Frontend Development with React.js
Project Documentation

COOKBOOK

(A VIRTUAL KITCHEN)

Team ID : SWTID1741161918157855

Team Size : 4

Team Leader : VIKINESH V

Team member : VISHAL N

Team member : VISHAL R

Team member : YUVANRAJ S

ABSTRACT:

IN today’s fast-paced world, managing recipes, meal planning, and kitchen Organization can be challenging. CookBook: Your Virtual Kitchen Assistant .Aims to simplify and enhance the cooking experience through an intelligent,User-friendly application. This virtual assistant leverages advanced algorithms To provide personalized recipe recommendations based on user preferences, Available ingredients, and dietary restrictions.

The system integrates voice commands for hands-free operation, ensuring Convenience while cooking. It also features meal planning, grocery list Generation, and nutritional analysis, making it an all-in-one solution for home Cooks. By incorporating machine learning, the application continuously refines Recommendations and adapts to user habits over time.

This virtual assistant not only improves efficiency in the kitchen but also Promotes healthier eating habits and reduces food waste by suggesting recipes That maximize ingredient utilization. CookBook is designed to be a smart, Interactive, and accessible tool that transforms everyday cooking into an Effortless and enjoyable experience.

2. PROJECT OVEREVIEW

- **Purpose**:

  This document outlines the development of Cookbook: A Virtual Kitchen, an intelligent kitchen assistant designed to streamline the cooking experience. The purpose of this document is to present the application's objectives, features, and functionalities, including personalized recipe recommendations, meal planning, grocery list generation, and hands-free voice commands. By leveraging machine learning, Cookbook aims to enhance efficiency in the kitchen, promote healthier eating habits, and minimize food waste. This document serves as a guide for the project's scope, design, and implementation, ensuring a structured approach to its development.

- **Features:**

  - **Recipe Search & Browsing –** Users can explore a rich database of recipes categorized by cuisine and dietary needs.

  - **User-Friendly Design –** A seamless and visually appealing interface for enhanced user experience.

  - **Recipe Management –** Save, share, and organize favourite recipes effortlessly.

  - **API Integration –** Fetch recipes dynamically using Meals DB API.

  - **Dynamic Content –** Includes trending dishes, newsletter subscriptions, and tutorial videos for enhanced learning.

3.ARCHITECTURE

➢ Component Structure:

The Components folder contains reusable UI elements that are used across different pages.

- **Navbar Component:** Provides site-wide navigation to different sections of the application.
- **Hero Component:** Displays the introductory banner with a brief description and call-to-action.
- **Popular Categories Component:** Showcases popular recipe categories fetched from the API.
- **Trending Dishes Component:** Lists the most popular dishes based on user engagement.
- **Newsletter Component:** Allows users to subscribe to newsletters by entering their email.
- **Footer Component:** Contains links to social media, contact details, and app credits.

➢ State Management:

The application uses Reacts use State and use Effect hooks to manage state and handle API requests efficiently.

- **Use State:** Stores fetched data (categories, recipes, recipe details).
- **Use Effect:** Fetches data from APIs when components mount.

➢ Routing:

Uses React Router Dom for navigation between different pages.

- / → Home Page
- /category/:name → Category Page (shows dishes under a category)
- /recipe/:id → Recipe Page (displays a detailed recipe)

4.SETUP INSTRUCTIONS

- **Pre-Requisites for Developing a React.js Frontend Application**

Set Up a React.js Project:

- Create a new React app
- npx create-react-app my-react-app
- cd my-react-app

Start the development server:

- npm start
- Open http://localhost:3000 to view the app.
- Basic Web Development Knowledge
- Understand HTML, CSS, and JavaScript for structuring, styling, and interactivity.
- Choose a Code Editor
- Download a preferred IDE:
- Visual Studio Code

Clone and Run the Recipe App

- Download the project from Google Drive.
- Install dependencies
- cd recipe-app-react
- npm install
- Start the app
- npm start
- Access the app at http://localhost:3000.

