam@gmail.com
- **INTRODUCTION ABOUT COOKBOOK YOUR VIRTUAL KITCHEN ASSISTANT**

Welcome to the world of culinary exploration with CookBook: Your Virtual Kitchen Assistant. This innovative recipe app is designed to revolutionize the way you interact with cooking, recipes, and meal planning. Whether you are a beginner in the kitchen or a seasoned chef, CookBook provides an intuitive and user-friendly platform that makes cooking fun, interactive, and efficient

With features like advanced search, recipe categories, visual galleries, and integrated video tutorials, CookBook ensures that every cooking session becomes a joyful experience. Our goal is not just to provide recipes but to build a vibrant community of food lovers who share their passion for good food and healthy living.

CookBook bridges the gap between technology and cooking by offering a seamless platform that caters to diverse tastes, dietary preferences, and skill levels. Every tap brings you closer to

discovering new cuisines, trying out exciting recipes, and building a healthier lifestyle.

## FEATURES OF COOKBOOK:

- Recipes from API/Database: Access thousands of recipes across cuisines and dietary preferences.
- Visual Recipe Exploration: Explore through curated images and galleries.
- Advanced Search: Search by ingredient, cuisine, or dietary restriction. Recipe Details: Step-by-step instructions, ingredients list, cooking time, and nutrition info.
- Video Integration: Embedded YouTube tutorials for better cooking guidance.
- Responsive Design: Optimized for desktops, tablets, and smart phones.

## PROJECT GOALS AND OBJECTIVES:

- The primary aim of CookBook is to provide an accessible and comprehensive platform for food enthusiasts, home cooks, and professional chefs.
- Key Objectives:
  - User-Friendly Experience: Develop a clean and intuitive interface to search, save, and share recipes.
  - Comprehensive Recipe Management: Organize recipes by categories, ingredients, and cuisines with advanced filtering.
  - Technology Stack: Utilize modern web technologies (React.js, Axios, Tailwind/Bootstrap) to ensure smooth performance.

- Community Building: Enable users to share their favorite recipes and tips.

## 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.

### ✓ Version Control:

Use Git for version control, enabling collaboration and tracking changes throughout the development process

Platforms like GitHub or Bitbucket can host your repository.

- Git: Download and installation instructions can be found at: <https://git-scm.com/downloads>

### ✓ 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>
- WebStorm: Download from <https://www.jetbrains.com/webstorm/download/?section=windows>

TO GET THE APPLICATION PROJECT FROM DRIVE:  
FOLLOW BELOW STEPS:

✓ Get the code:

Download the code from the drive link given below:

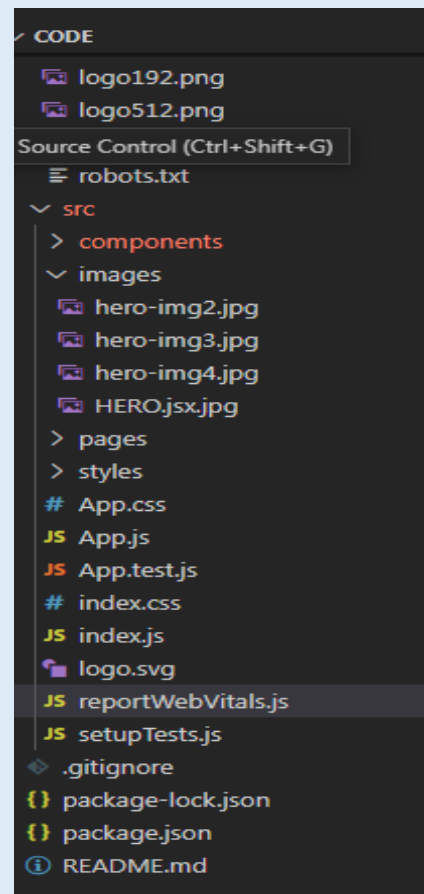
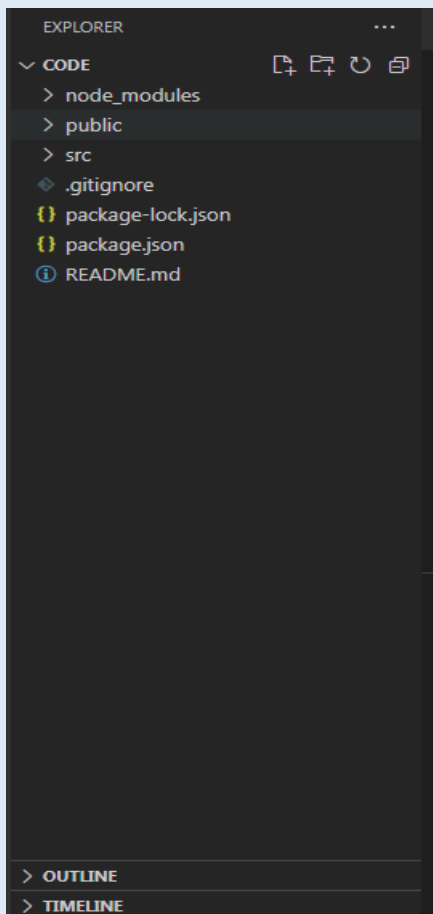
[https://drive.google.com/drive/u/0/folders/1u8PnV\\_mE0mwKkH\\_CvuNpliZtRLJZMqrO](https://drive.google.com/drive/u/0/folders/1u8PnV_mE0mwKkH_CvuNpliZtRLJZMqrO)

