PROJECT DOCUMENTATION

INTRODUCTION

Project Title: CookBook - Your Virtual Kitchen Assistant

TEAM MEMBERS	EMAIL ID
Harini.E	ajithsriram01@gmail.com
keerthana.B	bkeerthana989@gmail.com
Akshaya.R	Akshayakrachu@gmail.com
Divya.S	divyasdivyas34@gmail.com

PROJECT OVERVIEW:

CookBook is an innovative web application designed for users to explore, organize, and create recipes effortlessly. It caters to both beginner and professional chefs, providing an intuitive user experience and a vast collection of diverse recipes.

PURPOSE:

CookBook aims to revolutionize the way users interact with recipes by offering a seamless platform for discovering, saving, and sharing culinary inspirations. The main goals are:

- **User-Friendly Experience** – Easy navigation for discovering and managing recipes.
- **Comprehensive Recipe Management** – Advanced search and categorization for efficient organization.
- **Modern Tech Stack** – Utilizing React.js and Rapid API for enhanced functionality.

FEATURES:

- **Recipe API Integration** – Fetches meals from the MealsDB API.
- **Visual Recipe Browsing** – Image-based navigation of categories.
- **Search Functionality** – Easily find recipes using keywords.
- **Interactive UI** – Built using modern design principles for a smooth experience.

ARCHITECTURE

Component Structure

The application is divided into three main sections:

- **Pages** – Full-page components (Home, Category, Recipe Details).
- **Components** – Reusable UI elements (Navbar, Search Bar, Recipe Cards).
- **Styles** – CSS and styling files.

State Management

- **Global State:** Managed using React Context API.
- **Local State:** Controlled via React's useState for component-level updates.

Routing

Implemented using React Router to enable seamless navigation between pages.

SETUP INSTRUCTIONS

Prerequisites

- **Node.js & npm** – Install from Node.js website.
- **React.js** – Set up a new project using:

  ```
  npx create-react-app my-react-app
  cd my-react-app
  npm start
  ```

Installation Steps

1. **Clone the repository**

   ```
   git clone [repository URL]
   ```

2. **Navigate into the project directory**

   ```
   cd recipe-app-react
   ```

3. **Install dependencies**

   ```
   npm install
   ```

4. **Set up environment variables** (if required) by creating a .env file and adding necessary API keys.

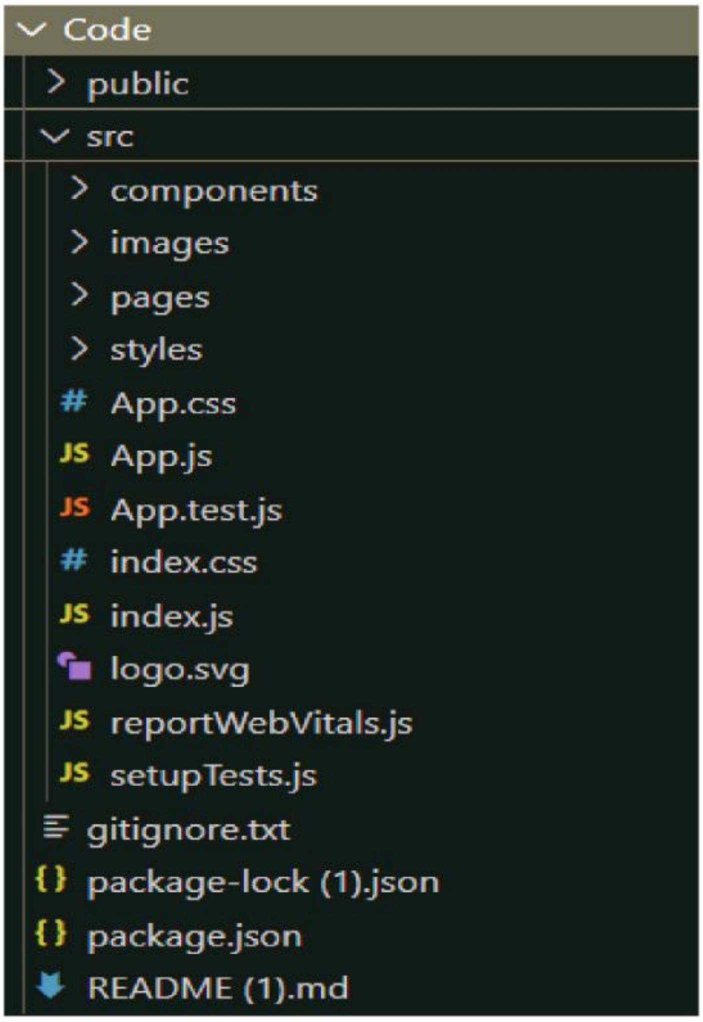
5. **Start the development server**

   npm start

6. **Access the application**
   Open http://localhost:3000 in your web browser.

FOLDER STRUCTURE

The project is structured into different directories for better organization and maintainability. Below is an overview of the folder structure.



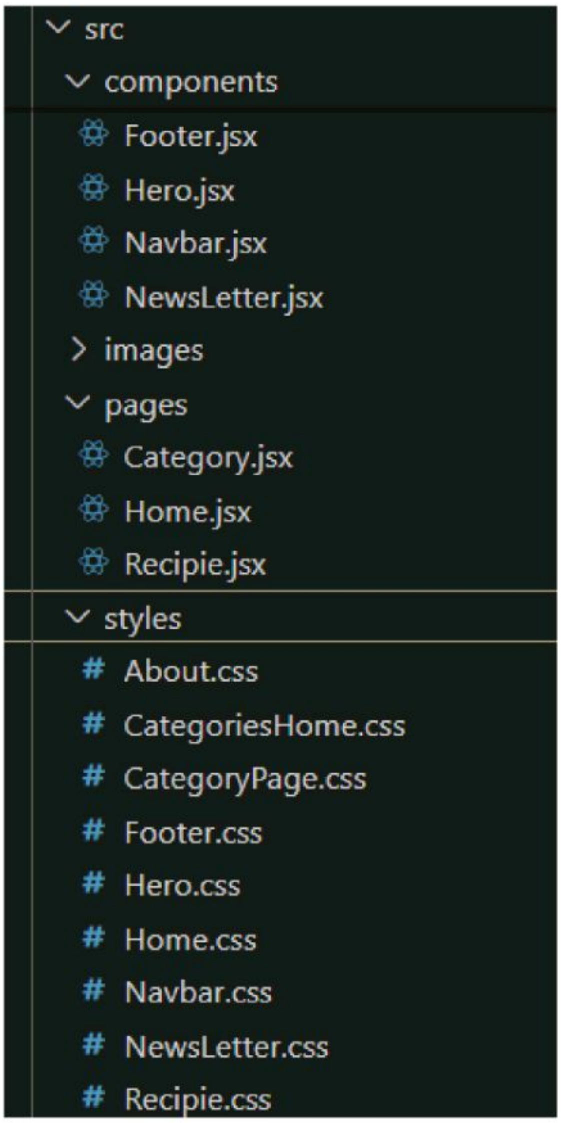
```
Code
  public
  src
    components
    images
    pages
    styles
    App.css
    App.js
    App.test.js
    index.css
    index.js
    logo.svg
    reportWebVitals.js
    setupTests.js
  gitignore.txt
  package-lock (1).json
  package.json
  README (1).md
```

