**PROJECT DOCUMENTATION**

**CookBook: Your Virtual Kitchen Assistant.**

**1. Introduction**

**• Project Title: CookBook: Your Virtual Kitchen Assistant.**

**• Team ID: NM2025TMID31261**

**• Team Leader: SANTHIYA K –lavanyasanthiya613@gmail.com**

**• Team Members:**

**O SARANYA A S –saranyaannadurai49@gmail.com**

**O SARITHA S –karthikeyan201991@gmail.com**

**O SAROJINI C – sarojinisaro234@gmail.com**

**O SATHURTHIKKA Y –ypathmini2@gmail.com**

**2. Project Overview**

* **Purpose:**

**"CookBook is your ultimate virtual kitchen assistant, designed to make cooking easier and more enjoyable. With a vast recipe library, meal planning tools, and step-by-step cooking guidance, you'll be whipping up delicious meals in no time. Whether you're a seasoned chef or a kitchen newbie, CookBook is here to help you cook with confidence. Get cooking and make every meal a masterpiece!"**

* **Goals:**
  + 1. **Centralized Recipe Collection – Provide a digital platform to store and access recipes easily.**
    2. **Easy Navigation – Use a clean UI and React Router for smooth browsing between categories and recipe details.**
    3. **Learning Support – Integrate YouTube tutorials for step-by-step cooking guidance.**
    4. **Category Organization – Group recipes by type (Beef, Chicken, Dessert, etc.) for quick discovery.**
    5. **User-Friendly Experience – Design an interface that is simple, responsive, and attractive.**
    6. **Reusable & Scalable – Build with React components so new features and recipes can be added easily.**
* **Key Features:**
  + 1. **Recipe Categories – Browse recipes by popular food categories (Beef, Chicken, Dessert, etc.).**
    2. **Recipe Details Page – View ingredients, preparation steps, and video tutorials.**
    3. **YouTube Video Embedding – Watch cooking videos directly inside the app.**
    4. **Search & Navigation – Quickly find specific recipes using smooth navigation.**
    5. **Responsive Design – Works on desktop, tablet, and mobile devices.**
    6. **Reusable Components – Built with React components for cards, lists, forms, etc.**
    7. **API Integration (if added) – Fetch recipes dynamically using Axios.**
    8. **Modern UI – Styled with CSS and React Icons for a professional look.**

**3. Architecture**

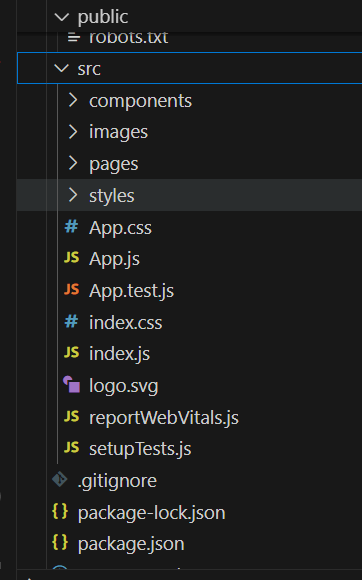
* **Component Structure**
  + **App.js — Root component, sets up routes and layout.**
  + **Navbar.jsx — Provides site navigation across pages (Home, Categories, Recipes).**
  + **Home.jsx — Homepage container, displays Hero, CategoriesHome, and NewsLetter.**
  + **CategoriesHome.jsx — Shows recipe categories on the homepage.**
  + **Category.jsx — Page that lists recipes filtered by category.**
  + **Recipie.jsx — Page displaying a single recipe with details.**
  + **About.jsx — Static page about the Cookbook application.**
  + **Footer.jsx — Footer with branding, copyright, and links.**
* **State Management**
  + **Local State: Managed using React useState and useEffect.**
  + **API Integration: Axios used for fetching data from CookBook API & YouTube API.**
* **Routing**
  + **Library: react-router-dom**
  + **Routes:**
    - **/ Home.jsx**
    - **/pages/Category => food Category.jsx**
    - **/category/Recipie ? Recipie.jsx**