INSTALL DEPENDENCIES:

- Navigate into the cloned repository directory and install libraries: `cd cookbook-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 application'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.

### Project 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 URLs 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.

### **Milestone 1: Project setup and configuration.**

- Installation of required tools:
  1. Open the project folder to install necessary tools

In this project, we use:

- React Js
- React Router Dom
- React Icons
- Bootstrap/tailwind css
- Axios
- For further reference, use the following resources
  - o <https://react.dev/learn/installation>
  - o <https://react-bootstrap-v4.netlify.app/getting-started/introduction/>
  - o <https://axios-http.com/docs/intro>
  - o <https://reactrouter.com/en/main/start/tutorial>

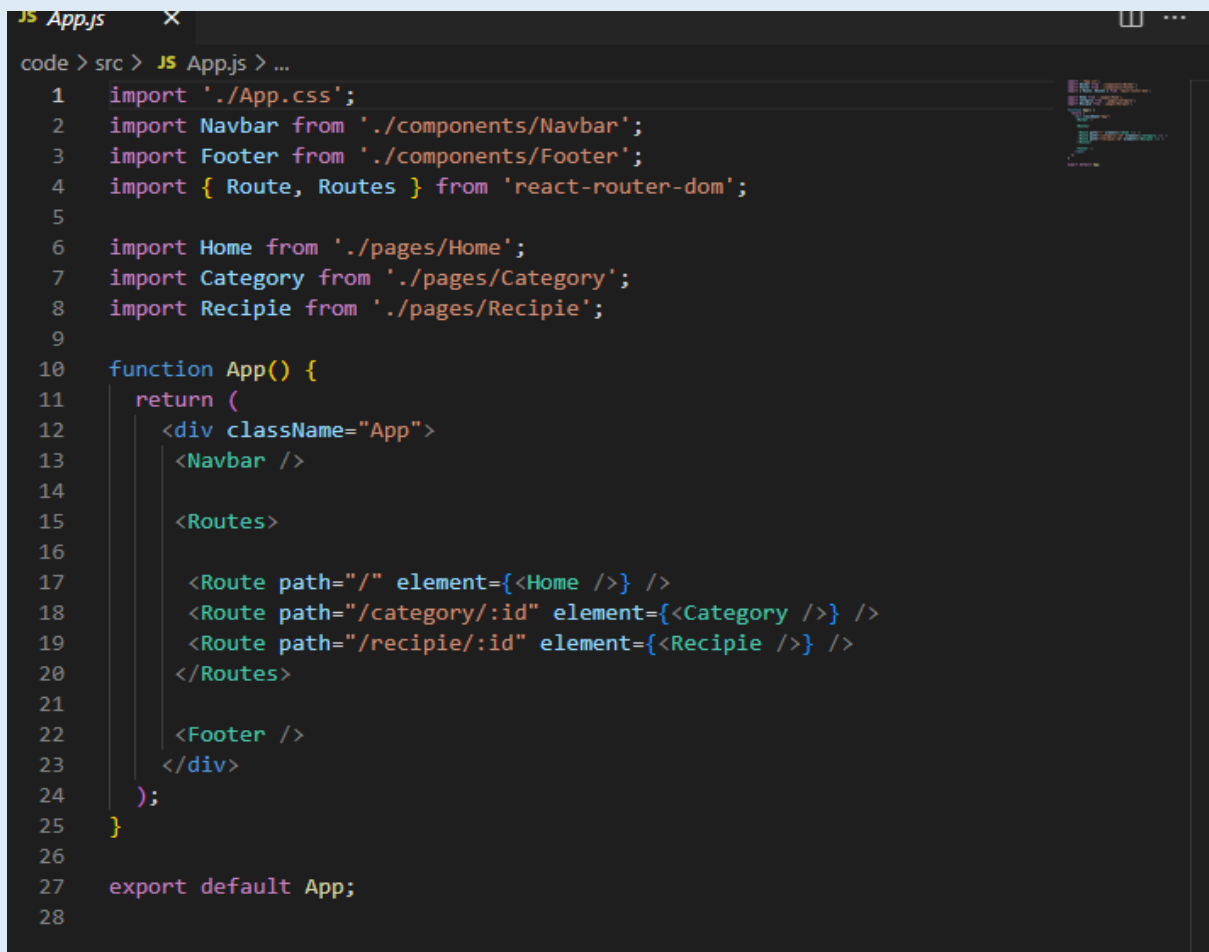


