**COOKBOOK :**

**Your Virtual Kitchen Assistant**

**Team ID SWTID1741166376153712**

**Team Size 4**

**Team Leader** ABHI PRITI R

**Team member** DIVYA PRIYA A

**Team member** HARINI K

**Team member** HARINI S

**Project Overview**

**Purpose:**

Cookbook serves as an intelligent virtual kitchen assistant, aiding users in discovering, organizing, and managing their recipes. It provides detailed cooking instructions, tracks ingredients, and facilitates meal planning to improve the overall cooking experience.

**Features:**

* Advanced recipe search with customizable filters
* Guided, step-by-step cooking instructions
* Integrated cooking tutorial videos
* Smart meal planning and scheduling system
* Personalized user profiles with saved recipe collections

**Architecture**

**Component Structure:**

 **App.js** – Serves as the core application component

 **Header.js** – Manages the navigation menu

 **RecipeList.js** – Displays all available recipes

 **RecipeDetails.js** – Presents comprehensive details of each recipe

 **ShoppingList.js** – Keeps track of required ingredients

 **MealPlanner.js** – Helps users organize their meals effectively

**State Management:**

The application employs the **Context API** to maintain a global state, managing user preferences, saved recipes, and ingredient inventories.

**Routing:**

Navigation across various sections such as the home page, recipe details, shopping list, and meal planner is facilitated using **React Router**.

**Setup Instructions**

**Prerequisites:**

Node.js installed (version14+ recommended)

**Installation:**

1. Clone the repository:

git clone https://github.com/yourusername/cookbook.git

1. Move into the project directory:

cd cook

1. Install necessary dependencies:

npm install

1. Start the application:

npm start

**Folder Structure**

### **Frontend (Client-Side):**

* **src/components** – Houses all React components
* **src/pages** – Contains different screens of the app
* **src/assets** – Stores images, icons, and styling elements

### **Utilities:**

* **src/utils/api.js** – Manages API requests
* **src/utils/helpers.js** – Provides helper functions for various tasks

**Running the Application:**

To start the application, run:

npm start

**Component Documentation**

**Key Components:**

* RecipeCard.js - Displays a summary of a recipe
* RecipeDetails.js - Shows detailed recipe information
* IngredientList.js - Manages shopping lists

**Reusable Components:**

* Button.js - Customizable button component
* Modal.js - Reusable modal for alerts and pop-ups

