**COOKBOOK : Your Virtual Kitchen Assistant**

Team Members:

Athishya .V

Aarthi .T

Kaviya .A

Dhanalakshmi .P

**Project Overview**

**Purpose:**

Cookbook is a virtual kitchen assistant that helps users discover, manage, and organize recipes. It provides step-by-step cooking instructions, ingredient tracking, and meal planning features to enhance the cooking experience.

**Features:**

* Recipe search and filtering
* Step-by-step cooking instructions
* Cooking videos
* Meal planning and scheduling
* User profile and saved recipes

**Architecture**

**Component Structure:**

* App.js - Main application component
* Header.js - Navigation bar
* RecipeList.js - Displays all available recipes
* RecipeDetails.js - Shows detailed recipe information
* ShoppingList.js - Manages ingredient lists
* MealPlanner.js - Allows users to plan meals

**State Management:**

Using Context API for global state management to manage user preferences, saved recipes, and ingredient tracking**.**

**Routing:**

Utilizing React Router for navigation between pages, including home, recipe details, shopping list, and meal planner.

**Setup Instructions**

**Prerequisites:**

Node.js installed (version14+ recommended)

**Installation:**

1. Clone the repository:

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

1. Navigate to the project directory:

cd cookbook

1. Install dependencies:

npm install

1. Start the application:

npm start

**Folder Structure**

**Client:**

* src/components - Contains React components
* src/pages - Different pages of the application
* src/assets - Images, icons, and styles

**Utilities:**

* src/utils/api.js - Handles API calls
* src/utils/helpers.js - Contains helper functions