## Milestone 2: Project Development

### ❖ Setup the Routing paths

Setup the clear routing paths to access various files in the application.

Ex:

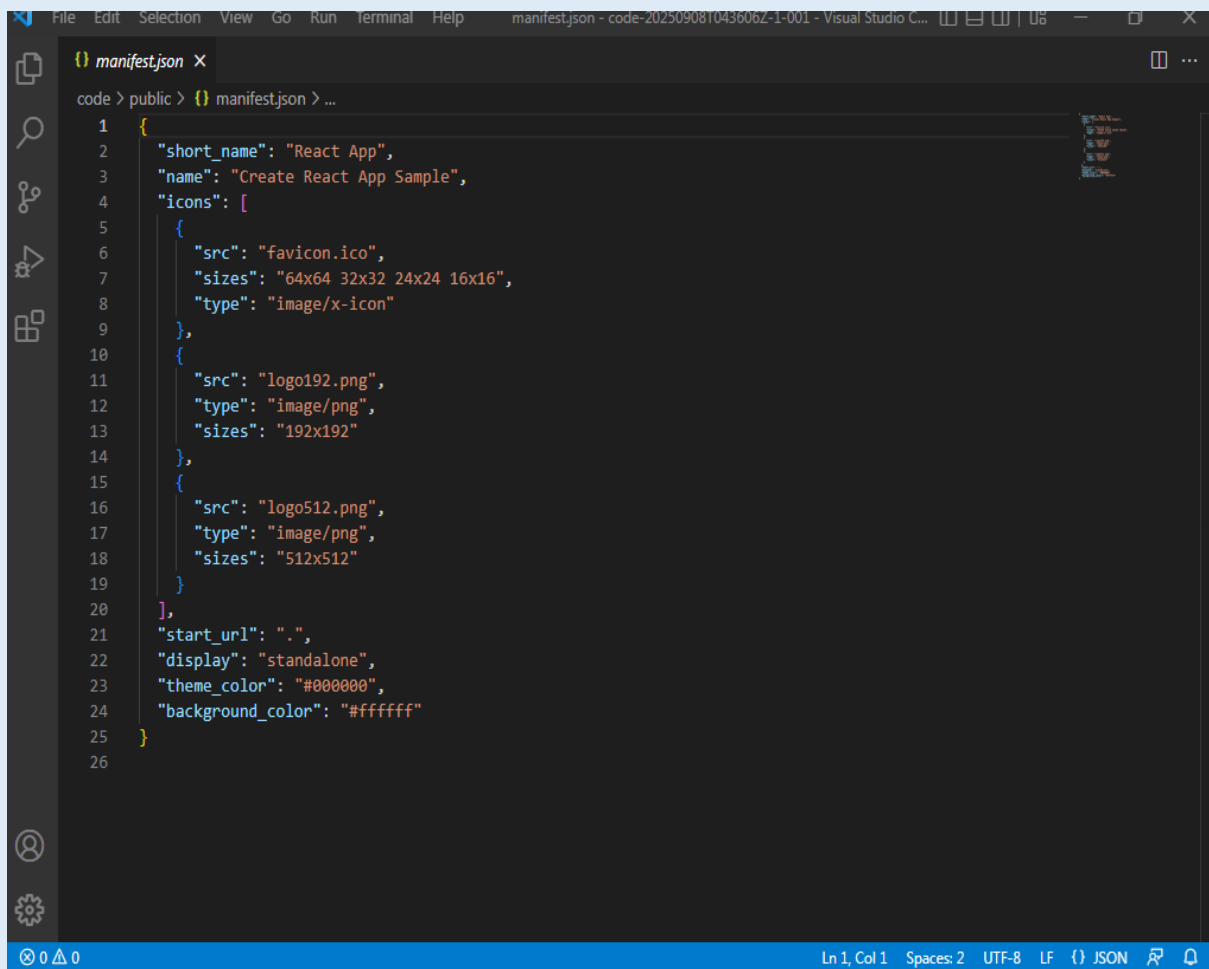


```
JS App.js
code > src > JS App.js > ...
1  import './App.css';
2  import Navbar from './components/Navbar';
3  import Footer from './components/Footer';
4  import { Route, Routes } from 'react-router-dom';
5
6  import Home from './pages/Home';
7  import Category from './pages/Category';
8  import Recipie from './pages/Recipie';
9
10 function App() {
11   return (
12     <div className="App">
13       <Navbar />
14
15       <Routes>
16
17         <Route path="/" element={<Home />} />
18         <Route path="/category/:id" element={<Category />} />
19         <Route path="/recipie/:id" element={<Recipie />} />
20       </Routes>
21
22       <Footer />
23     </div>
24   );
25 }
26
27 export default App;
28
```