**State Management**

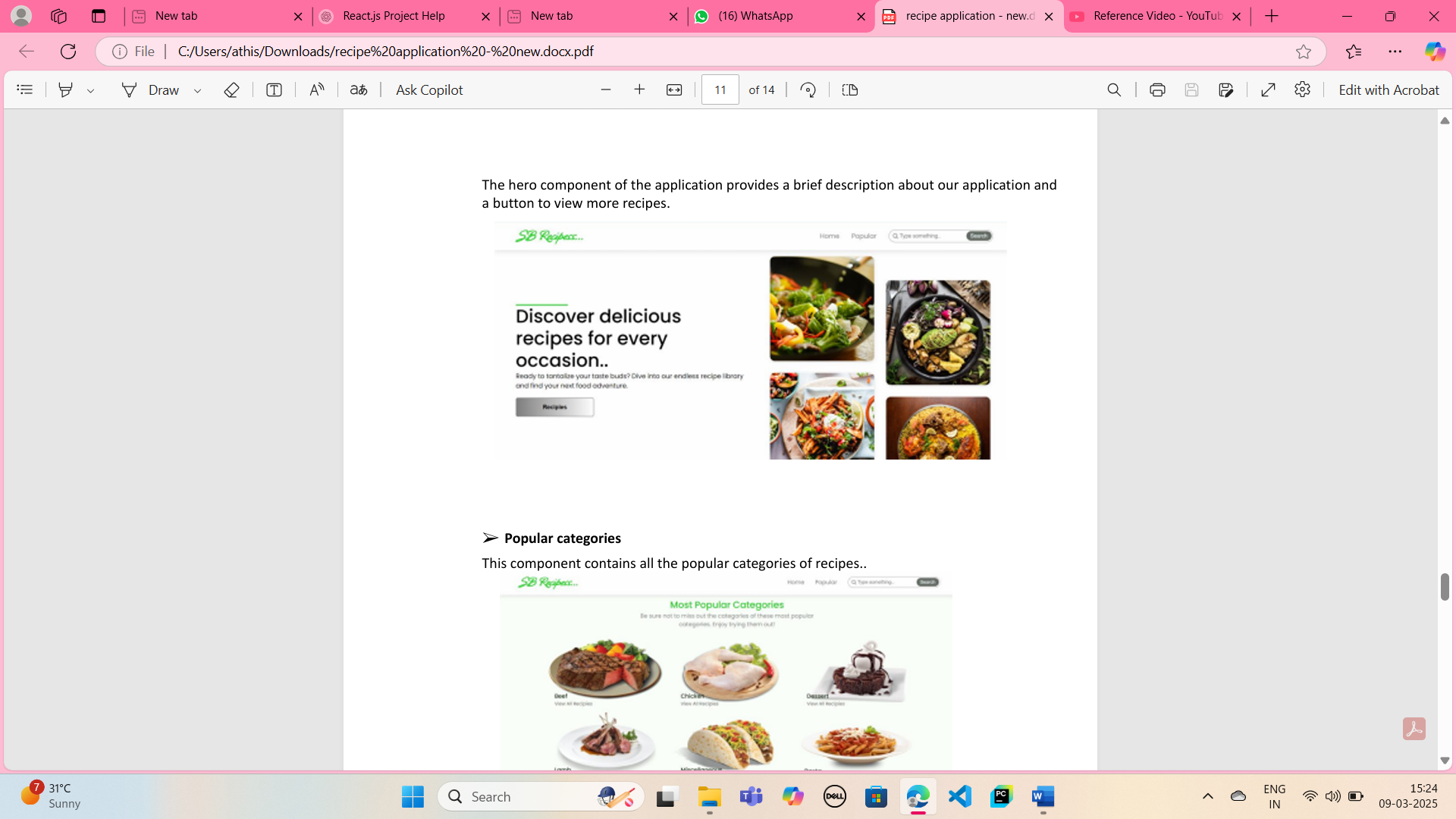
### **Global State:**

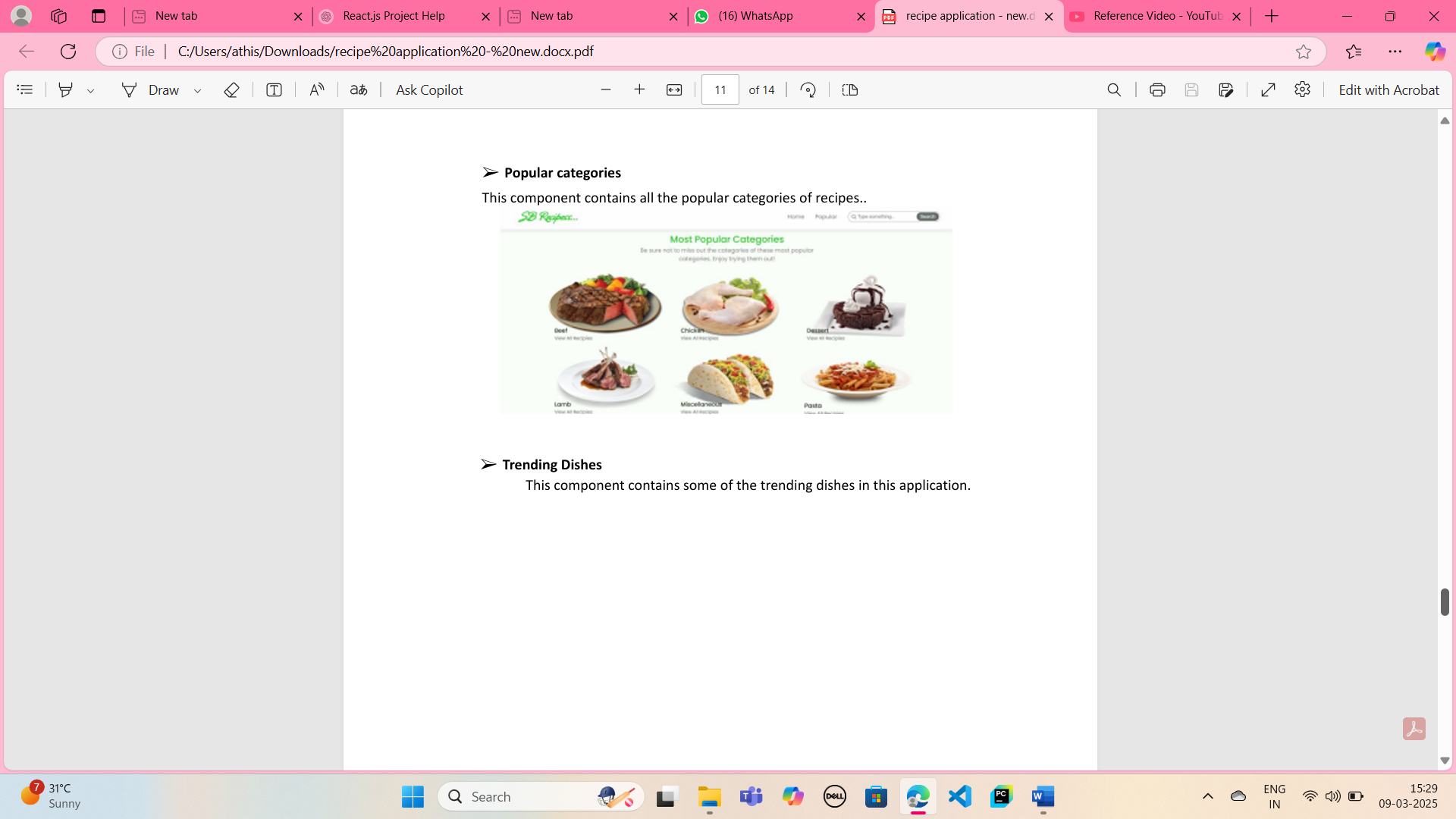
The **Context API** is utilized to handle user preferences, saved recipes, and shopping lists efficiently.