5.FOLDER STRUCTURE

➢ Client:

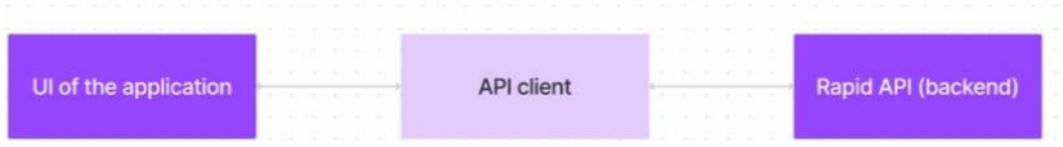
Here's a well-organized client-side (frontend) folder structure for your Cookbook: A Virtual Kitchen project:

- Components/ – Stores reusable UI components like Navbar, Hero Section, Popular Categories.
- Pages/ – Contains main pages like Home, Categories, Recipe Details.
- Styles/ – Includes CSS files for styling.
- /public – Contains static assets
- /src/utils – Includes helper functions (if any)

This structure keeps the frontend modular, scalable, and maintainable. Let me know if you need modifications

➢ Utilities:

- Recipe Discovery & Management
- User-Friendly Interface & Navigation
- Enhanced Cooking Experience



6.RUNNING THE APPLICATION

To set up and run the Cookbook: A Virtual Kitchen Assistant on your local machine, follow these steps:

- ❖ **Clone the Project Repository**
  - ○ Download the project files from the provided Google Drive link: Project Code
  - ○ Alternatively, clone the repository using Git:
  - ○ git clone <repository URL>
- ❖ **Install Required Dependencies**
  - ○ Navigate to the project folder and install the necessary dependencies using npm:
  - ○ cd recipe-app-react
  - ○ npm install
- ❖ **Start the Development Server**
  - ○ Run the following command to start the application:
  - ○ npm start
  - ○ This will launch the React application, and you can access it in your browser at:
  - ○ http://localhost:3000
- ❖ **Exploring the Features**
  - ○ Once the application is running, you can:
  - ○ Browse recipes from the MealsDB API
  - ○ Search for specific dishes
  - ○ This creates an optimized version of the app in the build folder.

7.COMPONENT DOCUMENTATION

➢ Key Components:

- Navbar – Provides navigation to home and category pages.
- Hero Section – Displays the main application overview.
- Popular Categories – Fetches and displays available recipe categories.
- Recipe Details – Shows ingredients, cooking instructions, and video tutorial.

➢ Reusable Components:

- Button – Customizable buttons used throughout the app.
- Recipe Card – A component that displays individual recipes

8. STATE MANAGEMENT

- **Global State:**

    - Not using Redux; instead, data fetching and state management are handled within components using React Hooks.

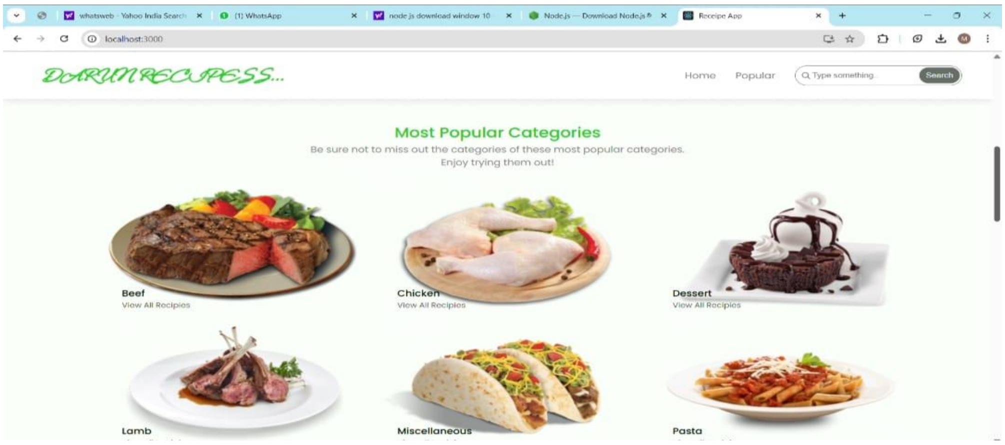
- **Local State:**

- Each component manages its own state using useState.