Client

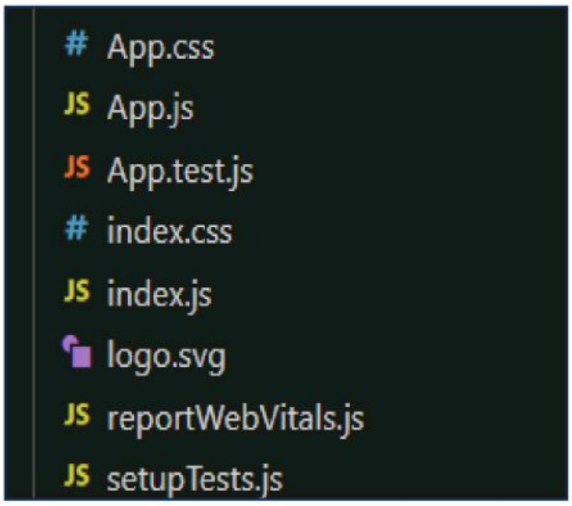
The **client** folder (inside src/) contains the core files of the frontend application. It consists of:

- **Components:** Houses all reusable UI components such as buttons, cards, and the navigation bar.
- **Pages:** Contains full-page components, including the homepage, recipe details page, and category listings.
- **Styles:** Includes all CSS or SCSS files to style the application.
- **Assets:** Stores images, icons, and other static files used throughout the app.

Example structure inside src/:



```
src
  components
    Footer.jsx
    Hero.jsx
    Navbar.jsx
    NewsLetter.jsx
  images
  pages
    Category.jsx
    Home.jsx
    Recipie.jsx
  styles
    About.css
    CategoriesHome.css
    CategoryPage.css
    Footer.css
    Hero.css
    Home.css
    Navbar.css
    NewsLetter.css
    Recipie.css
```



```
App.css
App.js
App.test.js
index.css
index.js
logo.svg
reportWebVitals.js
setupTests.js
```

RUNNING THE APPLICATION

To start the frontend server, run:

```
npm start
```

Then, open http://localhost:3000 in your browser.

COMPONENT DOCUMENTATION

KEY COMPONENTS

Below are the major components of the **CookBook** application, along with their purpose and props:

1. Navbar Component (Navbar.js)

Purpose:

- Provides site-wide navigation.
- Contains links to **Home**, **Categories**, and **Search**.

Props:

- logo (string): Path to the logo image.
- menuItems (array): List of menu items for navigation.

2. Hero Component (Hero.js)

Purpose:

- Displays an introduction to the application.
- Contains a call-to-action button to explore recipes.

Props:

- title (string): Main heading text.
- subtitle (string): Supporting description text.
- buttonText (string): Label for the action button.

3. Recipe Card Component (RecipeCard.js)

Purpose:

- Displays a brief summary of a recipe, including an image, title, and category.
- Redirects users to the detailed recipe page when clicked.

Props:

- recipe (object): Contains id, title, image, and category.

4. Category Component (Category.js)

Purpose:

- Displays different categories of meals.
- Allows users to filter recipes by category.

Props:

- categoryName (string): Name of the category.
- image (string): Category thumbnail.

5. Recipe Details Component (RecipeDetails.js)

Purpose:

- Displays a full recipe, including ingredients, instructions, and a demo video.

Props:

- recipeId (string): Unique ID to fetch the recipe details.

REUSABLE COMPONENTS

1. Search Bar Component (SearchBar.js)

Purpose:

- Allows users to search for recipes using keywords.

Props:

- onSearch (function): Callback function for handling search queries.

2. Button Component (Button.js)

Purpose:

- A reusable button component with customizable styles.

Props:

- text (string): Button label.
- onClick (function): Click event handler.
- variant (string): Defines button styles (e.g., primary, secondary).

3. Loading Spinner Component (Loading.js)

Purpose:

- Displays a loading animation while fetching data.

Props:

- size (string): Size of the spinner (small, medium, large).

STATE MANAGEMENT

State management in the **CookBook** application ensures efficient data handling across different components. The project incorporates both **global state** for shared data and **local state** for component-specific interactions.

GLOBAL STATE

The application uses the React Context API to manage global state, allowing components to access shared data without prop drilling.

Global State Usage in CookBook:

- Recipe Data Storage: Stores recipe categories and details fetched from the MealsDB API.
- Navigation State: Manages active categories and filters for seamless user experience.
- User Preferences: Keeps track of saved or favorite recipes.

How global state flows across the application:

1. The RecipeContext.js file initializes and manages the global state.
2. The RecipeProvider component wraps the application to provide access to the global state.

3. Components like Category.js and RecipeDetails.js consume global state using useContext.
4. Any updates to the global state (e.g., fetching new recipes) automatically reflect across all dependent components.

LOCAL STATE

Local state is managed using the useState hook within individual components. It is used for handling temporary UI interactions that don't need to persist across multiple components.

Local State Usage in CookBook:

- Search Bar: Stores and updates the user's search input dynamically.
- Recipe Page: Keeps track of selected ingredients or active tabs.
- UI Components: Controls modals, dropdown menus, and loading states.

How Local State Works in Components:

1. The component initializes a state variable using useState().
2. State updates dynamically based on user interactions.
3. Changes to local state trigger a re-render of the specific component without affecting others.

USER INTERFACE

The Cook Book application features a modern and intuitive user interface, designed to provide a seamless experience for users exploring and managing recipes. Below are key UI elements along with their descriptions.

1. Home Page (Hero Section & Search)

📌 **Features:**

- A **welcome banner** introducing the app.
- A **search bar** allowing users to quickly find recipes.
- A call-to-action button for exploring trending recipes.

2. Recipe Categories Page

📌 **Features:**

- Displays various recipe categories fetched from the API.
- Each category card includes an image and title.
- Clicking a category leads to the list of dishes under it.

3. **Recipe Details Page**

📌 **Features:**

- Shows recipe name, ingredients, instructions, and a demo video.
- Includes a save to favorites option.
- Responsive layout for mobile and desktop users.

4. Trending Recipes Section

📌 **Features:**

- Displays popular and trending dishes.
- Each dish card includes an image, title, and quick view button.
- Clicking a recipe redirects to the detailed view.

5. Newsletter Subscription Form

📌 **Features:**

- Users can subscribe to receive new recipes via email.
- Clean and simple input field with a submit button.

6. Responsive UI & Mobile View

📌 **Features:**

- The application is fully responsive across different screen sizes.
- Mobile-friendly navigation menu and recipe cards.

STYLING

The Recipe Application is designed with a modern, responsive, and clean aesthetic using industry-standard styling techniques.

1. **CSS Frameworks & Libraries Used**

Bootstrap/Tailwind CSS:

- Bootstrap is a widely used CSS framework that provides a grid system, pre-defined styling components, and responsiveness without writing extensive custom CSS.
- Tailwind CSS is a utility-first framework that allows for highly customized and flexible styling, reducing the need for external stylesheets.
- The combination of these frameworks ensures:
  - ✓ Consistent layouts across all screen sizes (mobile, tablet, desktop).
  - ✓ Reusable components such as buttons, cards, modals, and forms.
  - ✓ Faster development due to pre-built styles.

