PROJECT DOCUMENTATION

CookBook: Your Virtual Kitchen Assistant.

1. Introduction

- Project Title: CookBook: Your Virtual Kitchen Assistant.

- Team ID: NM2025TMID31189

- Team Leader: ABISHEK P-abishekbca73@gmail.com

- Team Members:

    o ABDUL MAJEETH K-abdulmajeeth3765@gmail.com

    o ABINESH P-abineshabinesh26823@gmail.com

    o AKASHKUMAR C-mrminato444@gmail.com

    o AMUTHAN D-d.amuthan.07@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:**
    i. **Centralized Recipe Collection – Provide a digital platform to store and access recipes easily.**
    ii. **Easy Navigation – Use a clean UI and React Router for smooth browsing between categories and recipe details.**
    iii. **Learning Support – Integrate YouTube tutorials for step-by-step cooking guidance.**
    iv. **Category Organization – Group recipes by type (Beef, Chicken, Dessert, etc.) for quick discovery.**
    v. **User-Friendly Experience – Design an interface that is simple, responsive, and attractive.**
    vi. **Reusable & Scalable – Build with React components so new features and recipes can be added easily.**
- **Key Features:**
    i. **Recipe Categories – Browse recipes by popular food categories (Beef, Chicken, Dessert, etc.).**
    ii. **Recipe Details Page – View ingredients, preparation steps, and video tutorials.**
    iii. **YouTube Video Embedding – Watch cooking videos directly inside the app.**
    iv. **Search & Navigation – Quickly find specific recipes using smooth navigation.**
    v. **Responsive Design – Works on desktop, tablet, and mobile devices.**
    vi. **Reusable Components – Built with React components for cards, lists, forms, etc.**
    vii. **API Integration (if added) – Fetch recipes dynamically using Axios.**
    viii. **Modern UI – Styled with CSS and React Icons for a professional look.**

3. Architecture

- **Component Structure**
    o **App.js — Root component, sets up routes and layout.**
    o **Navbar.jsx — Provides site navigation across pages (Home, Categories, Recipes).**
    o **Home.jsx — Homepage container, displays Hero, CategoriesHome, and NewsLetter.**
    o **CategoriesHome.jsx — Shows recipe categories on the homepage.**
    o **Category.jsx — Page that lists recipes filtered by category.**

- o **Recipie.jsx — Page displaying a single recipe with details.**
- o **About.jsx — Static page about the Cookbook application.**
- o **Footer.jsx — Footer with branding, copyright, and links.**
- **State Management**
  - o **Local State: Managed using React useState and useEffect.**
  - o **API Integration: Axios used for fetching data from CookBook API & YouTube API.**
- **Routing**
  - o **Library: react-router-dom**
  - o **Routes:**
    - ■ **/ Home.jsx**
    - ■ **/pages/Category => food Category.jsx**
    - ■ **/category/Recipie ? Recipie.jsx**

4. Setup Instructions

Prerequisites

- **Node.js & npm**
  - o **Node.js is required to run React applications.**
  - o **npm (Node Package Manager) is used to install dependencies.**
  - o **Download Node.js**
- **React.js**
  - o **React is the main JavaScript library used to build this project.**
  - o **If you don't have an existing React app, create one using:**
  - o **npx create-react-app my-app**
  - o **cd my-app**
  - o **npm start**
  - o **In SB Fitzz, the React app is already created, so you just need to install dependencies (npm install).**
- **Git**
  - o **Used for cloning and version control.**
  - o **Download Git**
- **Code Editor**
  - o **Recommended: Visual Studio Code (VS Code)**
  - o **Download VS Code**
- **Basic Knowledge**
  - o **HTML, CSS, JavaScript**
  - o **React concepts (components, props, hooks, state, routing)**
- **Installation**
  - o **Get the code:**
    - ■ **Download the code from the drive link given below:**
      - o https://drive.google.com/drive/folders/1u8PnV_mEOmwKkH_CvuNpliZtRLJZMqrO?usp=sharing
- **Install Dependencies:**
  - o **Navigate into the cloned repository directory and install libraries:**
    - ■ **Navigate into the cloned repository directory and install libraries:**
      - • **cd CODE**
      - • **npm install**
  - o **Start the Development Server:**
    - ■ **To start the development server, execute the following command:**
      - • **npm start**
- **Access the App:**
  - o **Open your web browser and navigate to http://localhost:3000.**
  - o **You should see the application's homepage, indicating that the installation and setup were successful.**
- **Environment Variables**
  - o **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

```
∨ public
    ≡ robots.txt
∨ src
    > components
    > images
    > pages
    > styles
    # App.css
    JS App.js
    JS App.test.js
    # index.css
    JS index.js
    🔲 logo.svg
    JS reportWebVitals.js
    JS setupTests.js
  ◈ .gitignore
 {} package-lock.json
 {} package.json
```

```
> node_modules
∨ public
    ★ favicon.ico
    <> index.html
    🖼 logo192.png
    🖼 logo512.png
    {} manifest.json
    ≡ robots.txt
```
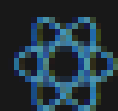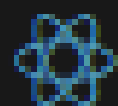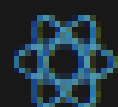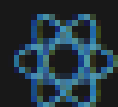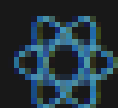
components