9.USER INTERFACE

➢ Popular categories

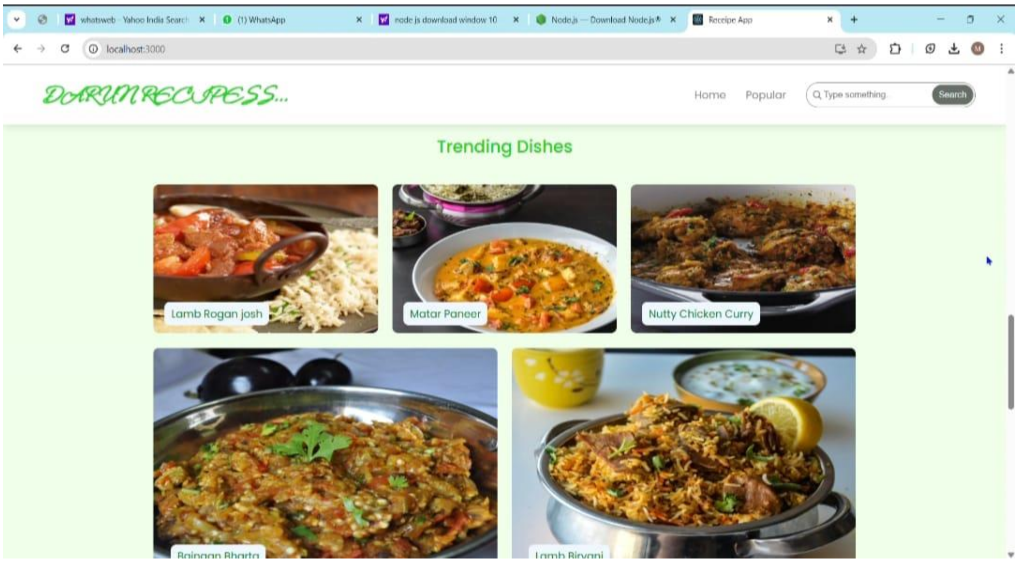
- This component contains all the popular categories of recipes..



Most Popular Categories

Be sure not to miss out the categories of these most popular categories.
Enjoy trying them out!

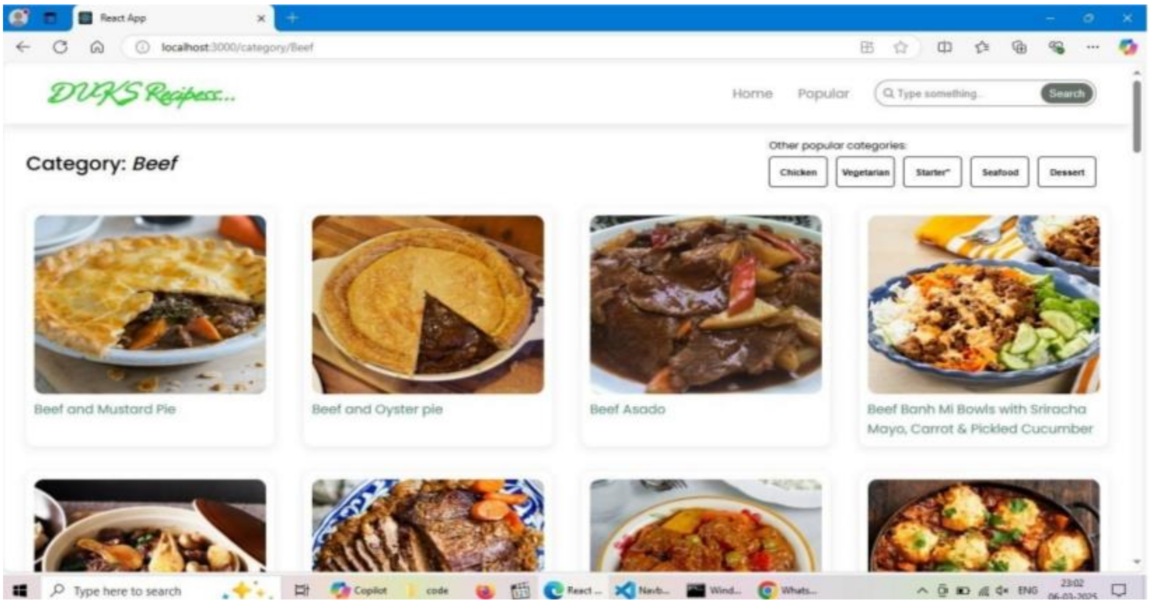
➢ Trending Dishes :