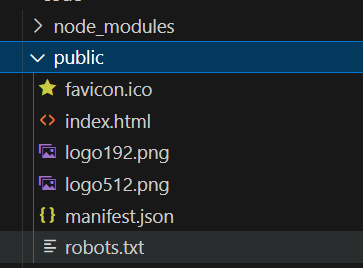
**4. Setup Instructions**

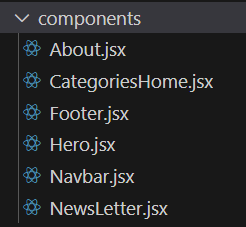
**Prerequisites**

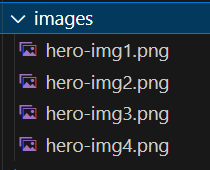
* **Node.js & npm**
  + **Node.js is required to run React applications.**
  + **npm (Node Package Manager) is used to install dependencies.**
  + **Download Node.js**
* **React.js**
  + **React is the main JavaScript library used to build this project.**
  + **If you don’t have an existing React app, create one using:**
  + **npx create-react-app my-app**
  + **cd my-app**
  + **npm start**
  + **In SB Fitzz, the React app is already created, so you just need to install dependencies (npm install).**
* **Git**
  + **Used for cloning and version control.**
  + **Download Git**
* **Code Editor**
  + **Recommended: Visual Studio Code (VS Code)**
  + **Download VS Code**
* **Basic Knowledge**
  + **HTML, CSS, JavaScript**
  + **React concepts (components, props, hooks, state, routing)**
* **Installation**
  + **Get the code:**
    - **Download the code from the drive link given below:**
      * + https://drive.google.com/drive/folders/1u8PnV\_mE0mwKkH\_CvuNpliZtRLJZMqrO?usp=sharing
* **Install Dependencies:**
  + **Navigate into the cloned repository directory and install libraries:**
    - **Navigate into the cloned repository directory and install libraries:**
      * **cd CODE**
      * **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.**
* **Environment Variables**
  + **Create a .env file with:**
    - **REACT\_APP\_API\_URL=<https://exercisedb.p.rapidapi.com/exercises/equipmentList>**
    - **REACT\_APP\_YOUTUBE\_API\_KEY=<33cf3a7616msh4c3b1e3204f24e2p1294b3jsne16a7323d732>**

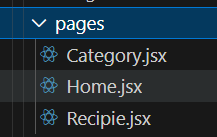
**5. Folder Structure**

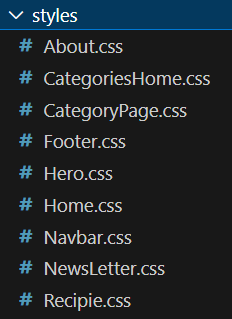


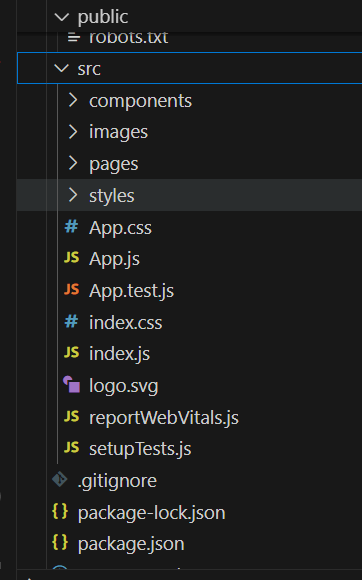












**6.Running the Application**

* **Start development server:**
  + **npm start**
* **Build for production:**
  + **npm run build**
* **Run tests:**
  + **npm test**

