**COOKBOOK: YOUR VIRTUAL KITCHEN ASSISTANT**

**TEAM DETAILS**

|  |  |
| --- | --- |
| **TEAM LEADER** | KAVIYADHARSHINI. V |

|  |  |  |
| --- | --- | --- |
| **TEAM MEMBERS** | **NAMES** | **ROLES** |

|  |  |  |
| --- | --- | --- |
| **TEAM MEMBER 1** | KAVIYADHARSHINI. V | CODING EXECUTION |
| **TEAM MEMBER 2** | KEERTHANA. N | DOCUMENTATION |
| **TEAM MEMBER 3** | LATHIKA. J | DEMO VIDEO |
| **TEAM MEMBER 4** | MAHIMA. S | DOCUMENTATION |

**INTRODUCTION**

**Cookbook: Your Virtual Kitchen Assistant** is a smart, user-friendly mobile application designed to simplify and enhance your cooking experience. Whether you're a seasoned home chef or just getting started in the kitchen, Cookbook helps you discover, organize, and prepare meals with confidence and ease.

The app serves as your all-in-one digital kitchen companion, offering features such as personalized recipe recommendations, step-by-step cooking guidance, grocery list management, and meal planning tools. With an intuitive interface and intelligent features, Cookbook adapts to your preferences, dietary needs, and cooking habits to make every meal a success.

Cookbook transforms the way users approach cooking — from planning and shopping to prepping and plating — turning everyday cooking into an effortless and enjoyable experience.

**PROJECT OVERVIEW**

**Purpose**

The primary goal of CookBook is to provide a user-friendly platform that caters to individuals  passionate about cooking, baking, and exploring new culinary horizons. Our objectives  include:

• **User-Friendly Experience:** Create an interface that is easy to navigate, ensuring users  can effortlessly discover, save, and share their favourite recipes.

• **Comprehensive Recipe Management:** Offer robust features for organizing and  managing recipes, including advanced search options.

• **Technology Stack:** Leverage modern web development technologies, including  React.js, to ensure an efficient, and enjoyable user experience.

**Key features**

* The app helps users find recipes suitable for various occasions, likely sorted by meal types, cuisines, or dietary preferences.
* The homepage prominently features **high-quality images** of dishes.
* This visual-first approach makes it easy to browse and get inspired by the appearance of the meals.
* A **search bar** is visible in the top right corner with a placeholder: *“Type something…”* and a “Search” button. This allows users to **quickly find specific recipes** or ingredients.
* A button labeled **"Recipes"** encourages users to dive deeper into the app’s recipe library.

**ARCHITECTURE**

**Component structure**

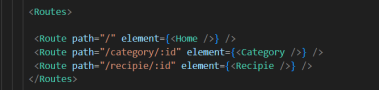
* App component: Root component
* Header components: Logo (e.g., "SB Recipes") and Navigation.
* Sidebar components: Filter categories (e.g., cuisine, difficulty, dietary type)
* Footer components: Links to About, Contact, Terms, etc.

**State Management**

Currently managed using React’s built-in useState and props.

**Routing**

Using React Routing Configuration defines how different components/pages are mapped to specific URL paths.



**Figure 1:** Routing path of Application

**SETUP INSTRUCTION**

**Pre requisites**

* Node.js and npm
* React.js
* HTML, CSS, and JavaScript
* Vs-code
* Git/Github

**Installation**

* ***Node.js :***

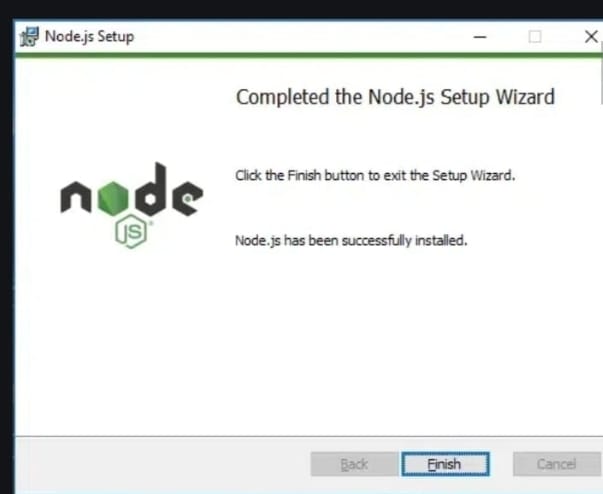
Install Node.js and npm on your development machine, as they are required to run JavaScript on the server-side.