- About.jsx
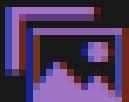- CategoriesHome.jsx
- Footer.jsx
- Hero.jsx
- Navbar.jsx
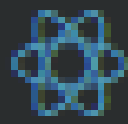- NewsLetter.jsx
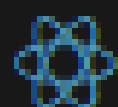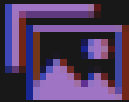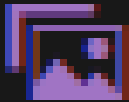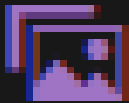
images

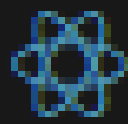- hero-img1.png
- hero-img2.png
- hero-img3.png
- hero-img4.png

pages
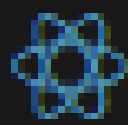
- Category.jsx
- Home.jsx
- Recipie.jsx

styles

- # About.css
- # CategoriesHome.css
- # CategoryPage.css
- # Footer.css
- # Hero.css
- # Home.css
- # Navbar.css
- # NewsLetter.css
- # Recipie.css
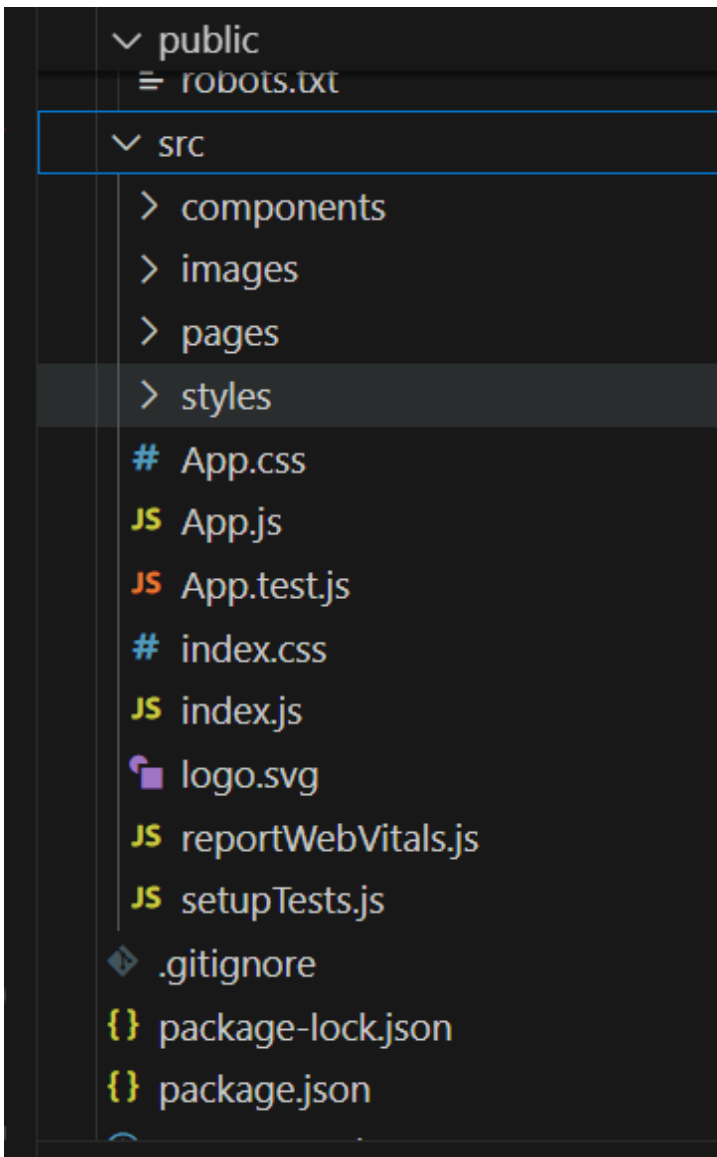
6. Running the Application