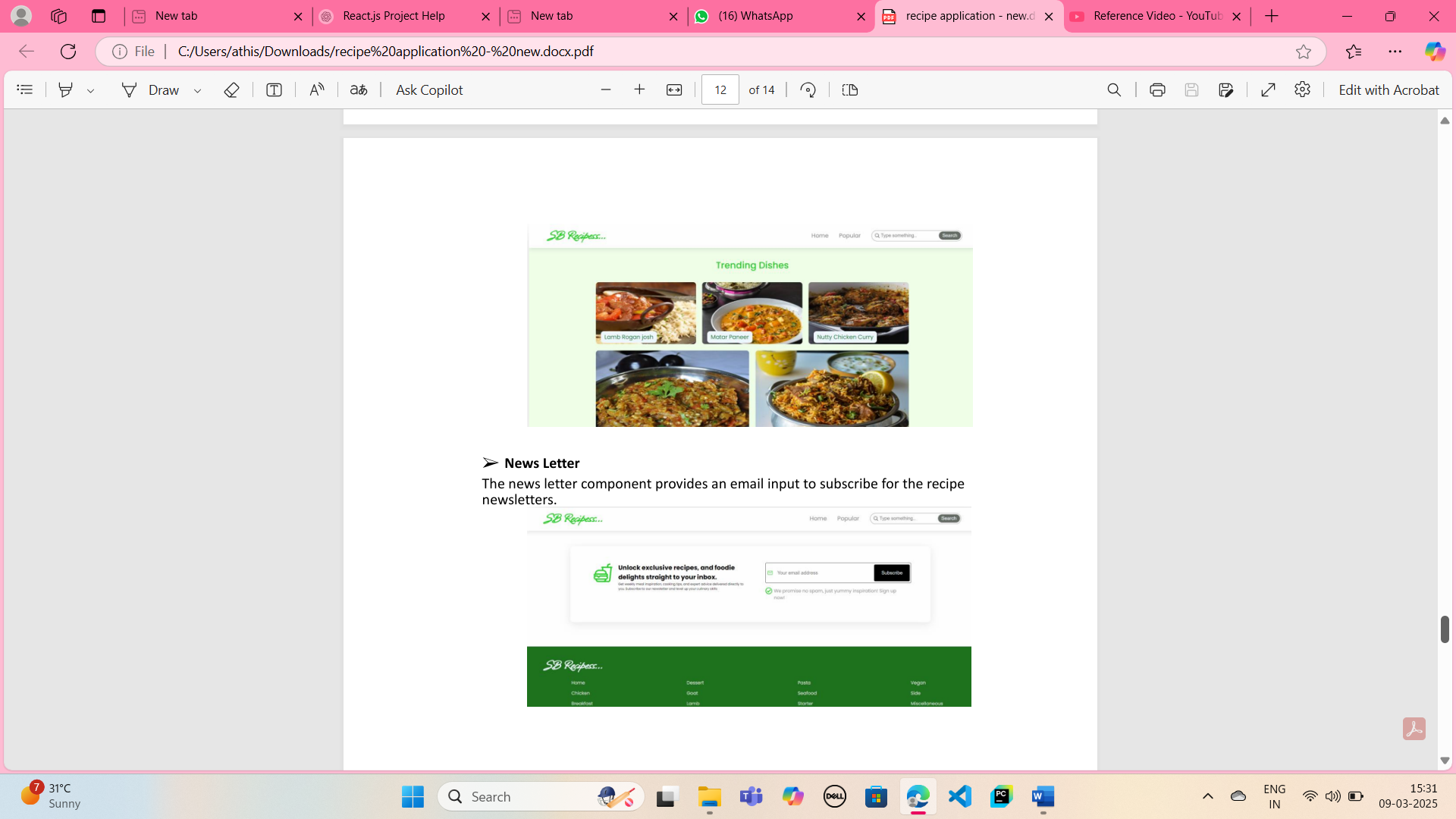
### **Local State:**

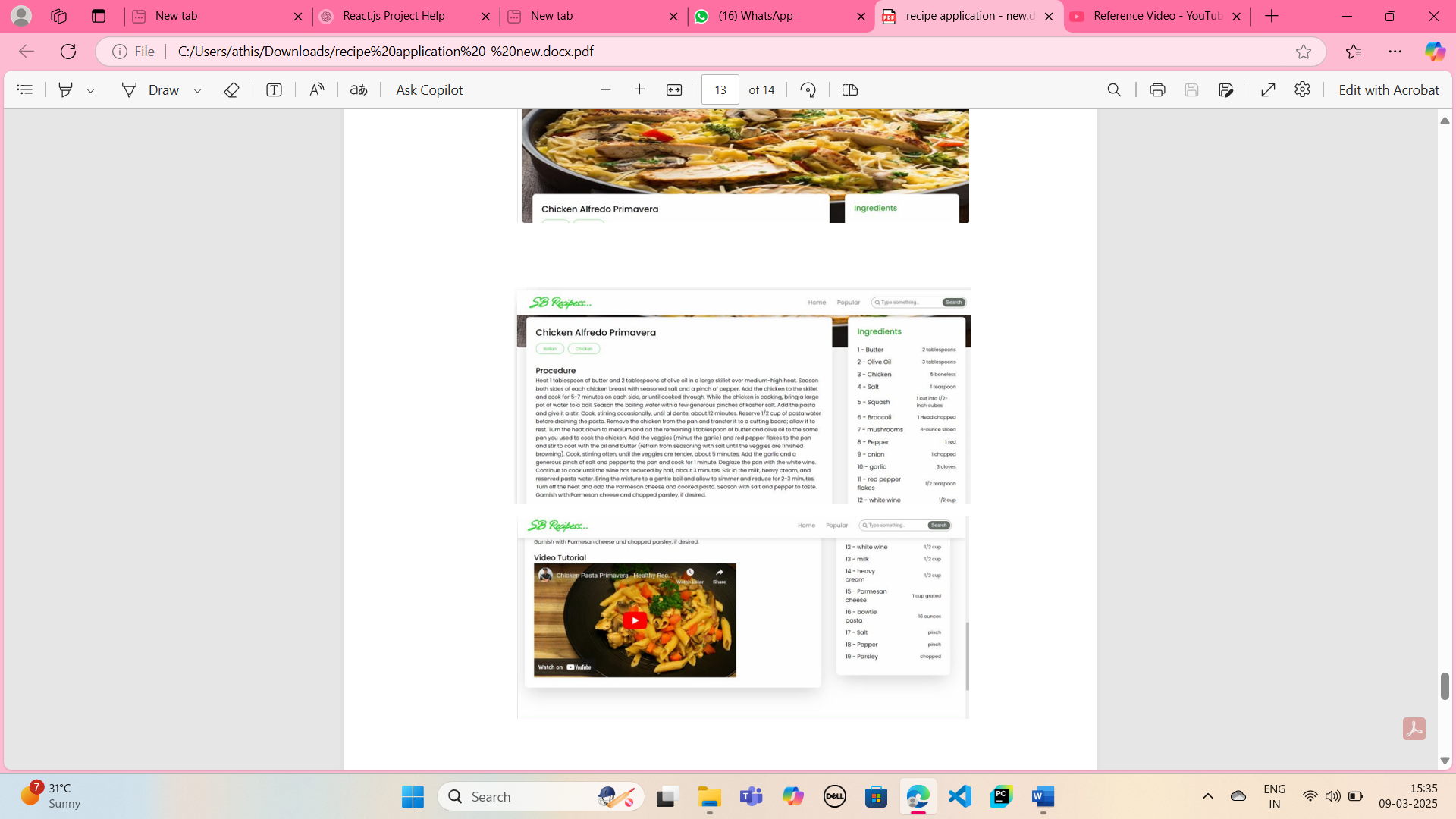
The **useState** hook is implemented for managing form inputs and user interactions within the UI.

**User Interface**









**Styling**

**CSS Frameworks/Libraries:**

* Using Styled-Components for modular styling
* Material UI for pre-designed components