**7. Component Documentation**

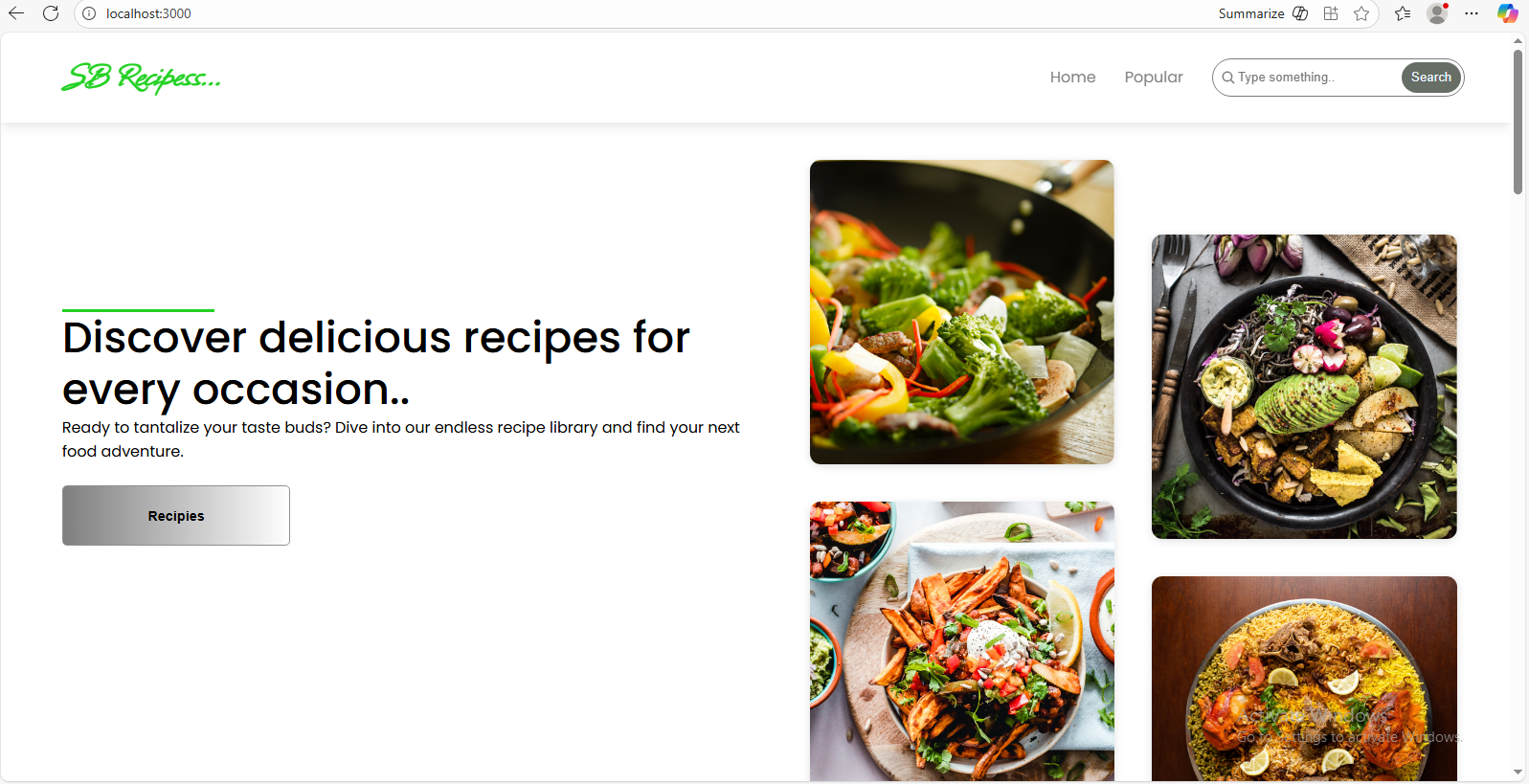
* **index.js => renders <App />**
* **App.js => wraps Navbar, Routes, and Footer**
* **Home.jsx => (uses Hero, CategoriesHome, NewsLetter)**
* **category => Category.jsx (may also reuse CategoriesHome)**
* **Recipie.jsx => (shows details of a recipe)Components import their CSS modules for styling.**
* **images => used in Hero, Home, and Category pages**