- ❖ Develop the Navbar and Hero components
- ❖ Code the popular search/categories components and fetch the categories from *rapid Api*.
- ❖ Additionally, we can add the component to subscribe for the newsletter and the footer.
- ❖ Now, develop the category page to display various exercises under the category.

❖ Finally, code the exercise page, where the instructions, other details along with related videos from the YouTube will be displayed.

## FONT AND THEME CODE



```
1 {
2   "short_name": "React App",
3   "name": "Create React App Sample",
4   "icons": [
5     {
6       "src": "favicon.ico",
7       "sizes": "64x64 32x32 24x24 16x16",
8       "type": "image/x-icon"
9     },
10    {
11      "src": "logo192.png",
12      "type": "image/png",
13      "sizes": "192x192"
14    },
15    {
16      "src": "logo512.png",
17      "type": "image/png",
18      "sizes": "512x512"
19    }
20  ],
21   "start_url": ".",
22   "display": "standalone",
23   "theme_color": "#000000",
24   "background_color": "#ffffff"
25 }
26
```

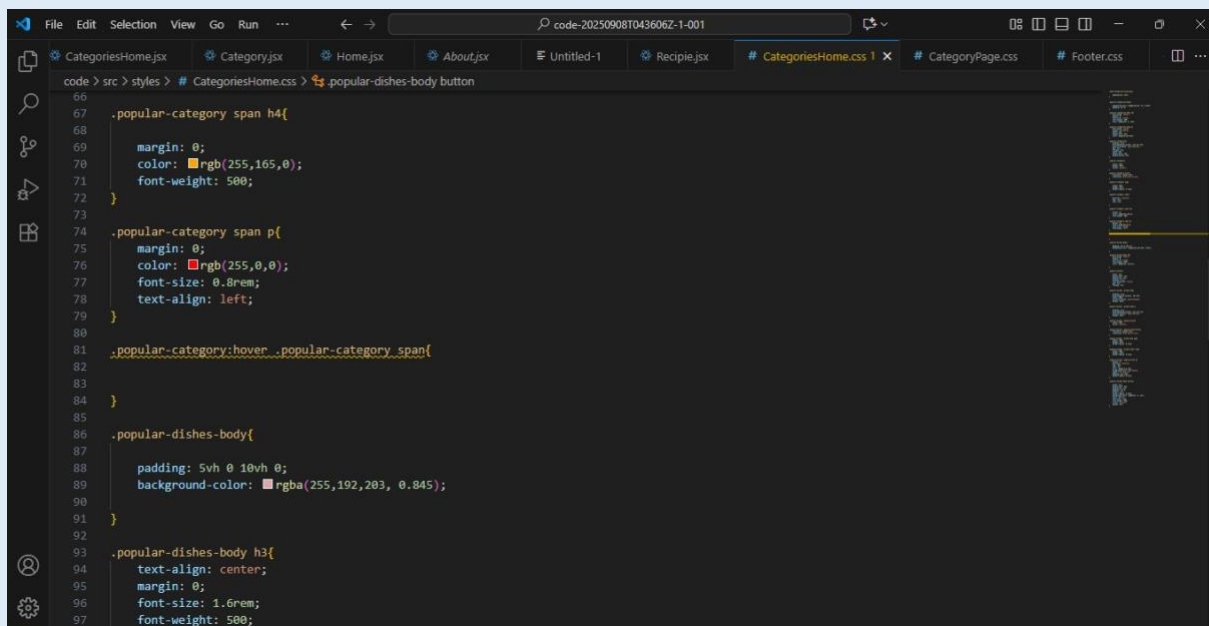
- **Brute Force Algorithm** – Tries all possible solutions until the correct one is found. Example: Linear Search.
- **Divide and Conquer Algorithm** – Breaks the problem into smaller sub-problems, solves them, then combines the results. Example: Merge Sort, Binary Search.