**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**

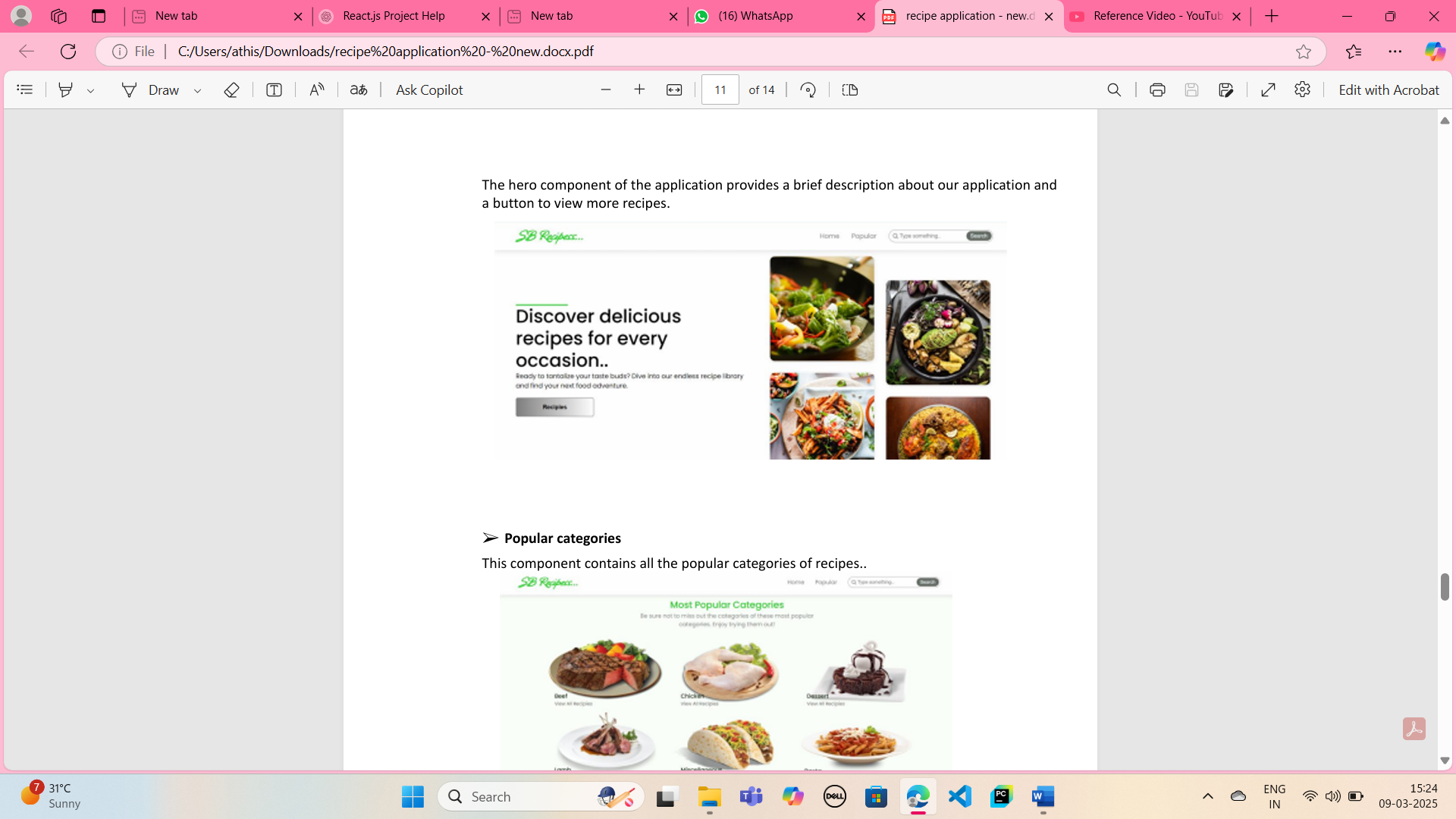
**Global State:**