**8. State Management**

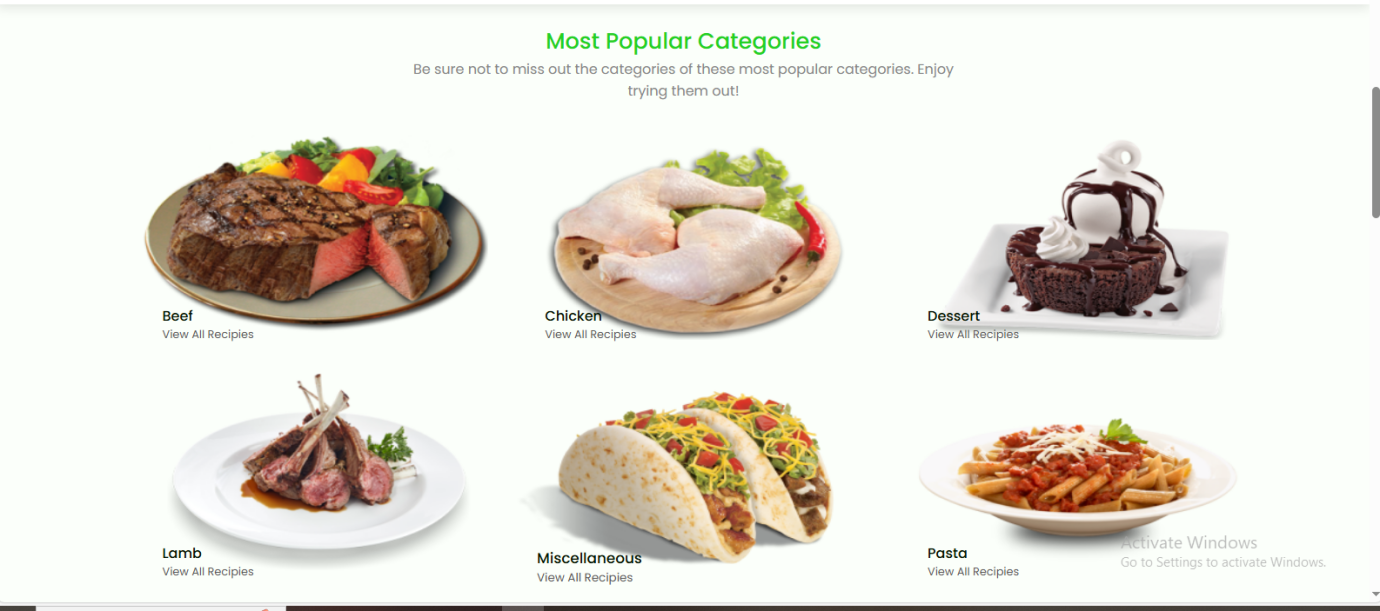
* **Local State:**
  + **Search queries stored in HomeSearch.**
  + **API data fetched and stored per-page.**
* **Global State:**
  + **Not implemented — app uses component-level state.**