React Icons:

- Used for UI elements such as search icons, navigation arrows, social media links, and buttons.
- Helps improve user navigation and interaction, making the interface more visually appealing**.**

2. **Theming & Custom Design**

Consistent Color Scheme**:**

- The application follows a uniform color palette for a clean and readable UI
- Colors are chosen to provide good contrast and accessibility.
- Example:
  - Primary Color: Used for buttons and highlights.
  - Secondary Color: Used for backgrounds and supporting elements.
  - Text Color: Optimized for readability.

Dark Mode & Light Mode**:**

- Dark Mode is implemented to provide a comfortable user experience, especially in low-light environments.
- Users can toggle between Light Mode (default) and Dark Mod**e**, reducing eye strain.
- Future Enhancement: Saving user preference in local storage to persist across sessions.

Custom CSS Modules & Styled Components**:**

CSS Modules**:**

- Ensures that styles are scoped to specific components, preventing unwanted global style conflicts.

Styled Components**:**

- Allows writing CSS directly within JavaScript files.
- Improves maintainability by keeping styles within relevant components.
- Enables dynamic styling based on **props** (e.g., different styles for different recipe categories).

3. Animations & Transitions

Smooth UI Interactions**:**

- CSS animations and React libraries (like Framer Motion) are used to create a seamless browsing experience.
- Examples:
    - Hover effects on buttons & images to indicate interactivity.
    - Fade-in animations for content appearing dynamically.
    - Slide transitions when navigating between pages.

Performance Optimization:

- Avoiding excessive animations to keep the app lightweight.
- Using lazy loading for images and content to enhance page load speed.

TESTING

Ensuring the reliability and performance of the application is critical. The testing approach includes multiple testing strategies

1. Testing Strategy

Unit Testing:

- Used **Jest** and **React Testing Library** to test individual components.
- Ensured that components render correctly and respond to user actions appropriately.
- Tested UI components such as buttons, forms, and search functionality to verify their expected behavior.

Integration Testing:

- Verified if API calls fetch data correctly from MealsDB API and display recipes properly.
- Ensured that selecting a category updates the displayed recipe list dynamically.
- Tested data flow between different components to validate overall functionality.

End-to-End Testing:

- Used **Cypress** to simulate real user interactions and validate complete workflows.
- Automated tests covered:

  i. Searching for a recipe and verifying relevant results.
  ii. Navigating from the home page to a specific recipe and checking content accuracy.
  iii. Subscribing to the newsletter and ensuring form validation works properly.

2. Code Coverage:

Jest Coverage Reports:

- Jest generates code coverage reports to ensure all critical components and functions are tested.
- The reports help identify untested parts of the code and improve test coverage.

Testing Goals:

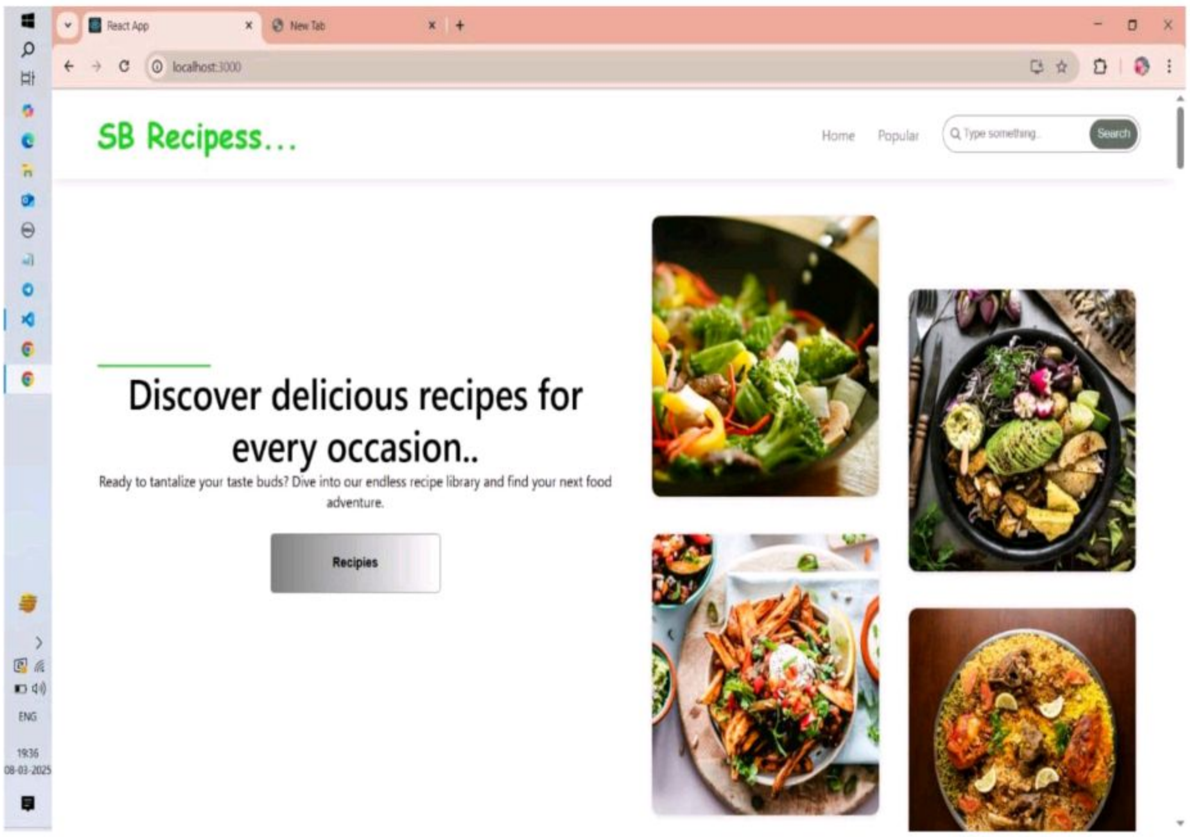
- Achieve **80%+ test coverage**, focusing on critical functionalities such as:

- API calls for fetching recipes.
- UI rendering and user interactions.
- Navigation and form validation.