**Theming:**

* Light and Dark mode support
* Custom color palette for branding

**Testing**

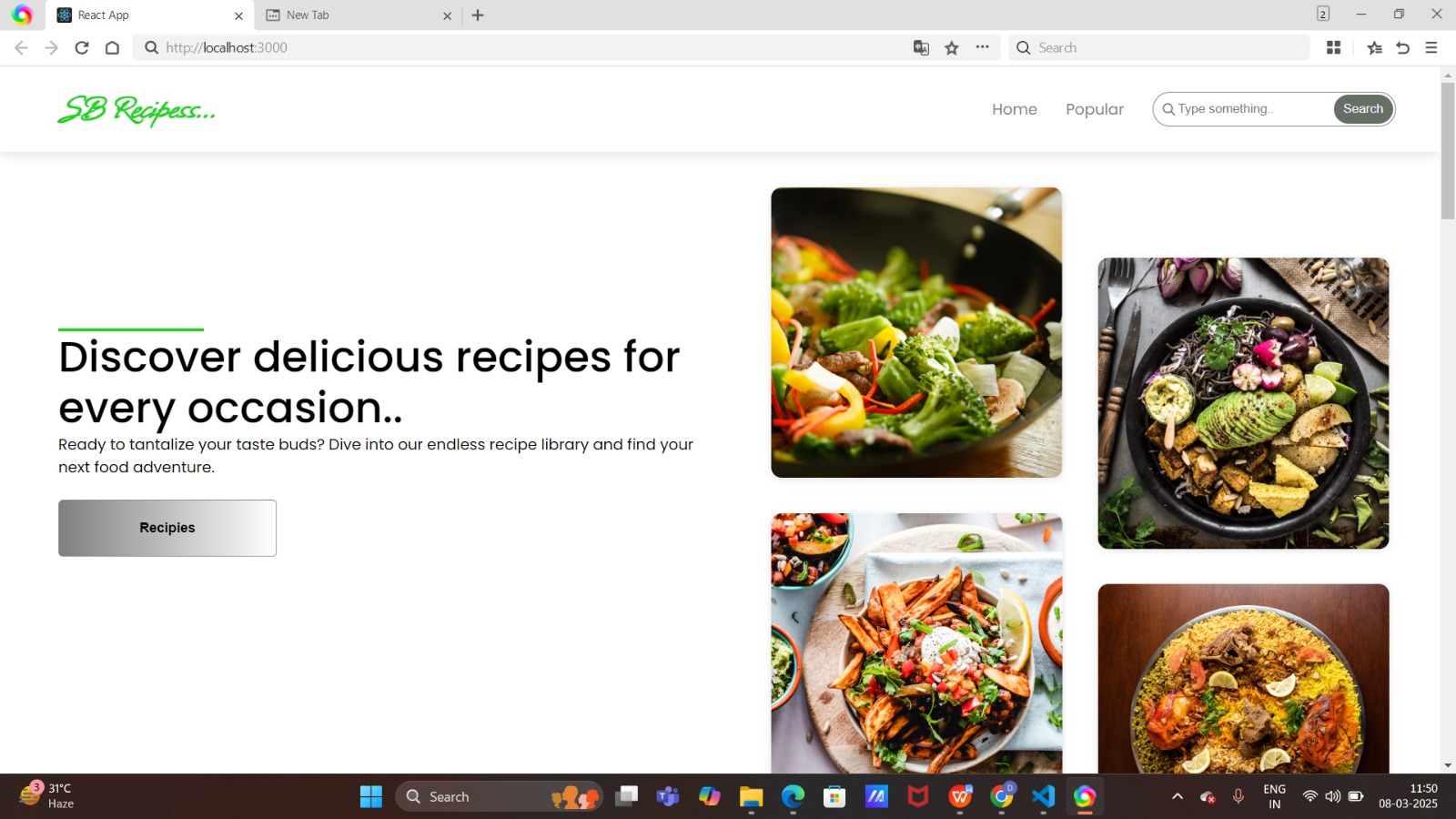
**Testing Strategy:**

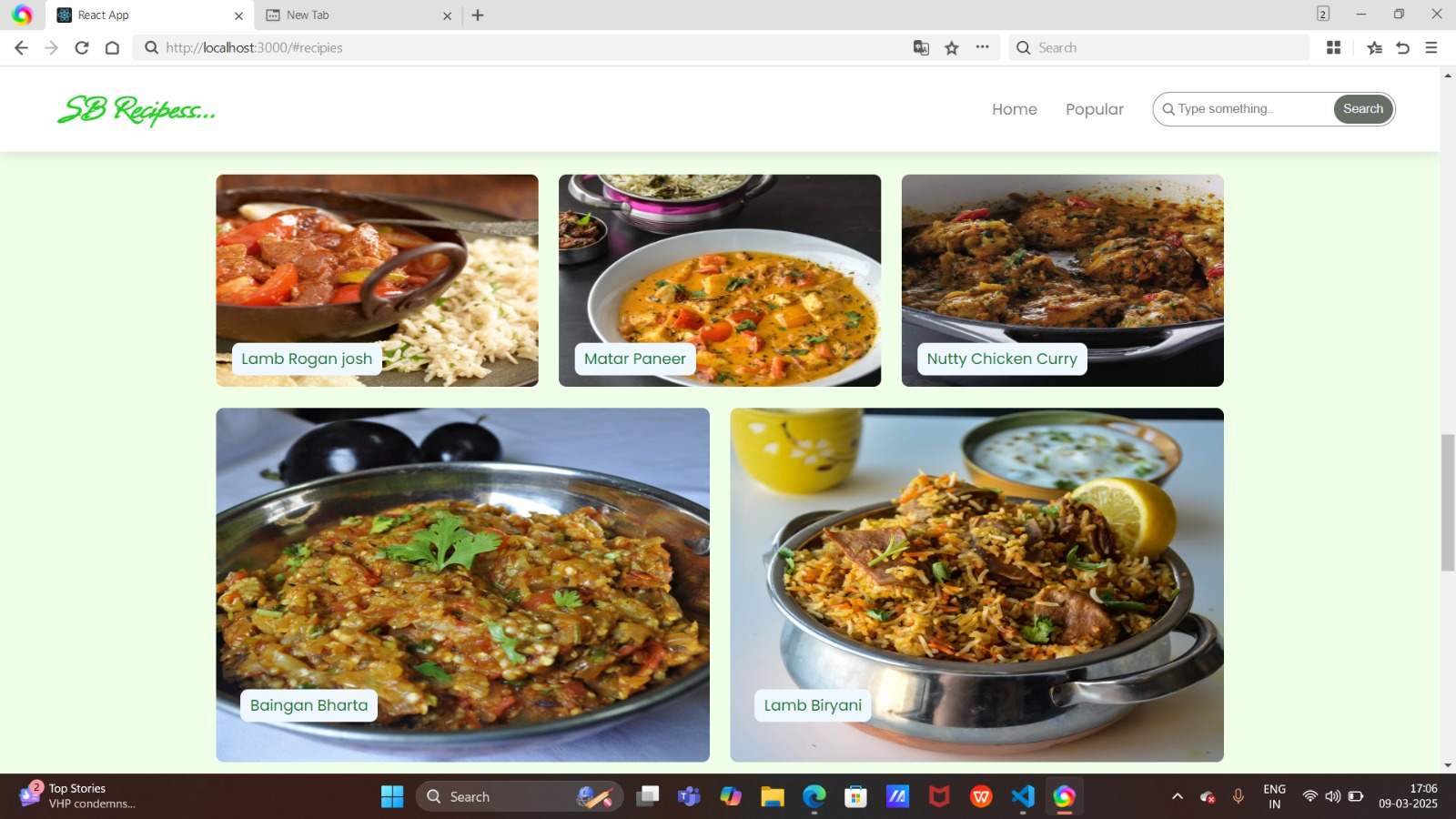
* Unit Testing using Jest
* Component Testing with React Testing Library