**9. User Interface**

* **Pages include:**
  + **Home (Hero + Search )**



* **Popular:**



**10. Styling**

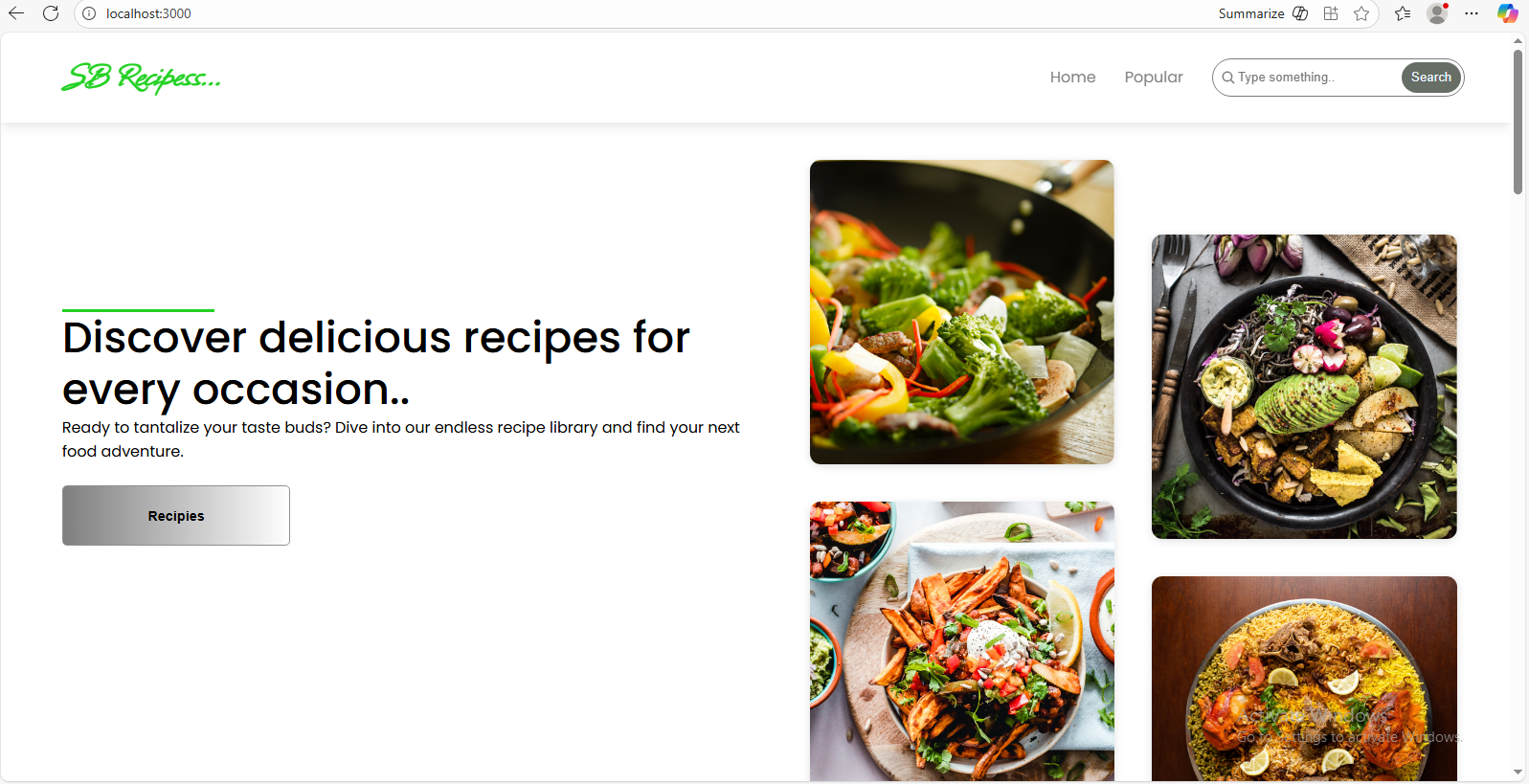
* **Frameworks Used: Tailwind CSS / Bootstrap.**
* **Custom CSS: Stored in src/styles/.**
* **Each page/component has a dedicated CSS file for modularity.**

**11. Testing**

* **Libraries Used: Jest, React Testing Library.**
* **Unit Tests: Written in App.test.js.**
* **Setup: Configured with setupTests.js.**

**12. Screenshots / Demo**

* **Demo Link:**
  + **[WhatsApp Video 2025-09-08 at 5.11.46 PM.mp4](https://drive.google.com/file/d/1X1h8u4Y5ebqBiN2rjgiPK1mfhOpS61u3/view?usp=drive_link)**
* **Screenshot:**



**13. Known Issues**

* + **API rate-limit may cause some exercises not to load.**
  + **YouTube API sometimes fails to fetch related videos.**

**14. Future Enhancements**

* **User Authentication** – Allow users to sign up, log in, and save their favorite recipes.
* **Recipe Submission** – Let users add and share their own recipes with the community.
* **Search & Filters** – Advanced search by ingredients, cuisine, prep time, or dietary needs.
* **Favorites & Collections** – Users can bookmark recipes and organize them into collections.