- **Greedy Algorithm** – Makes the best local choice at each step, hoping it leads to the global solution. Example: Dijkstra's Algorithm.
- **Dynamic Programming Algorithm** – Solves problems by storing and reusing solutions of sub-problems. Example: Fibonacci, Knapsack.
- **Backtracking Algorithm** – Tries solutions step by step and goes back if the current path fails. Example: N-Queens Problem.
- **Randomized Algorithm** – Uses random numbers to make decisions or improve efficiency. Example: Randomized Quick Sort.
- **Recursive Algorithm** – A function calls itself to solve smaller parts of the same problem. Example: Factorial, Tower of Hanoi.
- **Searching Algorithms** – Used to find an element in a data set. Examples: Linear Search, Binary Search.
- **Sorting Algorithms** – Used to arrange data in a specific order. Examples: Bubble Sort, Quick Sort, Merge Sort

## HERO PGH CODE

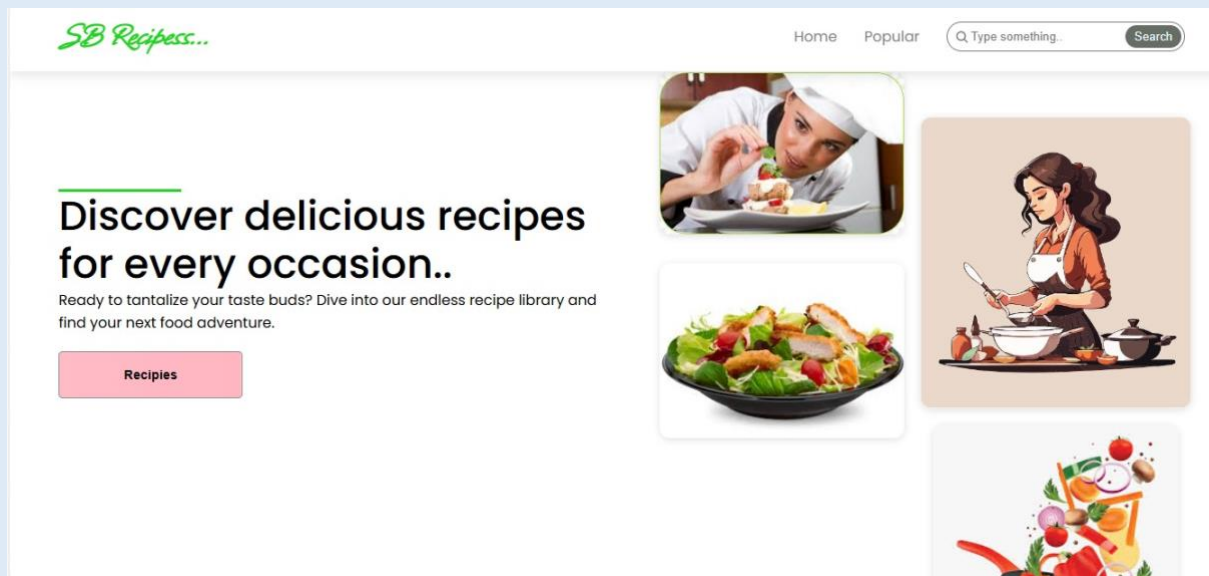
```
File Edit Selection View Go Run Terminal Help Hero.jsx - code-20250908T043606Z-1-001 - Visual Studio Code
Hero.jsx x
code > src > components > Hero.jsx > ...
1 import React from 'react'
2 import '../styles/Hero.css'
3 import heroImg1 from '../images/hero-img1.png'
4 import heroImg2 from '../images/hero-img2.png'
5 import heroImg3 from '../images/hero-img3.png'
6 import heroImg4 from '../images/hero-img4.png'
7
8 const Hero = () => {
9   return (
10     <div className='hero-container'>
11       <div className="hero-text">
12         <div className="hero-line" />
13         <h1>Discover delicious recipes for every occasion.</h1>
14         <p>Ready to tantalize your taste buds? Dive into our endless recipe library and find you
15         <a href="#recipies"><button>Recipies</button></a>
16       </div>
17       <div className="hero-images">
18         <span className='span1'>
19           <img src={heroImg2} alt="" />
20           <img src={heroImg4} alt="" />
21         </span>
22         <span className='span2'>
23           <img src={heroImg3} alt="" />
24           <img src={heroImg1} alt="" />
25         </span>
26       </div>
27     </div>
28   )
29 }
30
31 export default Hero
```