Download: <https://nodejs.org/en/download/>

After Downloading, the installation process begins

**✅ In Windows**

1. Run the downloaded .msi installer.
2. Follow the setup wizard:
   * Accept the license agreement
   * Choose the destination folder
   * Make sure **"Add to PATH"** is selected
   * Keep default options unless you know otherwise
3. Finish the installation.



**Figure2:** Completion of Node.js

1. Verify the installation:

* Open **Command Prompt** or **PowerShell**  Enter to check the Node.js version.

1. Type **npm -v** and press Enter to check the npm version.
2. Both commands should return version numbers, confirming successful installation.

* ***React.js:***
* 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 <https://localhost:3000> in your web browser.

**FOLDER STRUCTURE**

**Client**

* src/components – Reusable components
* src/pages – Individual page layouts
* src/assets – Images, styles
* src/App.js – Root application

****

**Figure3:** Sources of Application

**Utilities**

**The helper functions** likely focus on handling **API interactions**, **data formatting**, **error handling**, and **local storage**.

**RUNNING THE APPLICATION**

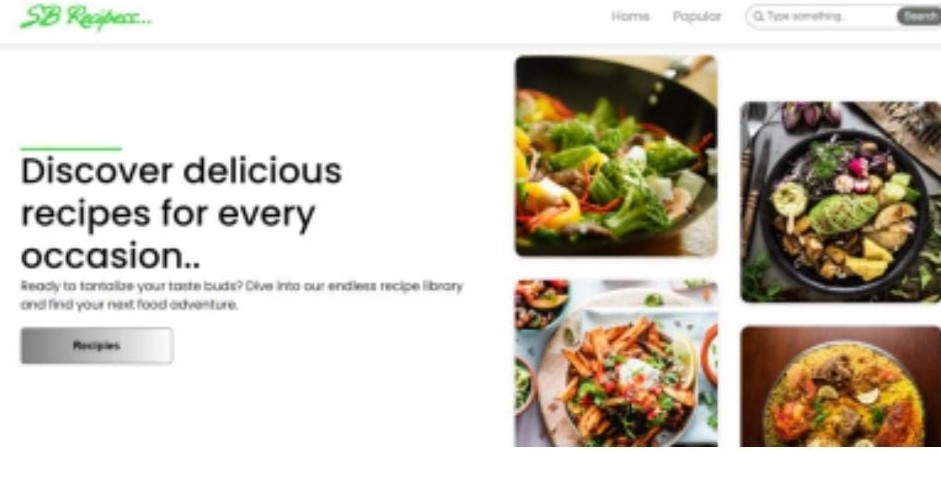
* Open the VS-code and you can see the code file in it.
* You have to install node modules for project execution, So use this command

npm install

* After installing all the node modules enter the project run command

npm start

* Now your application will be opened in browser with url <https://localhost:3000>



**Figure4:** Landing page of Application

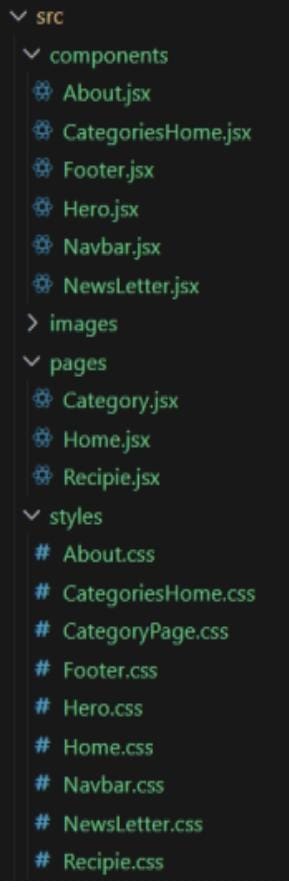
**COMPONENT DOCUMENTATION**

**Key components**

* The Home component serves as the landing page, showcasing featured recipes, a search bar, and a call-to-action to explore the recipe library.
* The **Recipe List component** displays all available recipes or filtered results based on user preferences.
* The **Popular Recipes component** highlights trending or most-viewed recipes to help users find popular meal ideas.
* The **Favorites component** shows a list of recipes the user has saved or bookmarked.
* The **User Profile component** lets users manage their profile, view saved recipes, and update preferences.