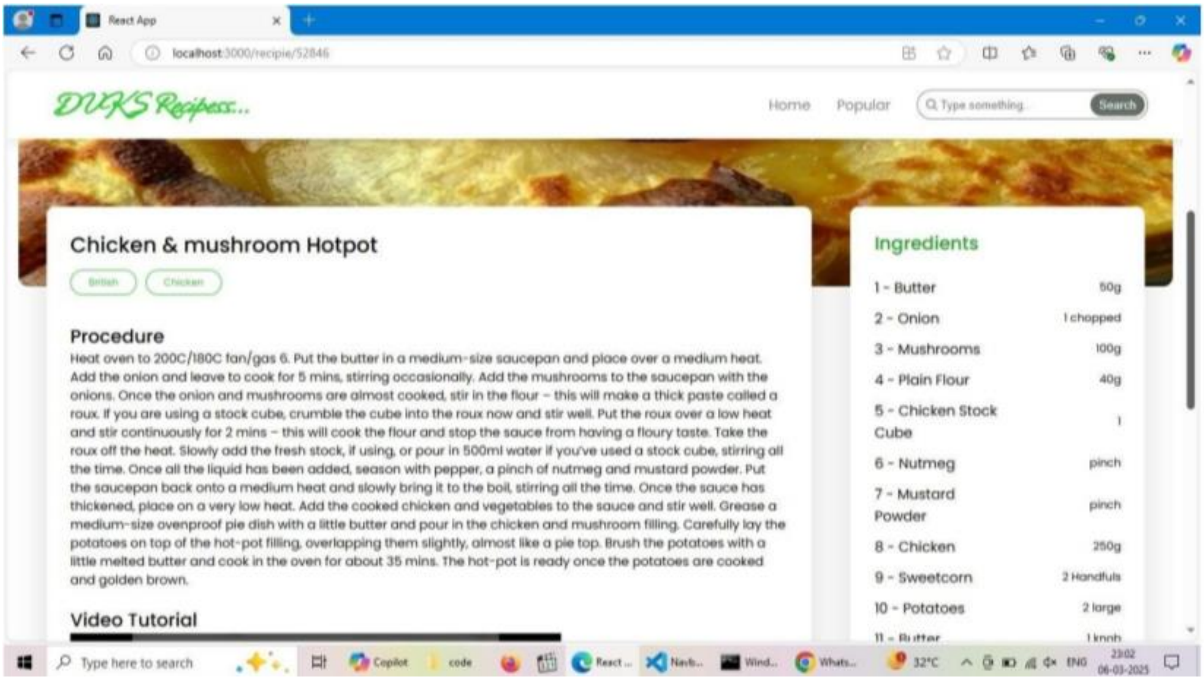
Trending Dishes

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

- ➢ **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



Chicken & mushroom Hotpot

Procedure

Heat oven to 200C/180C fan/gas 6. Put the butter in a medium-size saucepan and place over a medium heat. Add the onion and leave to cook for 5 mins, stirring occasionally. Add the mushrooms to the saucepan with the onions. Once the onion and mushrooms are almost cooked, stir in the flour – this will make a thick paste called a roux. If you are using a stock cube, crumble the cube into the roux now and stir well. Put the roux over a low heat and stir continuously for 2 mins – this will cook the flour and stop the sauce from having a floury taste. Take the roux off the heat. Slowly add the fresh stock, if using, or pour in 500ml water if you've used a stock cube, stirring all the time. Once all the liquid has been added, season with pepper, a pinch of nutmeg and mustard powder. Put the saucepan back onto a medium heat and slowly bring it to the boil, stirring all the time. Once the sauce has thickened, place on a very low heat. Add the cooked chicken and vegetables to the sauce and stir well. Grease a medium-size ovenproof pie dish with a little butter and pour in the chicken and mushroom filling. Carefully lay the potatoes on top of the hot-pot filling, overlapping them slightly, almost like a pie top. Brush the potatoes with a little melted butter and cook in the oven for about 35 mins. The hot-pot is ready once the potatoes are cooked and golden brown.

Video Tutorial

Ingredients

1 - Butter	50g
2 - Onion	1 chopped
3 - Mushrooms	100g
4 - Plain Flour	40g
5 - Chicken Stock Cube	1
6 - Nutmeg	pinch
7 - Mustard Powder	pinch
8 - Chicken	250g
9 - Sweetcorn	2 Handfuls
10 - Potatoes	2 large
11 - Butter	1 knob

10.STYLING

➢ CSS Frameworks & Libraries:

❖Bootstrap / Tailwind CS

- CookBook leverages either Bootstrap or Tailwind CSS for rapid UI development and responsiveness.
- Bootstrap provides a grid system, pre-designed components, and utility classes.
- Tailwind CSS offers a utility-first approach for custom styling while maintaining flexibility.