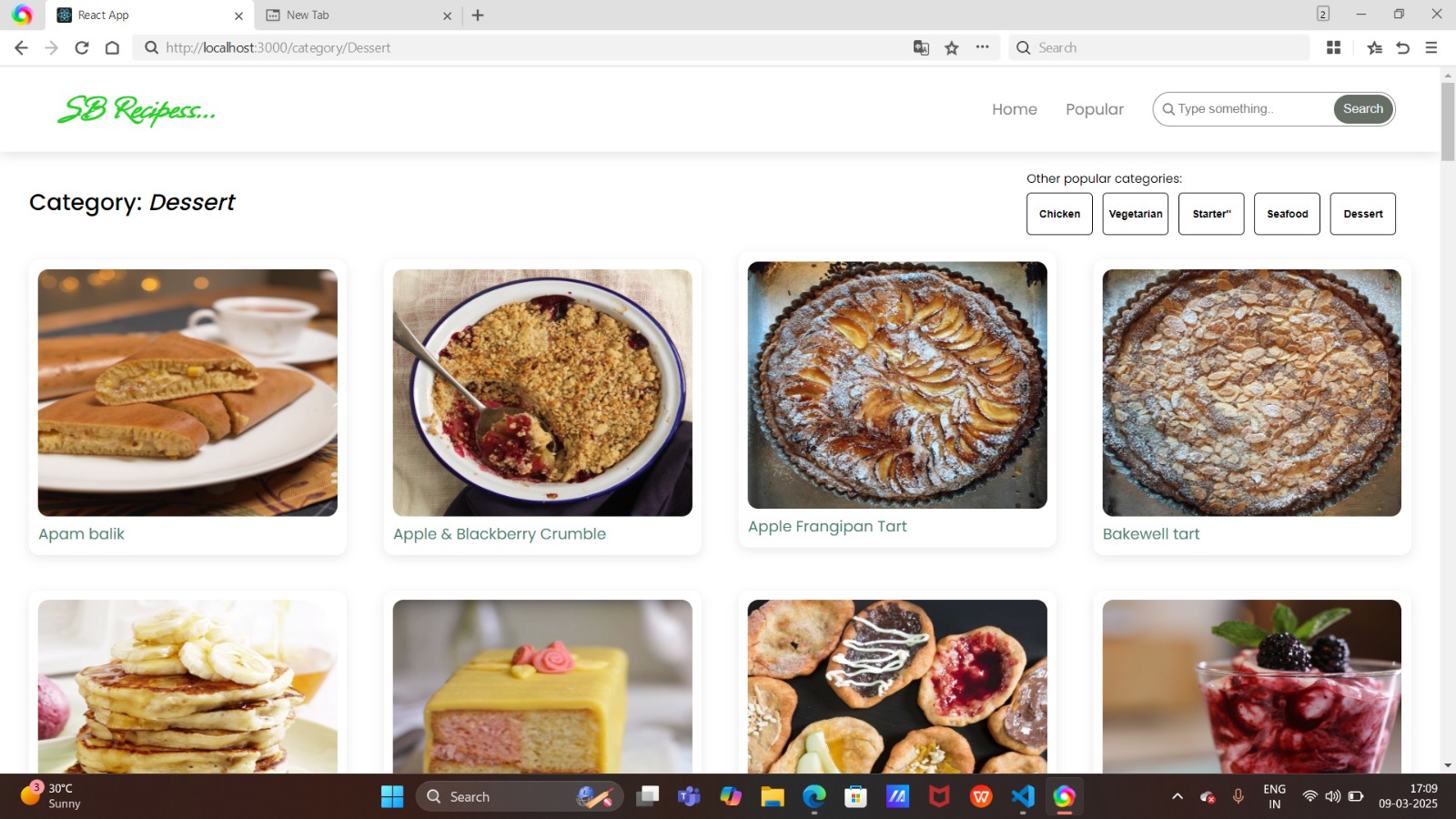
**Code Coverage:**

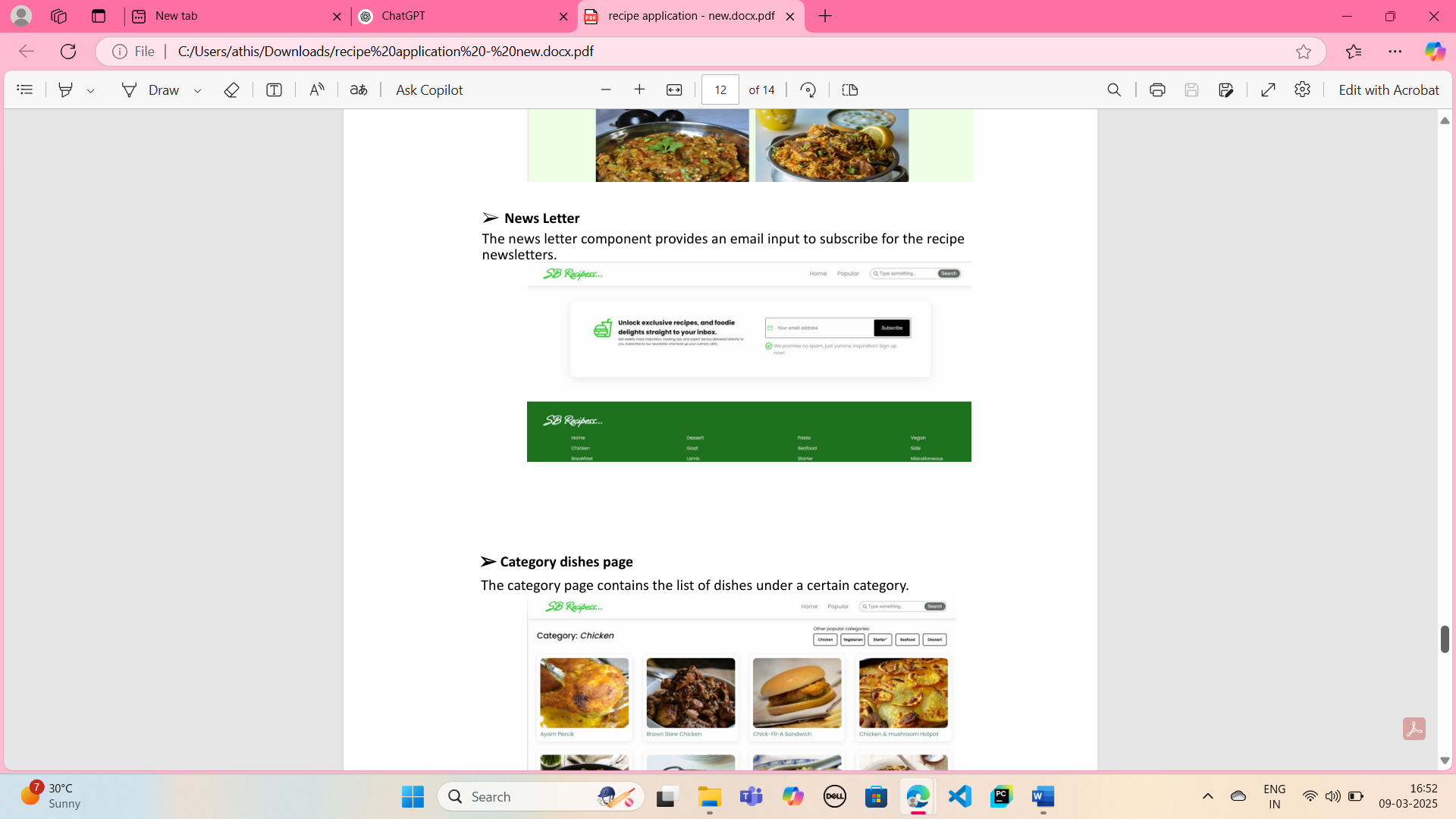
* Coverage reports generated using Jest

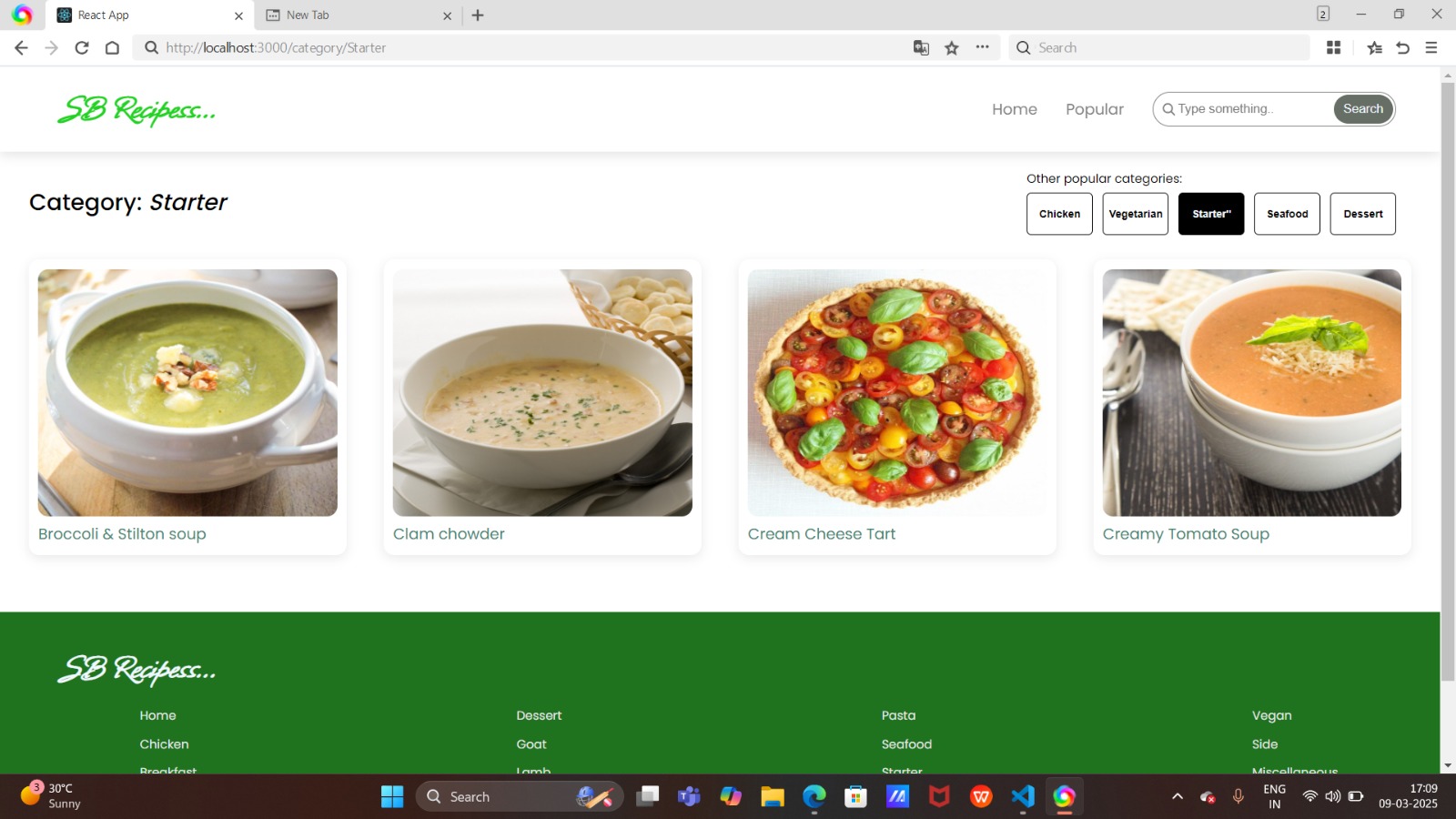