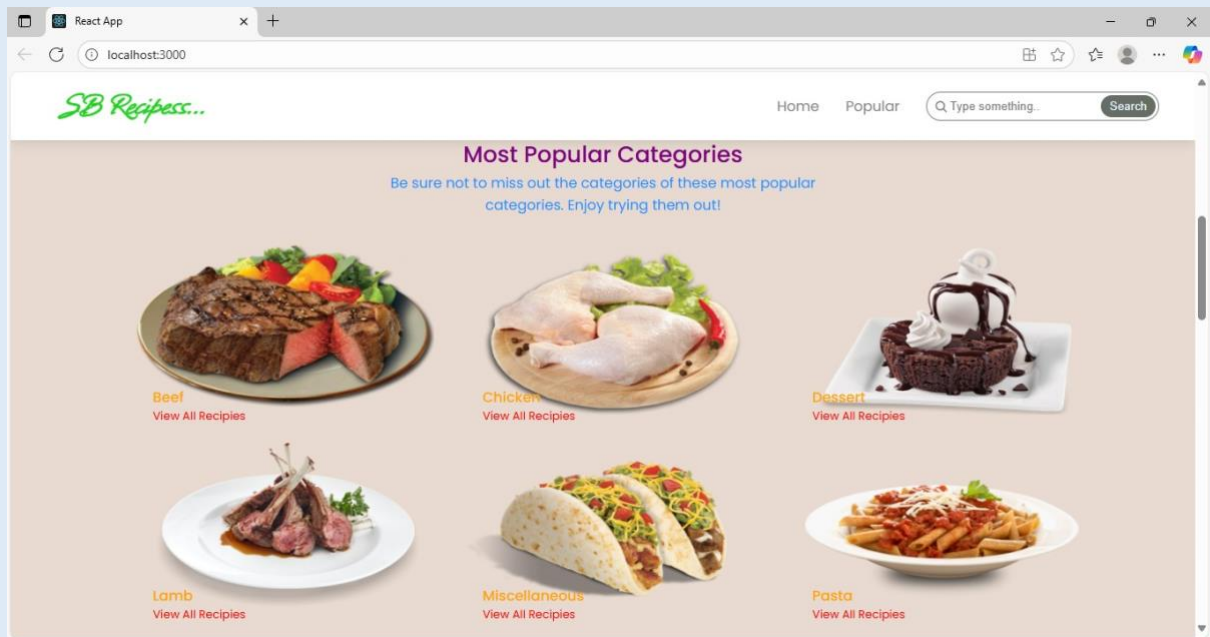
# COOKBOOK: YOUR VIRTUAL KITCHEN ASSISTANT



```
code > src > styles > # CategoriesHome.css > .popular-dishes-body button
66
67 .popular-category span h4{
68
69   margin: 0;
70   color: rgb(255,165,0);
71   font-weight: 500;
72 }
73
74 .popular-category span p{
75   margin: 0;
76   color: rgb(255,0,0);
77   font-size: 0.8rem;
78   text-align: left;
79 }
80
81 .popular-category:hover .popular-category span{
82
83 }
84
85
86 .popular-dishes-body{
87
88   padding: 5vh 0 10vh 0;
89   background-color: rgba(255,192,203, 0.845);
90 }
91
92
93 .popular-dishes-body h3{
94   text-align: center;
95   margin: 0;
96   font-size: 1.6rem;
97   font-weight: 500;
```

## AFTER CHANGES





\*\*\*\*\*

THANK YOU