To install Bootstrap:

- Npm install bootstrap
- Import Bootstrap in your index.js:
- Import 'bootstrap/dist/css/bootstrap.min.css';

To install Tailwind CSS:

- Npm install -D tailwindcss postcss autoprefixer
- Npx tailwindcss init -p
- Add Tailwind directives to index.css:
- @tailwind base;
- @tailwind components;
- @tailwind utilities;

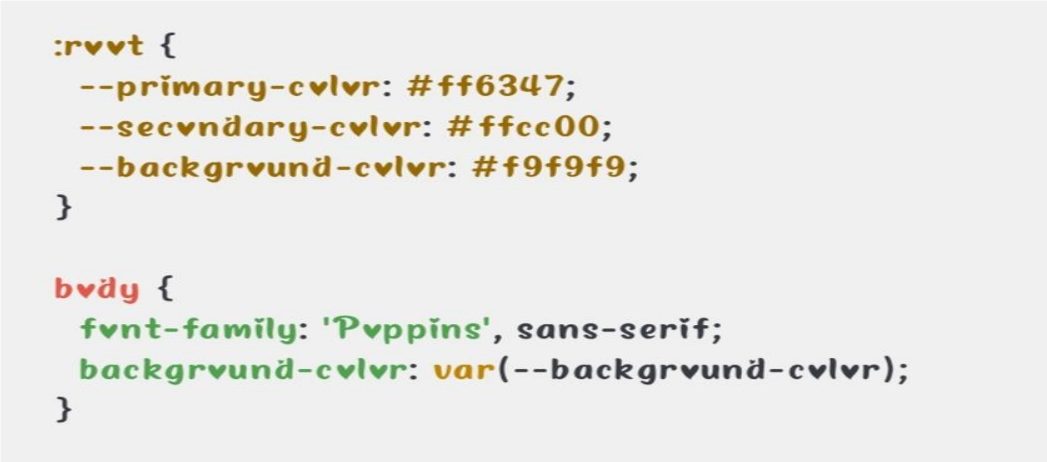
➢ Theming :

CookBook maintains a clean and modern UI with a focus on user experience. Theming is achieved using:

❖Global Stylesheet

The styles/ directory stores global CSS rules.

Example global.css:



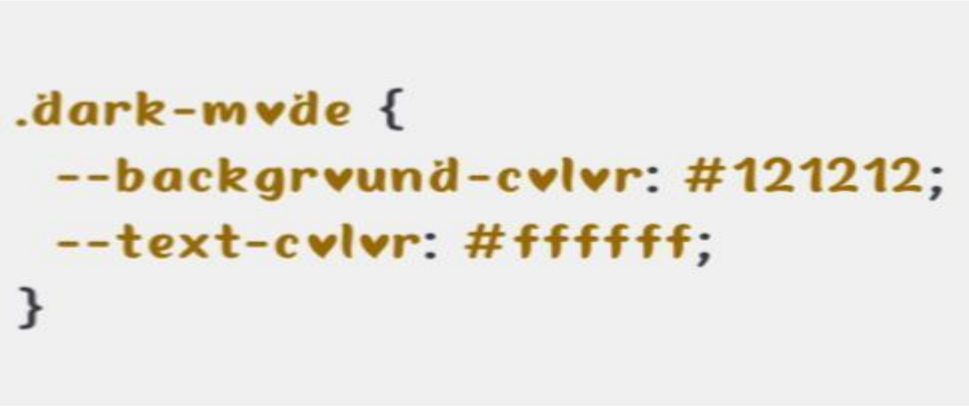
```
:root {
  --primary-color: #ff6347;
  --secondary-color: #ffcc00;
  --background-color: #f9f9f9;
}

body {
  font-family: 'Poppins', sans-serif;
  background-color: var(--background-color);
}
```

❖Dark Mode Support

CookBook supports light and dark themes using CSS variables.