**Screenshots**

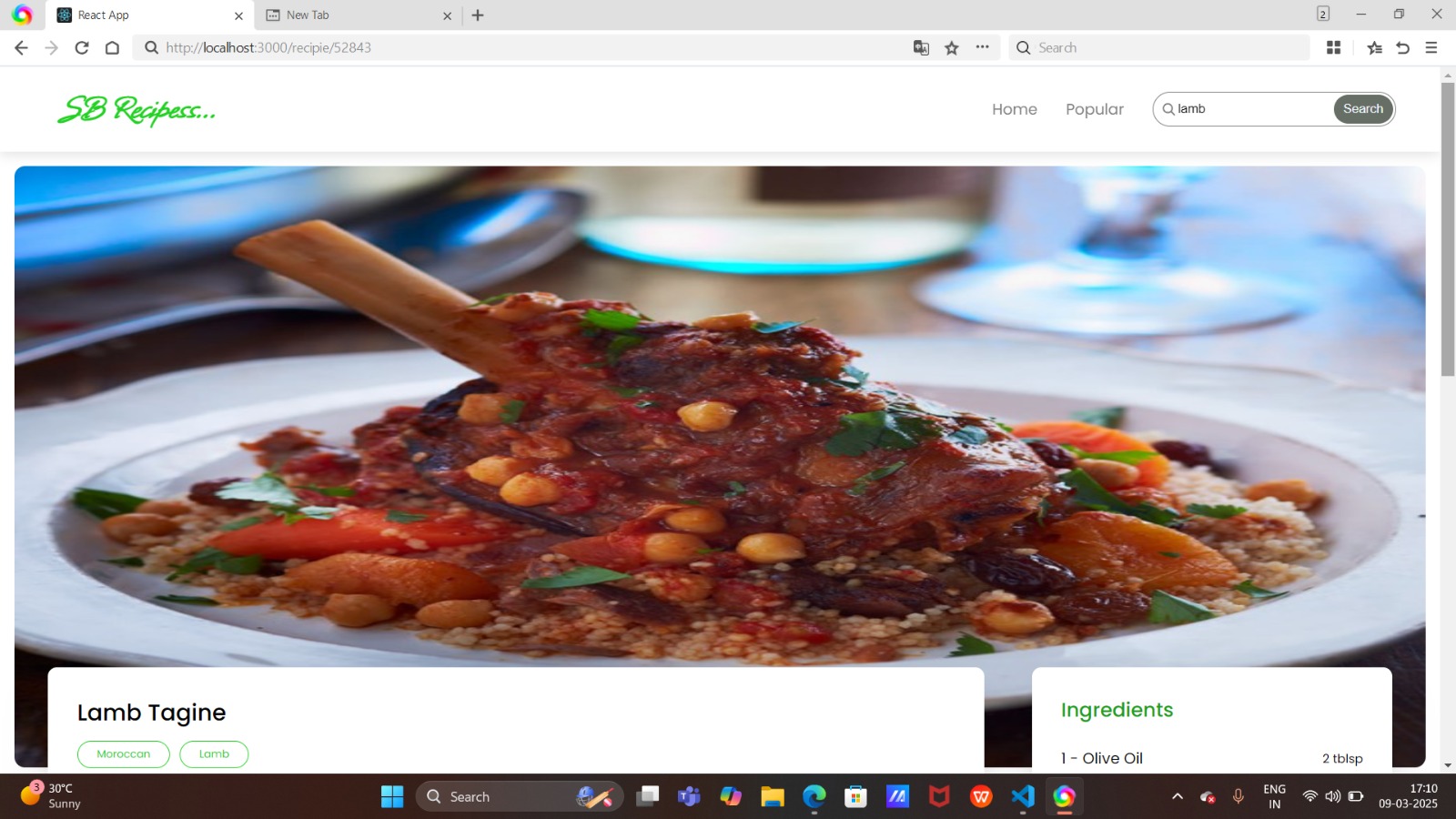








****



**Known Issues**

* Some recipes might not load properly due to API limitations.
* Performance optimization needed for large lists

**Future Enhancements**

* Adding voice command support for hands-free cooking
* Integration with smart kitchen devices
* Personalized recipe recommendations based on user preferences