**Reusable Components**

* The **Recipe Card component** is used to display a summary of a recipe (image, title, time) across recipe listings.
* The **Search Bar component** allows users to search for recipes by keyword, and can be reused on the homepage or inside the header.
* The **Header (or Navbar) component** contains the app logo, navigation links, and sometimes the search input.
* The **Footer component** appears at the bottom of all pages and contains links to legal or informational pages.

 **Figure5:** Components of Application

**STATE MANAGEMENT**

* Global State: Not implemented yet. Future plan → Context API.
* Local State: Managed using useState within components.

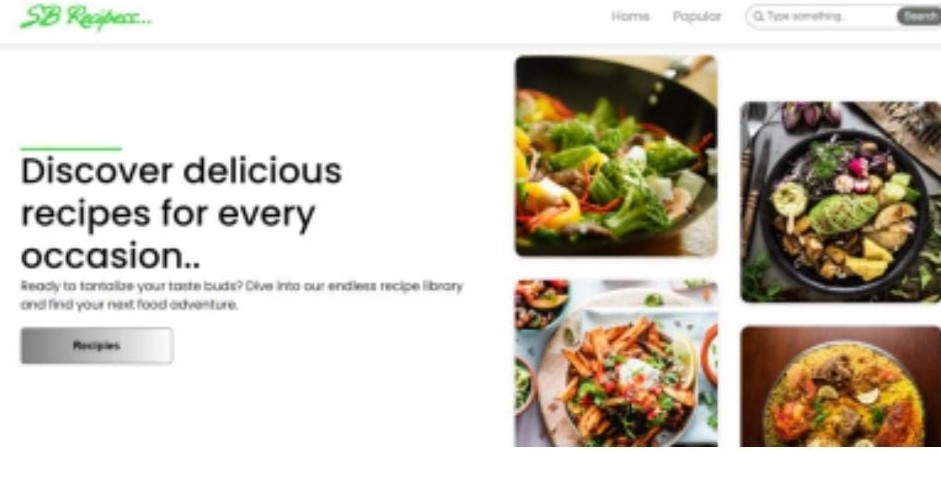
**STYLING**

* The application uses **CSS Modules** for component-scoped styling, allowing styles to be locally scoped and avoiding global class conflicts.
* Utility-first styling eg: with **Tailwind CSS** can also be used based on project preference.
* Global styles and theme variables (colors, fonts) are defined centrally for consistency.

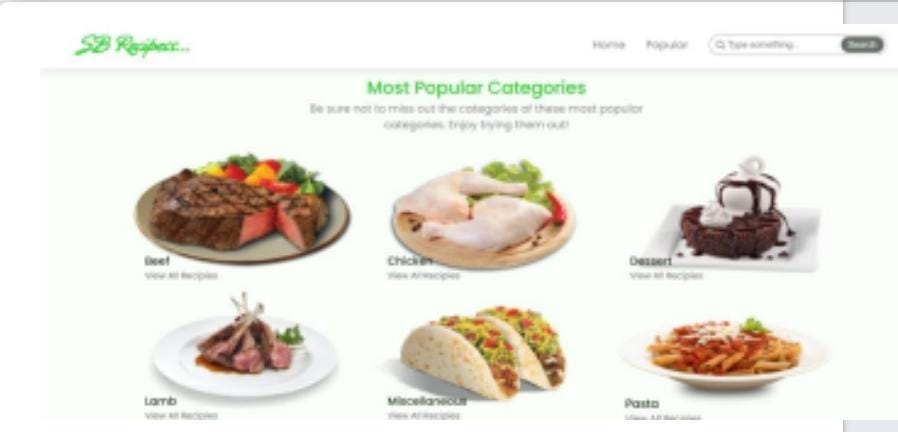
**TESTING**

* This application uses **automated testing** to ensure reliability, maintainability, and correctness of core features.
* Testing is performed across components, pages, and utility functions using industry-standard tools.
* In future Jest & React Testing Library for unit testing.

**USER INTERFACE**



**Figure6:** Home page



**Figure7:** Most Popular Categories

**KNOWN ISSUES**

* No global state management yet
* Limited routing

**FUTURE ENHANCEMENTS**

* Add authentication so users can create accounts and save their favorite recipes.
* Allow users to upload their own recipes with a form that includes title, image, ingredients, and steps.
* Add a feature where users can click a button to save or bookmark recipes for easy access later.