Continuous Testing:

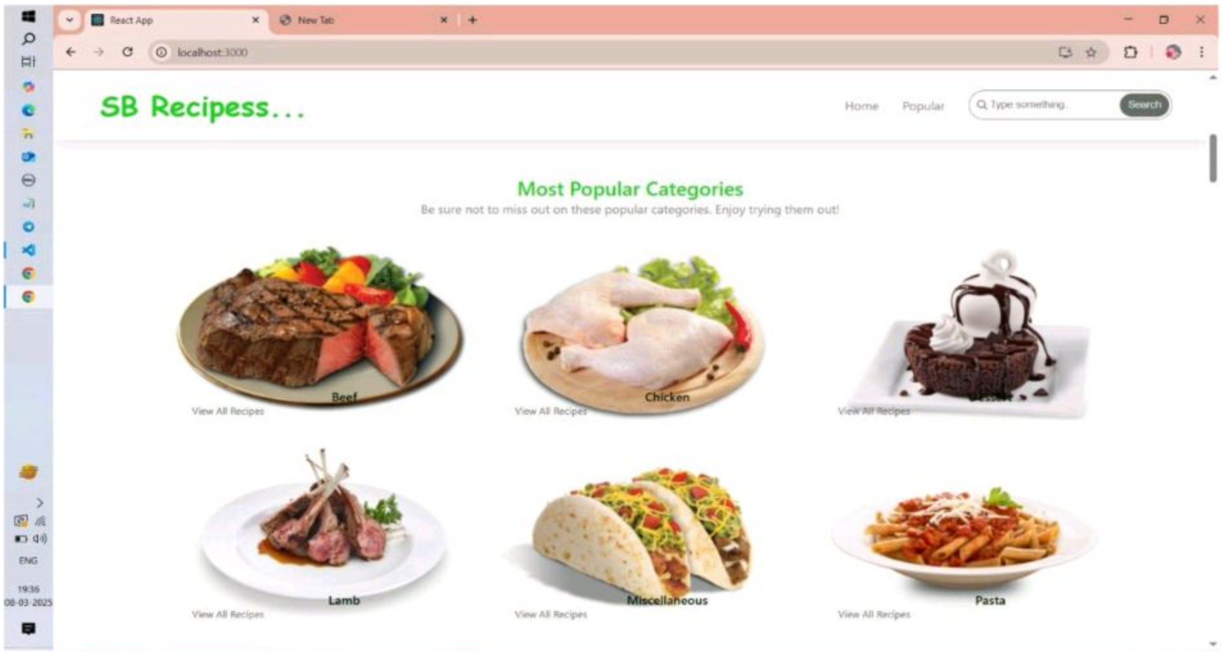
- Implement CI/CD pipelines to run tests automatically before deployment.
- Ensure new features do not break existing functionality.

SCREENSHOTS OR DEMO



Discover delicious recipes for every occasion..

Ready to tantalize your taste buds? Dive into our endless recipe library and find your next food adventure.



SB Recipess...

Most Popular Categories

Be sure not to miss out on these popular categories. Enjoy trying them out!

Beef — View All Recipes

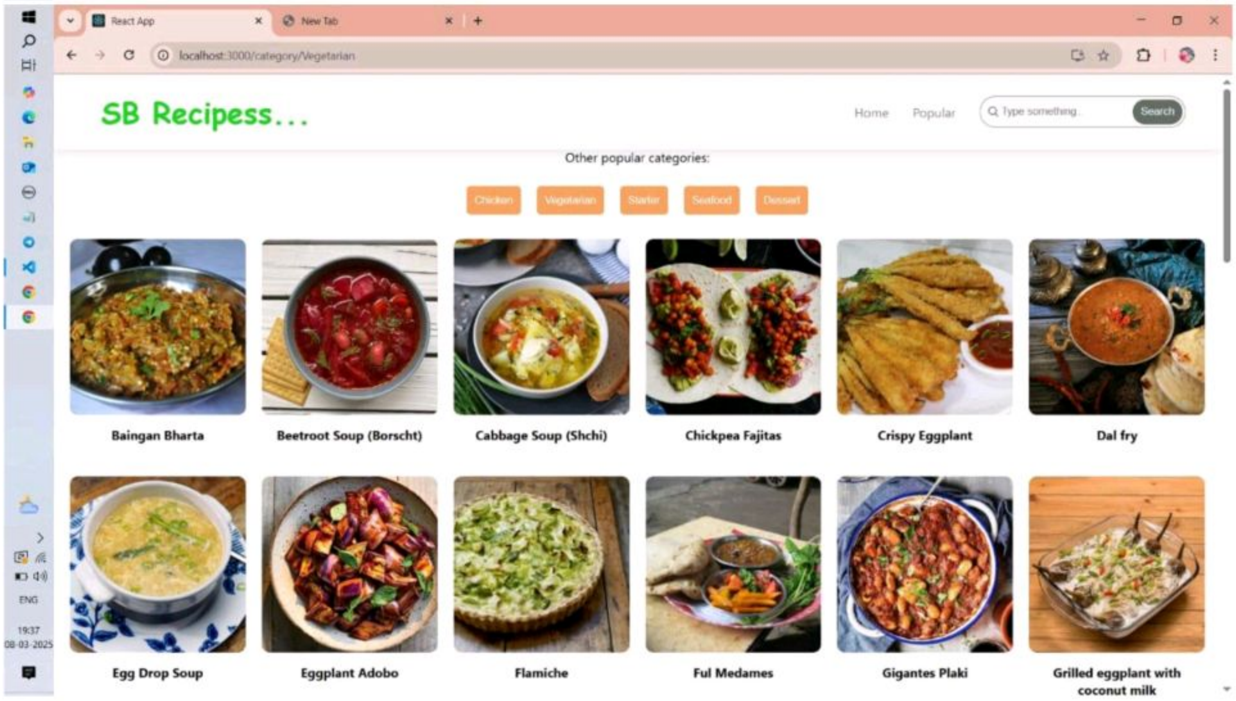
Chicken — View All Recipes

Dessert — View All Recipes

Lamb — View All Recipes

Miscellaneous — View All Recipes

Pasta — View All Recipes



SB Recipess...

Other popular categories:

Chicken Vegetarian Starter Seafood Dessert

Baingan Bharta

Beetroot Soup (Borscht)

Cabbage Soup (Shchi)

Chickpea Fajitas

Crispy Eggplant

Dal fry

Egg Drop Soup

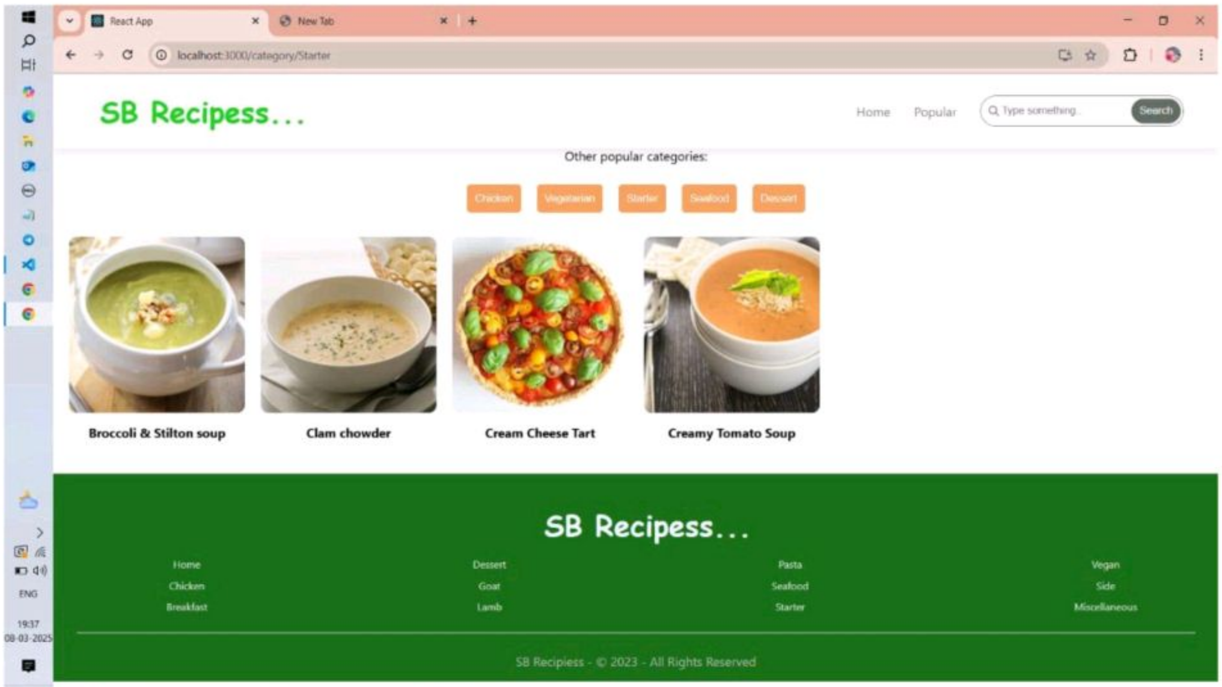
Eggplant Adobo

Flamiche

Ful Medames

Gigantes Plaki

Grilled eggplant with coconut milk



Other popular categories:

Broccoli & Stilton soup

Clam chowder

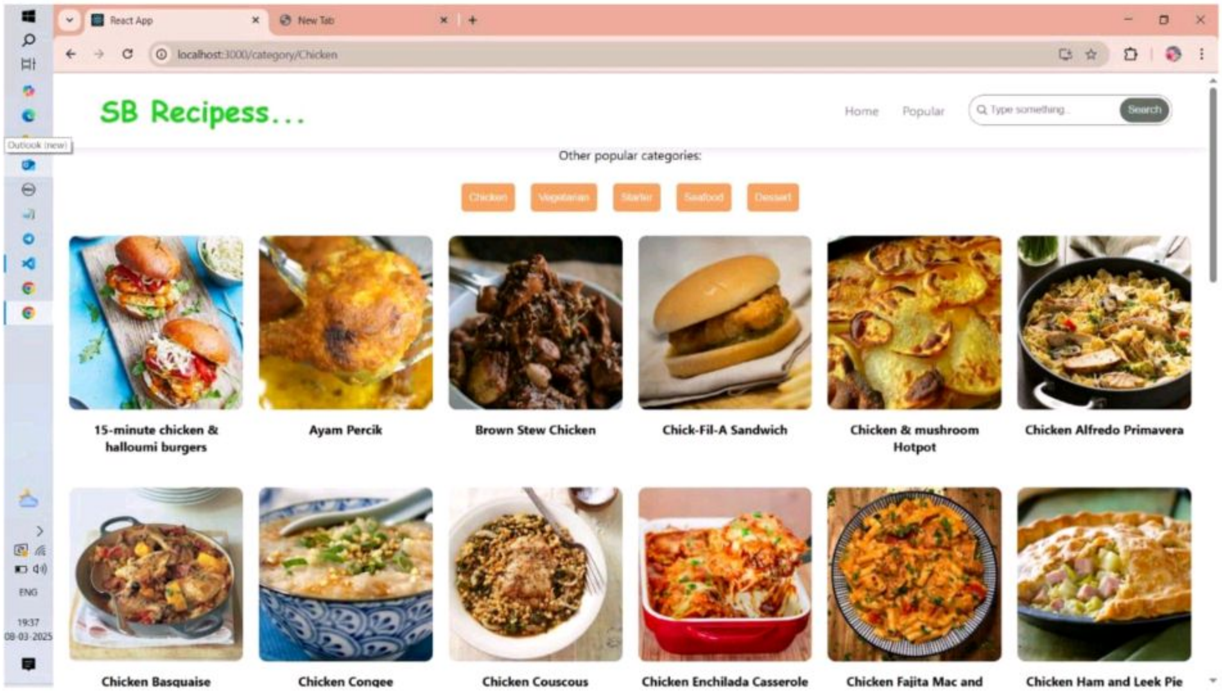
Cream Cheese Tart

Creamy Tomato Soup

SB Recipess...

Home	Dessert	Pasta	Vegan
Chicken	Goat	Seafood	Side
Breakfast	Lamb	Starter	Miscellaneous

SB Recipiess - © 2023 - All Rights Reserved



Other popular categories:

15-minute chicken & halloumi burgers

Ayam Percik

Brown Stew Chicken

Chick-Fil-A Sandwich

Chicken & mushroom Hotpot

Chicken Alfredo Primavera

Chicken Basquaise

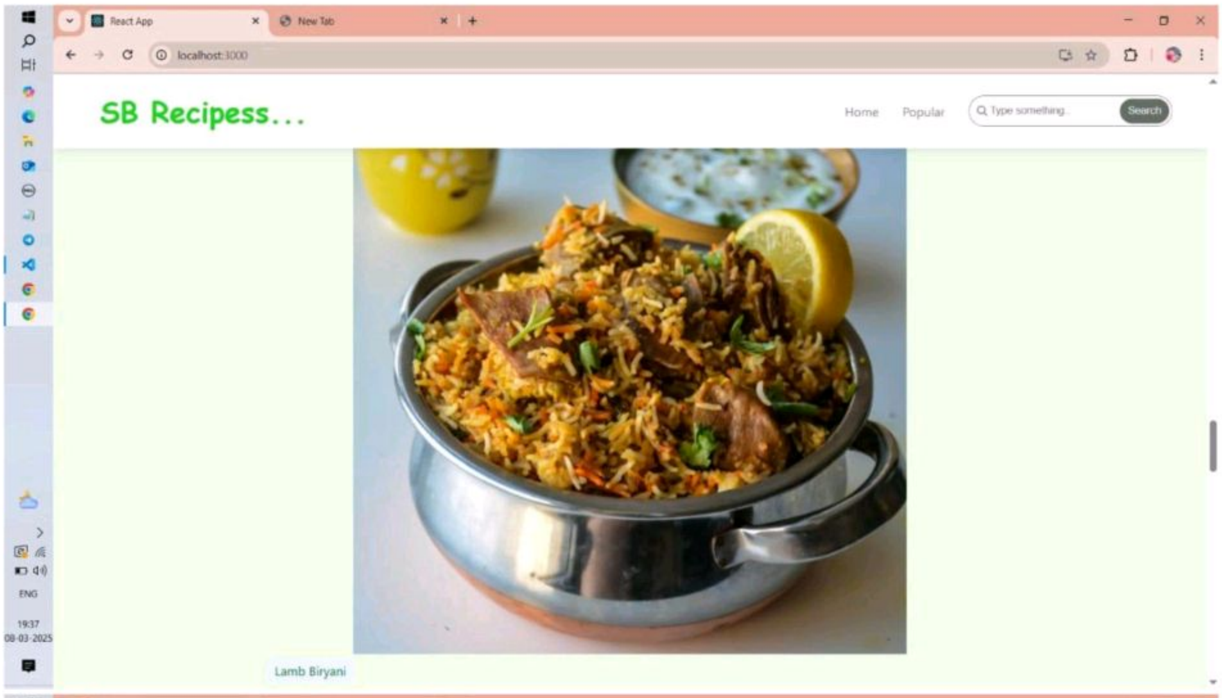
Chicken Congee

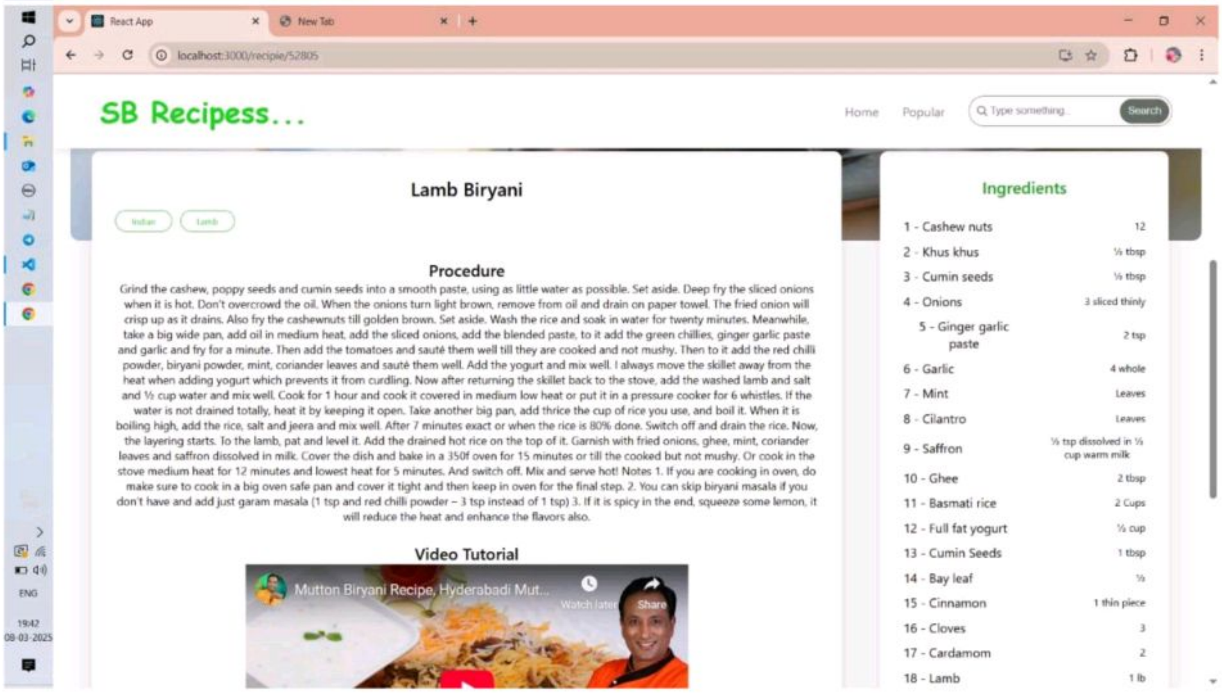
Chicken Couscous

Chicken Enchilada Casserole

Chicken Fajita Mac and

Chicken Ham and Leek Pie





Lamb Biryani

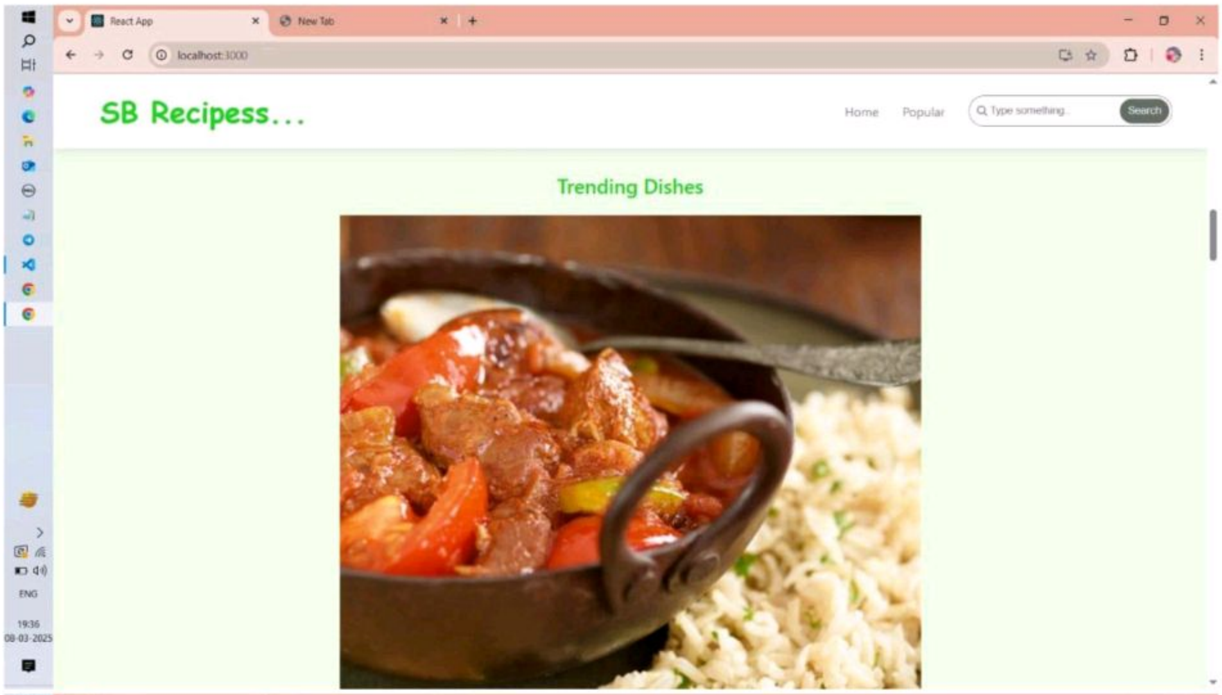
Procedure

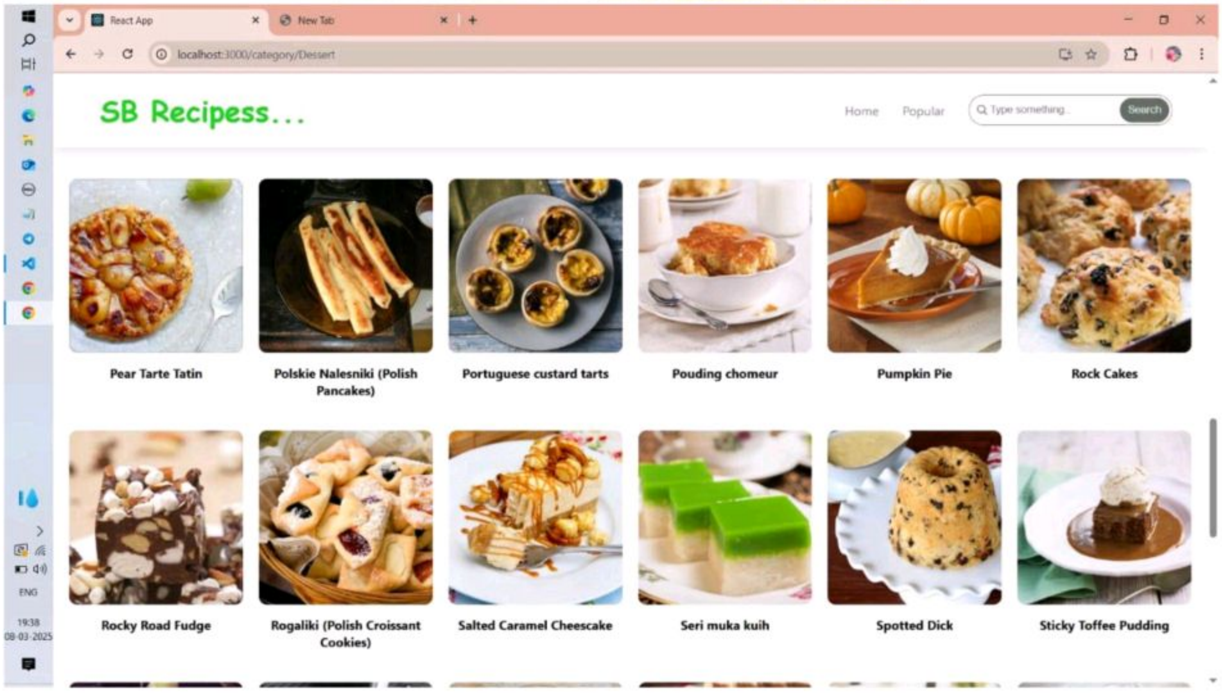
Grind the cashew, poppy seeds and cumin seeds into a smooth paste, using as little water as possible. Set aside. Deep fry the sliced onions when it is hot. Don't overcrowd the oil. When the onions turn light brown, remove from oil and drain on paper towel. The fried onion will crisp up as it drains. Also fry the cashewnuts till golden brown. Set aside. Wash the rice and soak in water for twenty minutes. Meanwhile, take a big wide pan, add oil in medium heat, add the sliced onions, add the blended paste, to it add the green chillies, ginger garlic paste and garlic and fry for a minute. Then add the tomatoes and sauté them well till they are cooked and not mushy. Then to it add the red chilli powder, biryani powder, mint, coriander leaves and sauté them well. Add the yogurt and mix well. I always move the skillet away from the heat when adding yogurt which prevents it from curdling. Now after returning the skillet back to the stove, add the washed lamb and salt and ½ cup water and mix well. Cook for 1 hour and cook it covered in medium low heat or put it in a pressure cooker for 6 whistles. If the water is not drained totally, heat it by keeping it open. Take another big pan, add thrice the cup of rice you use, and boil it. When it is boiling high, add the rice, salt and jeera and mix well. After 7 minutes exact or when the rice is 80% done. Switch off and drain the rice. Now, the layering starts. To the lamb, pat and level it. Add the drained hot rice on the top of it. Garnish with fried onions, ghee, mint, coriander leaves and saffron dissolved in milk. Cover the dish and bake in a 350f oven for 15 minutes or till the cooked but not mushy. Or cook in the stove medium heat for 12 minutes and lowest heat for 5 minutes. And switch off. Mix and serve hot! Notes 1. If you are cooking in oven, do make sure to cook in a big oven safe pan and cover it tight and then keep in oven for the final step. 2. You can skip biryani masala if you don't have and add just garam masala (1 tsp and red chilli powder – 3 tsp instead of 1 tsp) 3. If it is spicy in the end, squeeze some lemon, it will reduce the heat and enhance the flavors also.