Example:

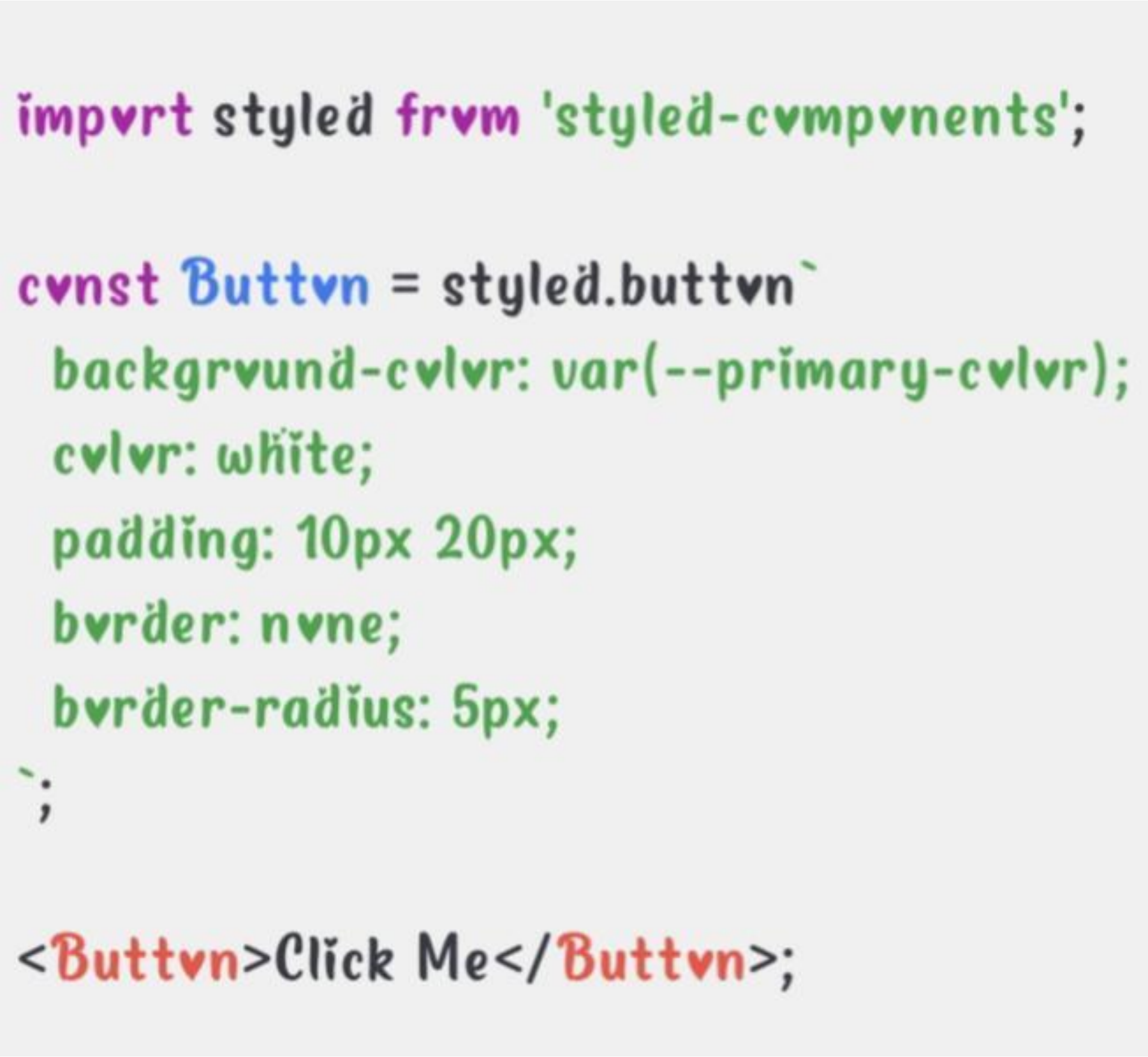


```
.dark-mode {
  --background-color: #121212;
  --text-color: #ffffff;
}
```

❖**Styled Components** (Optional)

- ○ Styled Components allow CSS-in-JS for scoped styling.
- ○ Install:
- ○ **Npm install styled-components**

Example usage:



```
import styled from 'styled-components';

const Button = styled.button`
  background-color: var(--primary-color);
  color: white;
  padding: 10px 20px;
  border: none;
  border-radius: 5px;
`;

<Button>Click Me</Button>;
```

11.TESTING

Testing Strategy

- The testing strategy for Cookbook includes the following types of tests:

1. Unit Testing:

- Tests individual functions, components, and API calls.
- Ensures correct rendering of UI components.

2. Integration Testing:

- Tests interaction between multiple components.
- Ensures proper data fetching and state management.

3. End-to-End (E2E) Testing:

- Simulates user interactions from loading the app to browsing recipes and viewing details.

4. Performance Testing:

- Measures load times for API calls and rendering.

5. Security Testing:

- Checks for vulnerabilities in API calls and user data handling.