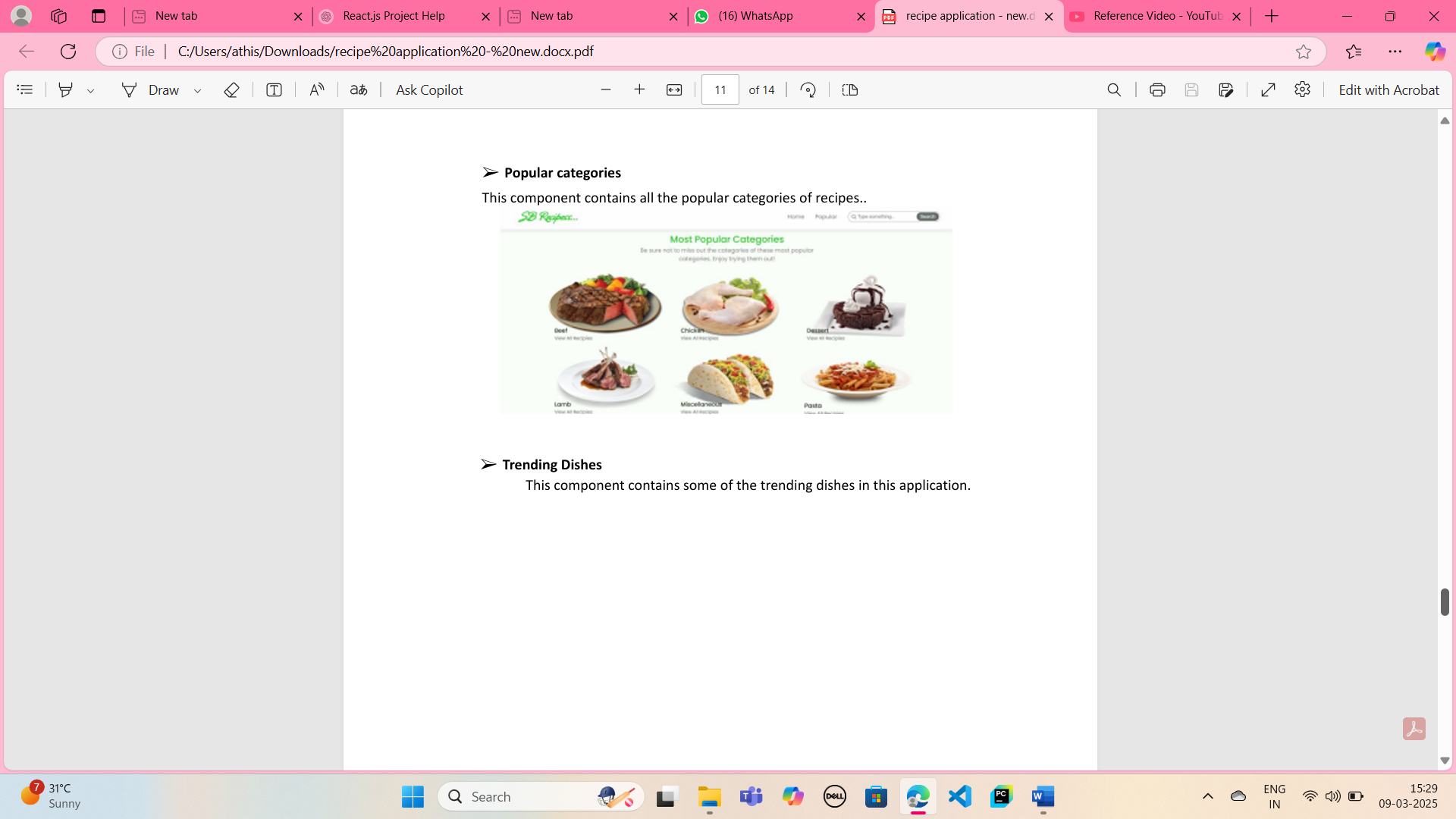
Managed using Context API to handle user preferences, saved recipes, and shopping lists.