- **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.**
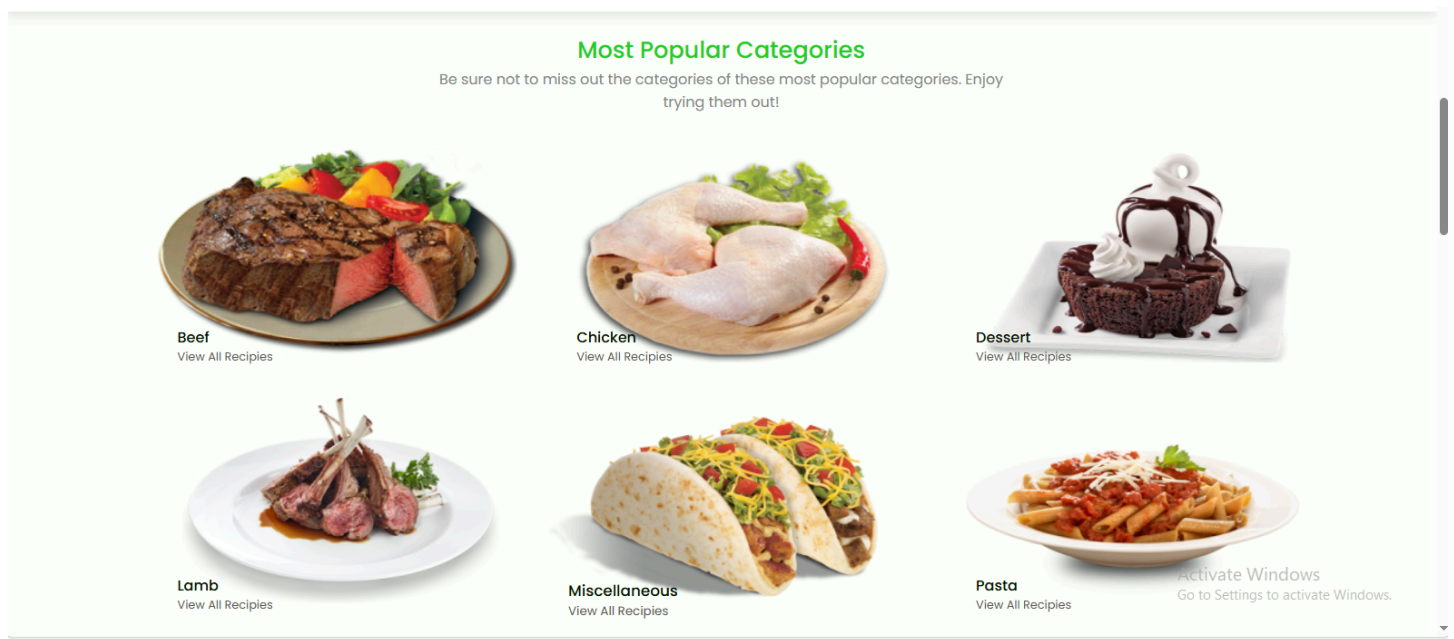- **images => used in Hero, Home, and Category pages**