6. Usability Testing:

- Ensures smooth navigation and an intuitive UI for users.

➢ Code Coverage:

- We use Jest and React Testing Library for unit and integration tests, while Cypress is used for end-to-end testing. Jest provides built-in support for measuring code coverage.

Enable Code Coverage in Jest

Modify the package.json file to include the following:

✓
```
"scripts": {
"test": "react-scripts test --coverage"
}
```

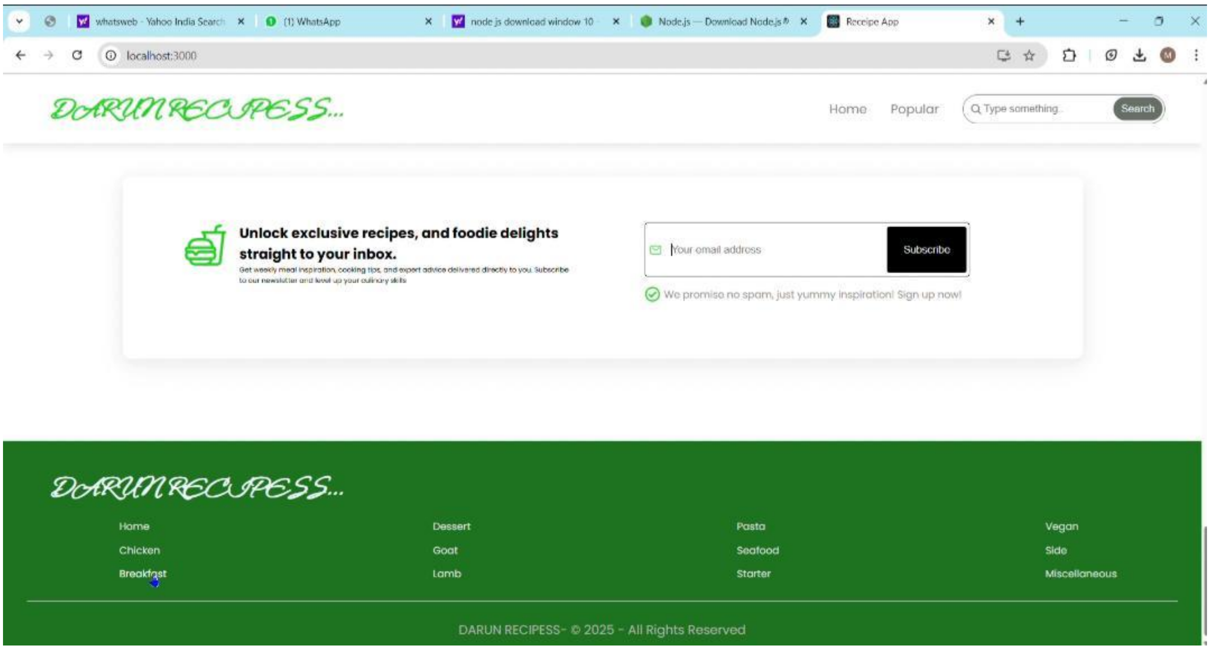
Run the test coverage report using:

✓
```
npm test -- --coverage
```

12.SCREENSHOTS

➢ News Letter

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



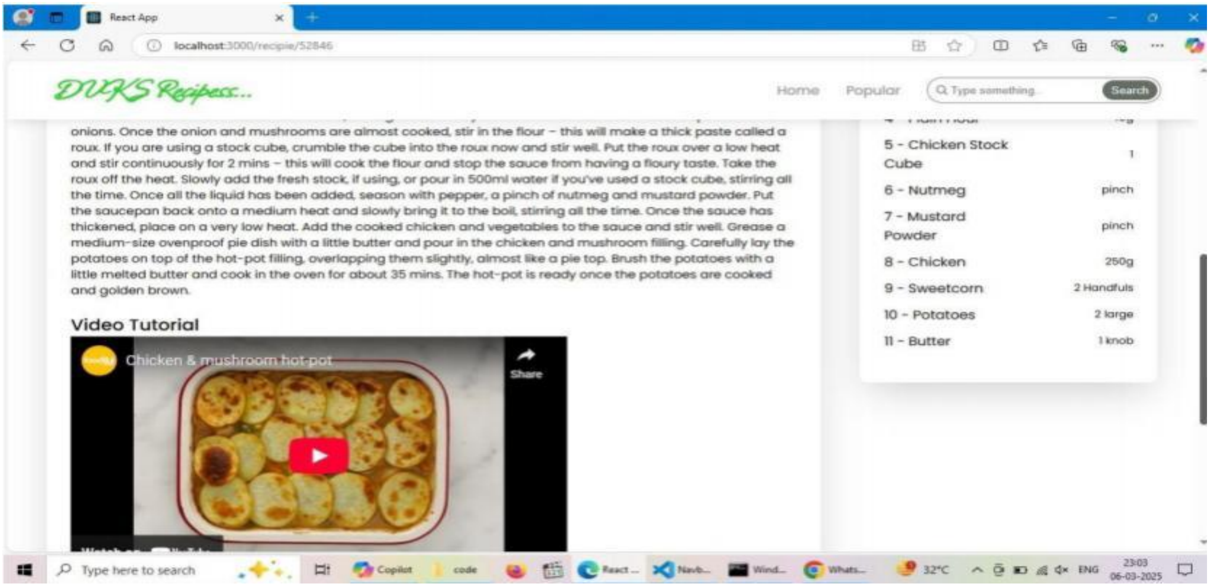
Unlock exclusive recipes, and foodie delights straight to your inbox.