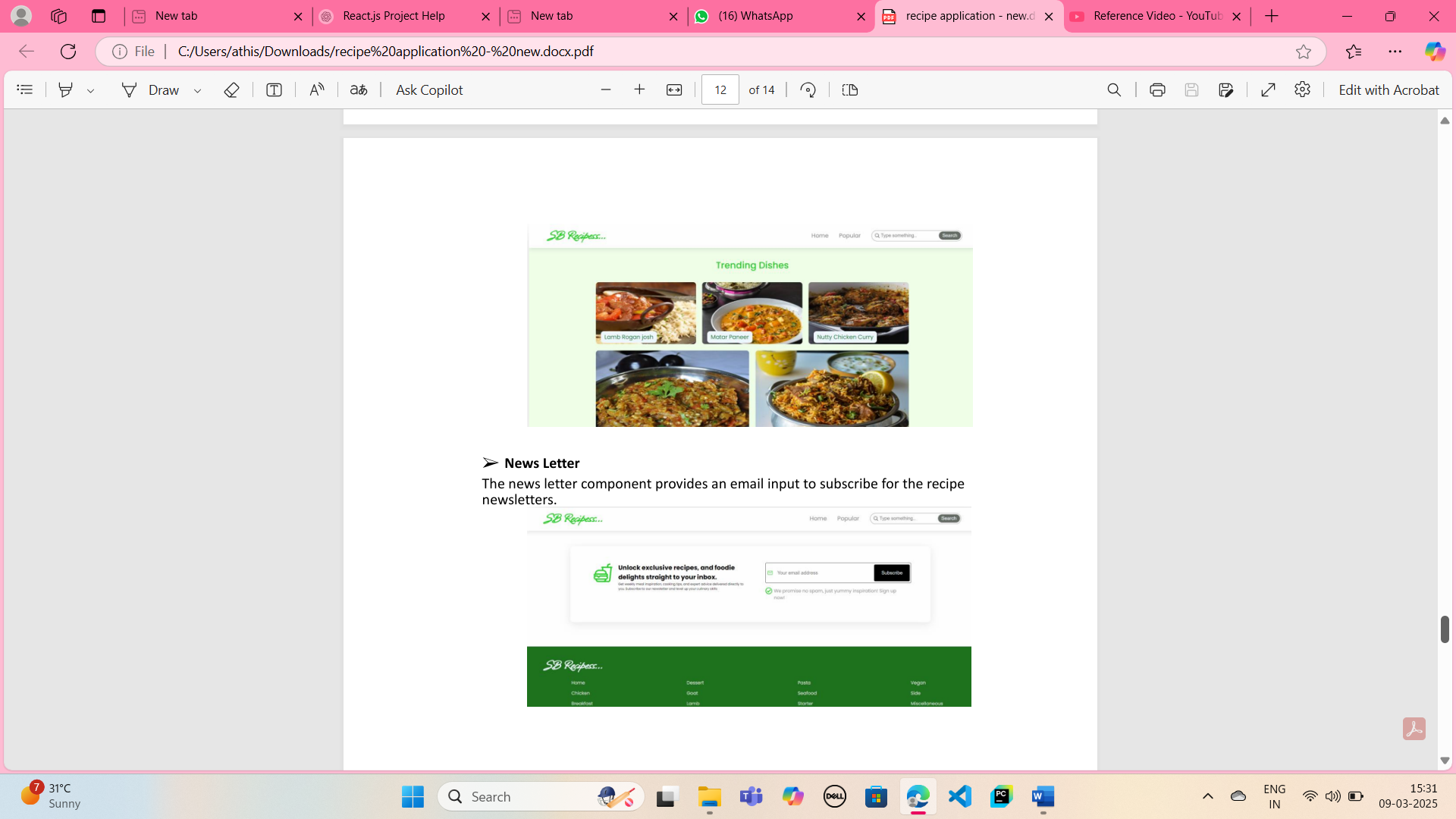
**Local State:**

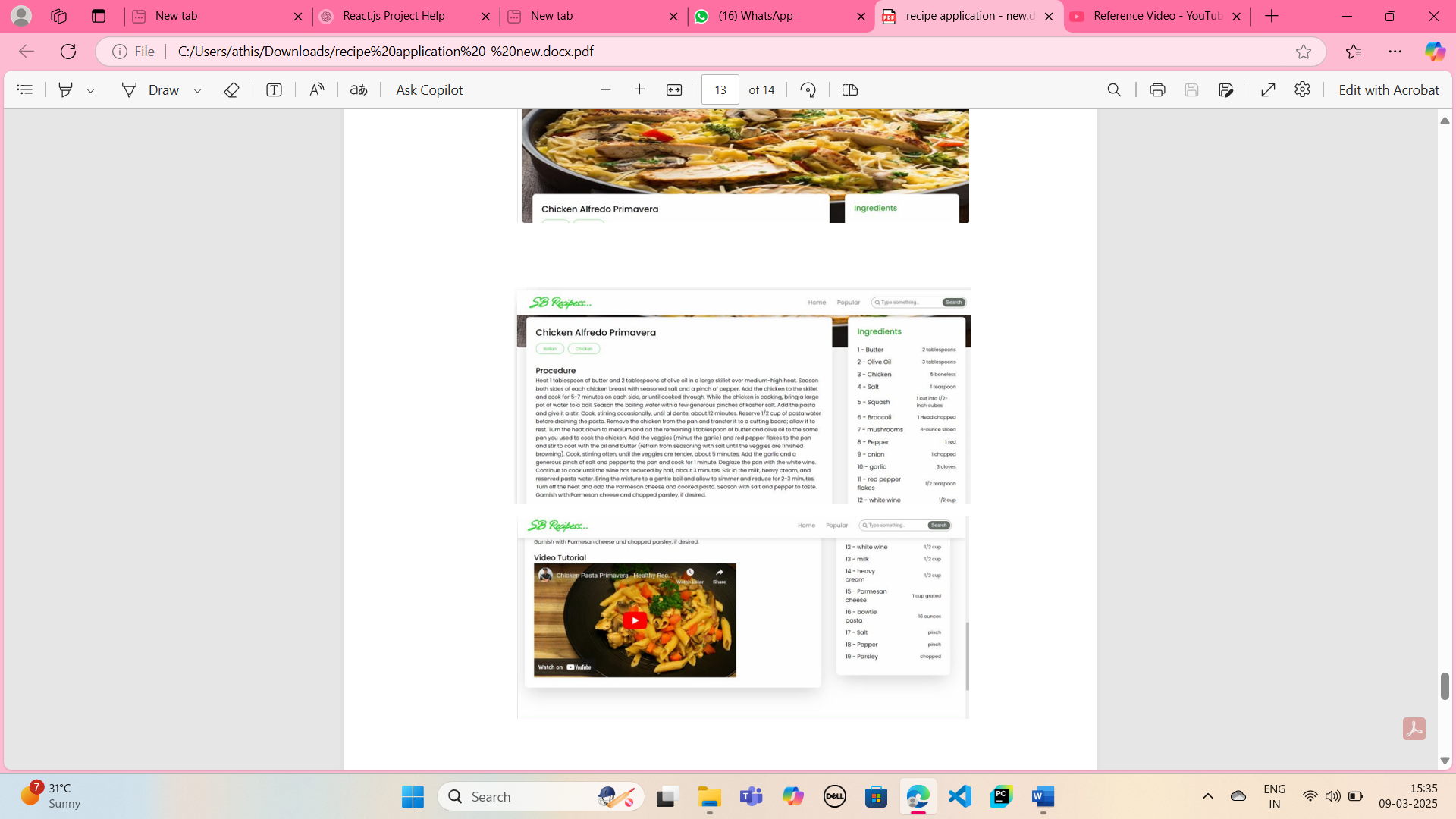
Use of useState for handling form inputs and UI interactions.