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)**
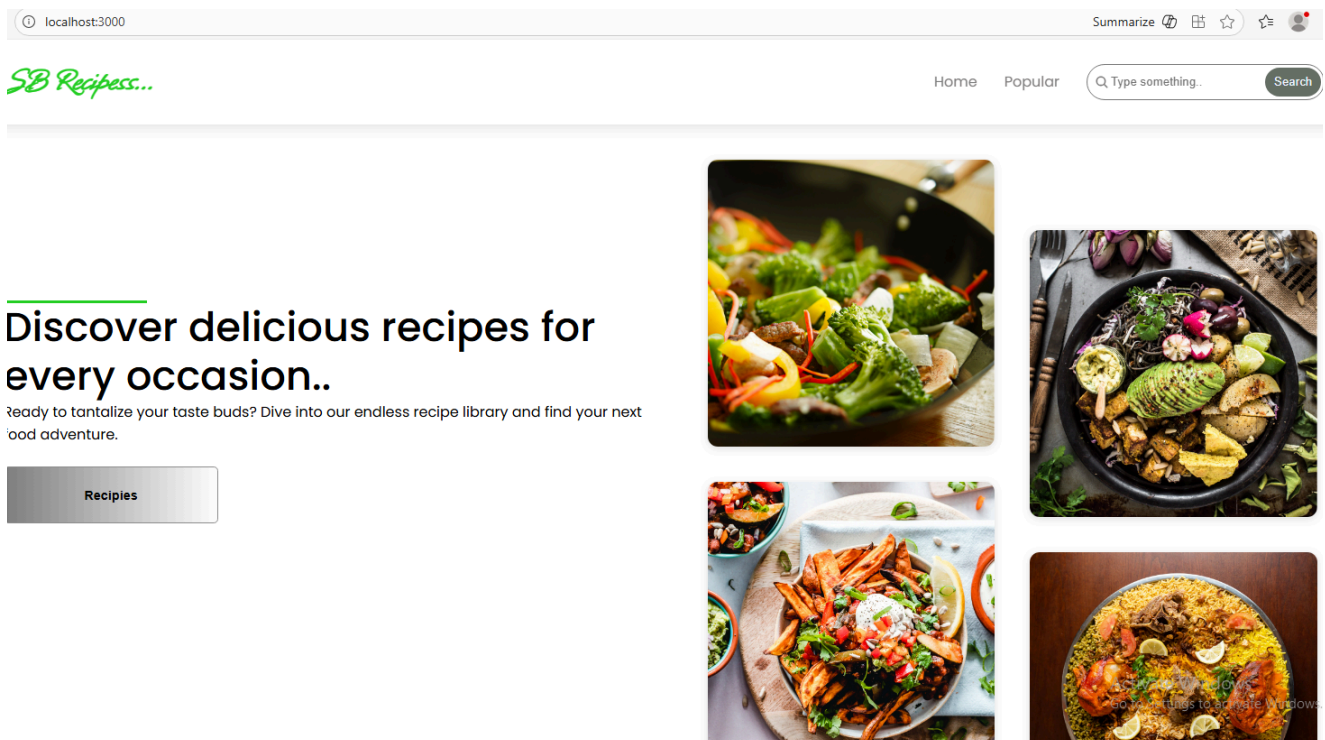
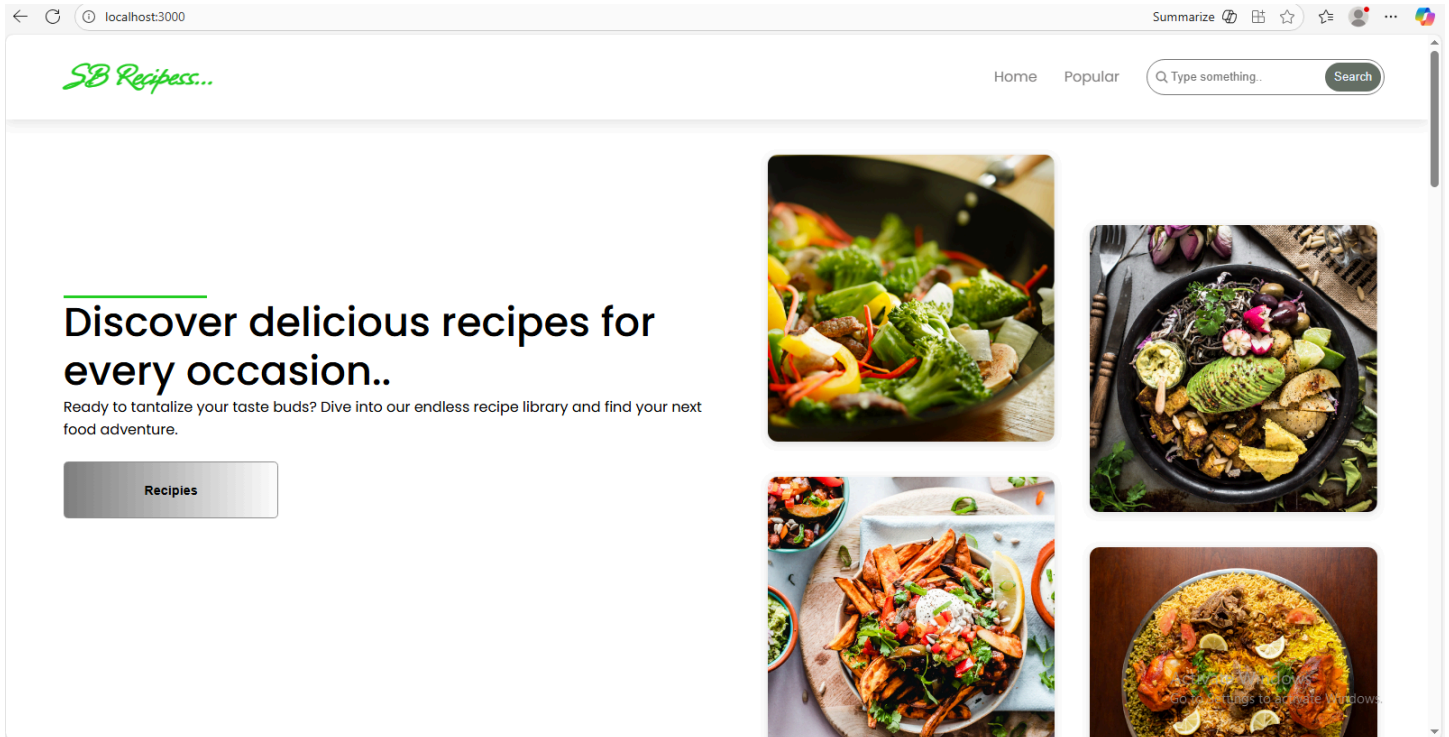- **Popular:**

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**
- **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.