Get weekly meal inspiration, cooking tips, and expert advice delivered directly to you. Subscribe to our newsletter and level up your culinary skills

We promise no spam, just yummy inspiration! Sign up now!

DARUN RECIPESS- © 2025 - All Rights Reserved

➢ Tutorial videos



onions. Once the onion and mushrooms are almost cooked, stir in the flour – this will make a thick paste called a roux. If you are using a stock cube, crumble the cube into the roux now and stir well. Put the roux over a low heat and stir continuously for 2 mins – this will cook the flour and stop the sauce from having a floury taste. Take the roux off the heat. Slowly add the fresh stock, if using, or pour in 500ml water if you've used a stock cube, stirring all the time. Once all the liquid has been added, season with pepper, a pinch of nutmeg and mustard powder. Put the saucepan back onto a medium heat and slowly bring it to the boil, stirring all the time. Once the sauce has thickened, place on a very low heat. Add the cooked chicken and vegetables to the sauce and stir well. Grease a medium-size ovenproof pie dish with a little butter and pour in the chicken and mushroom filling. Carefully lay the potatoes on top of the hot-pot filling, overlapping them slightly, almost like a pie top. Brush the potatoes with a little melted butter and cook in the oven for about 35 mins. The hot-pot is ready once the potatoes are cooked and golden brown.

Video Tutorial

5 - Chicken Stock Cube	1
6 - Nutmeg	pinch
7 - Mustard Powder	pinch
8 - Chicken	250g
9 - Sweetcorn	2 Handfuls
10 - Potatoes	2 large
11 - Butter	1 knob

Watch the demo: https://drive.google.com/file/d/1iMlDGBK2hLcAHnjCqgVNBlwCgQweL25B/view?usp=drivesdk

13.KNOWN ISSUES

1. API Limitations & Downtime

- Cookbook relies on the MealsDB API for fetching recipe data. Any downtime or rate limiting from the API provider may affect recipe availability.
- Some API requests might return incomplete or outdated data.

2. Slow Initial Load Time

- Due to external API requests and image-heavy content, the application may experience slow loading times, particularly on lower bandwidth networks.

3. Search Functionality Constraints

- The search feature depends on API responses, and certain queries might not yield expected results.
- There is currently no fuzzy search or typo correction, meaning users must enter exact keywords to find recipes.

4. Cross-Browser Compatibility

- While optimized for modern browsers like Chrome and Edge, some features may not work as expected on older versions of Safari or Internet Explorer.

5. Limited Offline Functionality

- The app requires an internet connection for most features. Recipes cannot be accessed offline unless cached in the browser.

6. User Interface (UI) Responsiveness Issues

- On smaller screens, some elements (like navigation bars or pop-ups) might not adjust properly.

7. Limited Personalization

- There is currently no option for user accounts, recipe bookmarking, or customization based on dietary preferences.

8. Security Considerations

- Since the app fetches data from external APIs, improper error handling may lead to exposure of API keys in developer tools.

14.FUTURE ENHANCEMENTS:

- Implement user authentication to allow users to save favorite recipes.
- Add custom recipe creation feature.
- Improve UI animations for better user experience.
- User authentication and personalized recipe collections
- Advanced filtering options (e.g., dietary preferences)
- Social sharing features