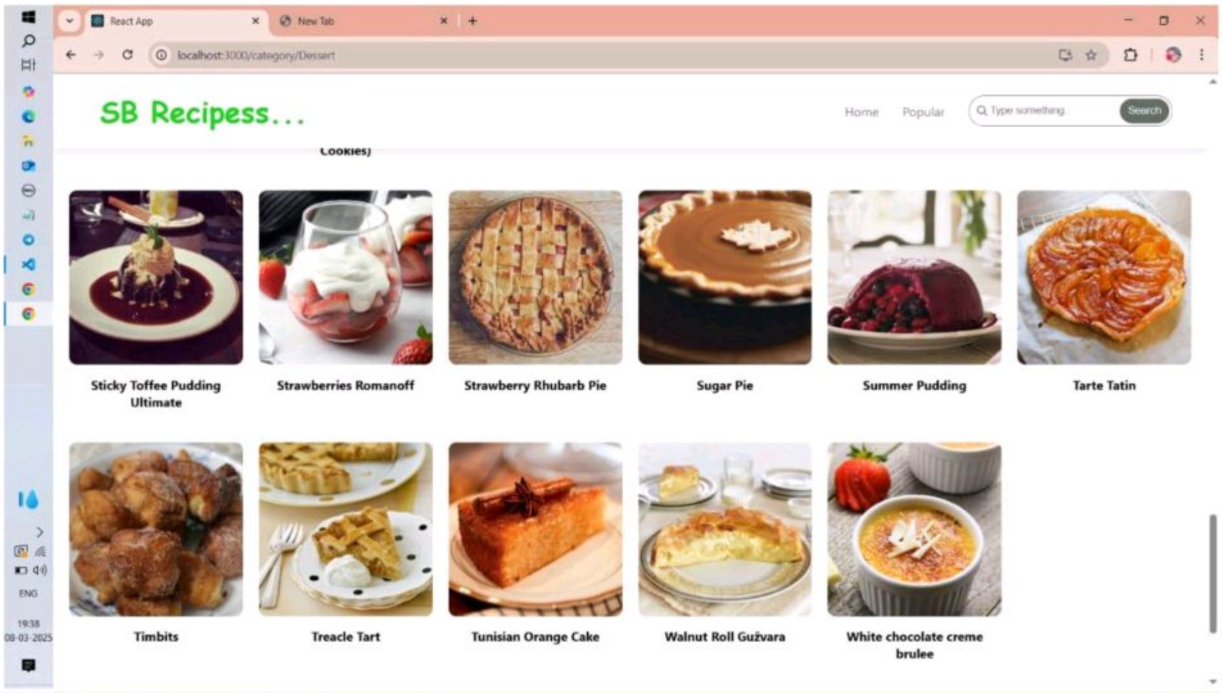
Video Tutorial

Ingredients

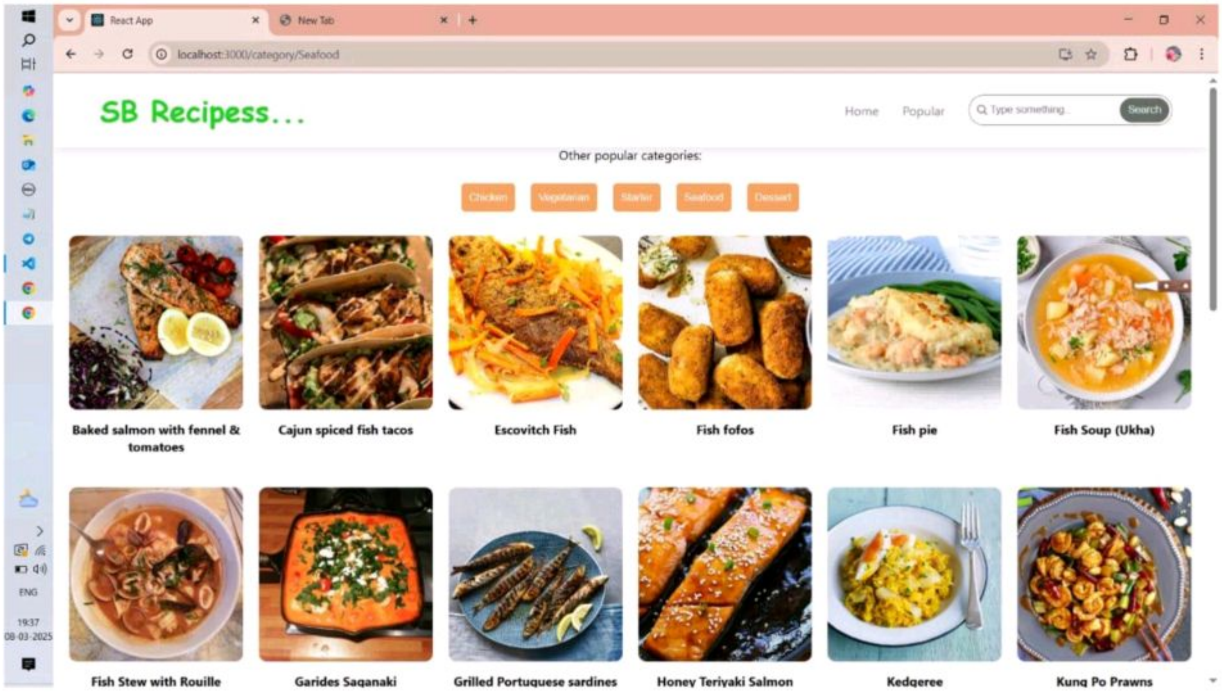
1 - Cashew nuts	12
2 - Khus khus	½ tbsp
3 - Cumin seeds	½ tbsp
4 - Onions	3 sliced thinly
5 - Ginger garlic paste	2 tsp
6 - Garlic	4 whole
7 - Mint	Leaves
8 - Cilantro	Leaves
9 - Saffron	½ tsp dissolved in ½ cup warm milk
10 - Ghee	2 tbsp
11 - Basmati rice	2 Cups
12 - Full fat yogurt	½ cup
13 - Cumin Seeds	1 tbsp
14 - Bay leaf	½
15 - Cinnamon	1 thin piece
16 - Cloves	3
17 - Cardamom	2
18 - Lamb	1 lb


React App
New Tab
localhost:3000
SB Recipess...
Home
Popular
Type something..
Search
Trending Dishes


React App
New Tab
localhost:3000/category/Dessert
SB Recipess...
Home
Popular
Type something..
Search
Pear Tarte Tatin
Polskie Nalesniki (Polish Pancakes)
Portuguese custard tarts
Pouding chomeur
Pumpkin Pie
Rock Cakes
Rocky Road Fudge
Rogaliki (Polish Croissant Cookies)
Salted Caramel Cheescake
Seri muka kuih
Spotted Dick
Sticky Toffee Pudding



React App
New Tab
localhost:3000/category/Dessert
SB Recipess...
Home
Popular
Type something...
Search
Cookies)
Sticky Toffee Pudding Ultimate
Strawberries Romanoff
Strawberry Rhubarb Pie
Sugar Pie
Summer Pudding
Tarte Tatin
Timbits
Treacle Tart
Tunisian Orange Cake
Walnut Roll Gužvara
White chocolate creme brulee



React App
New Tab
localhost:3000/category/Seafood
SB Recipess...
Home
Popular
Type something...
Search
Other popular categories:
Chicken
Vegetarian
Starter
Seafood
Dessert
Baked salmon with fennel & tomatoes
Cajun spiced fish tacos
Escovitch Fish
Fish fofos
Fish pie
Fish Soup (Ukha)
Fish Stew with Rouille
Garides Saganaki
Grilled Portuguese sardines
Honey Teriyaki Salmon
Kedgeree
Kung Po Prawns

KNOWN ISSUES

Despite extensive testing, the CookBook application may have some known issues that could affect user experience. Below are some identified issues along with possible causes and solutions.

1. Slow API Response

📌 Issue:

- Recipe data takes longer to load, especially when fetching from the MealsDB API.

💡 Possible Cause:

- API rate limits or network latency.

🛠 Potential Solution:

- Implement caching mechanisms to store previously fetched recipes.
- Show a loading spinner while waiting for the API response.

2. Search Function Case Sensitivity

📌 Issue:

- The search feature does not return results if queries don’t match exact case.

💡 Possible Cause:

- The search logic is case-sensitive and does not normalize input.

🛠 Potential Solution:

- Convert both user input and recipe names to lowercase before comparison.

3. UI Overlapping on Mobile Screens

📌 Issue:

- Some UI components (e.g., navbar, buttons) overlap on smaller screens.

💡 Possible Cause:

- Missing proper media queries for responsive design.

🛠 Potential Solution:

- Adjust CSS breakpoints for better mobile compatibility.
- Use flexbox and grid layouts for dynamic positioning.

4. Incorrect Category Filtering

📌 Issue:

- Clicking on a category sometimes displays incorrect recipes.

💡 Possible Cause:

- The selected category state is not updating correctly.

🛠 Potential Solution:

- Debug state management in Category.js to ensure proper updates.
- Implement a useEffect() hook to trigger a new API call when the category changes.

5. No Feedback on Failed API Calls

📌 Issue:

- When the API fails, users don't receive an error message.

💡 Possible Cause:

- No error handling implemented in API calls.

🛠 Potential Solution:

- Display error messages when API requests fail.
- Implement try-catch blocks and show fallback UI.

These issues will be addressed in future updates to enhance the performance, usability, and reliability of the CookBook application.

FUTURE ENHANCEMENTS

The CookBook application is designed to provide a seamless experience for recipe discovery and management. Below are potential future enhancements to improve usability, features, and overall performance.

1. User Authentication & Profiles

📌 Enhancement:

- Implement user sign-up and login functionality.
- Allow users to save favorite recipes to their profiles.

💡 Benefits:

- Personalized experience where users can bookmark and organize recipes.

2. Shopping List Feature

📌 Enhancement:

- Users can generate a shopping list from recipe ingredients.
- Option to add/remove items manually before shopping.

💡 Benefits:

- Makes grocery planning easier by providing a ready-to-use ingredient list.

3. Meal Planner Integration

📌 Enhancement:

- Allow users to schedule meals for the week.
- Provide caloric and nutritional breakdowns.

💡 Benefits:

- Helps users plan meals in advance and track their diet.

4. Dark Mode Support

📌 Enhancement:

- Implement a dark mode toggle for better user experience.

💡 Benefits:

- Reduces eye strain, especially during night-time browsing.

5. Community Recipe Sharing

📌 Enhancement:

- Enable users to submit their own recipes.
- Add a rating and review system for user-generated content.

💡 Benefits:

- Encourages community engagement and a wider variety of recipes.

6. Offline Mode for Saved Recipes

📌 Enhancement:

- Allow users to save recipes for offline access.

💡 Benefits:

- Users can view recipes without an internet connection, especially useful for kitchen use.

7. Voice-Activated Cooking Assistant

📌 Enhancement:

- Integrate voice commands for hands-free navigation.

💡 Benefits:

- Helps users follow recipes without touching the screen, making cooking more convenient.

8. AI-Based Recipe Recommendations

📌 Enhancement:

- Use AI to suggest recipes based on user preferences and available ingredients.

💡 Benefits:

- Provides personalized recommendations to match user tastes and dietary needs.