**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**

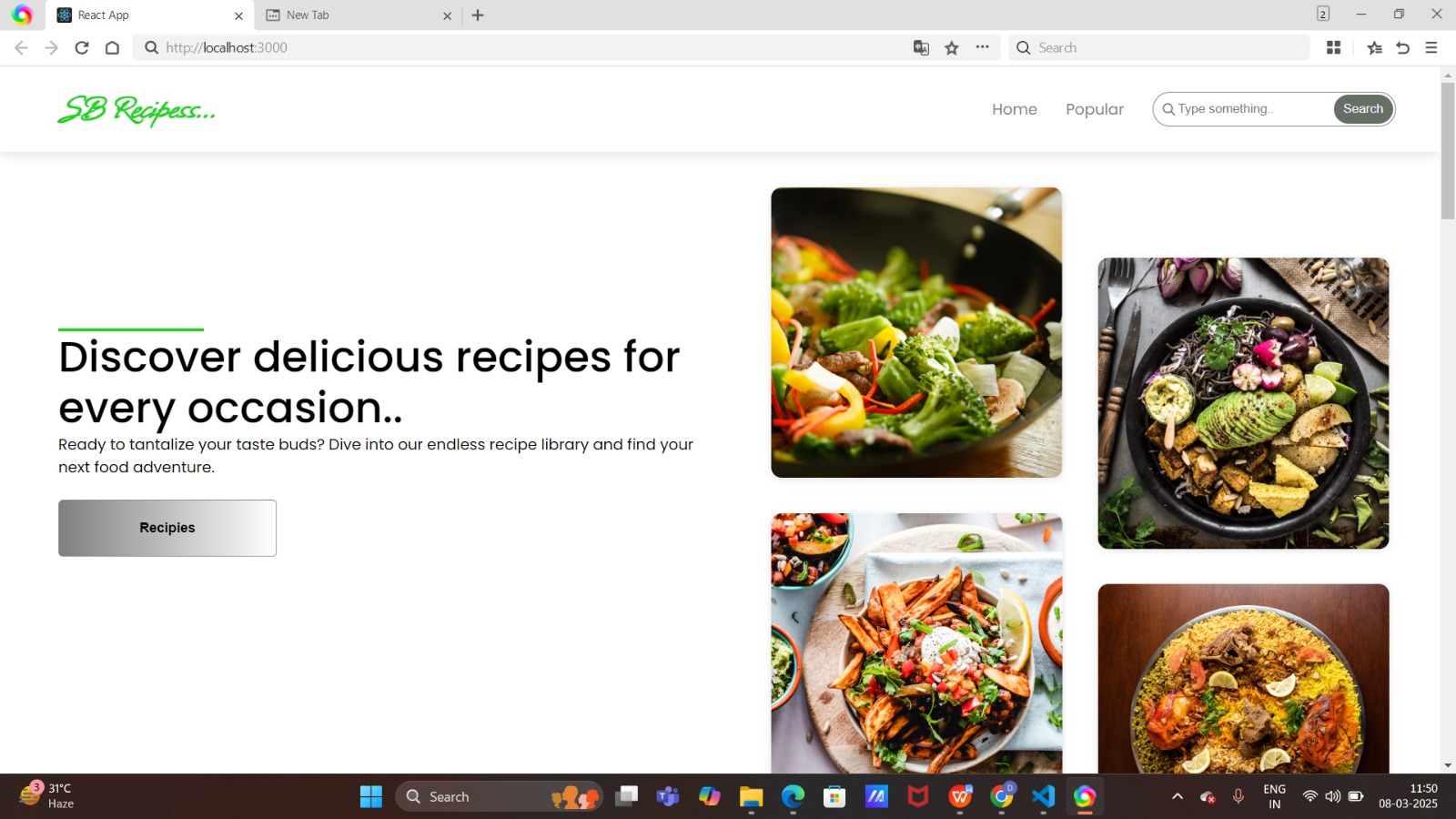
**Testing Strategy:**

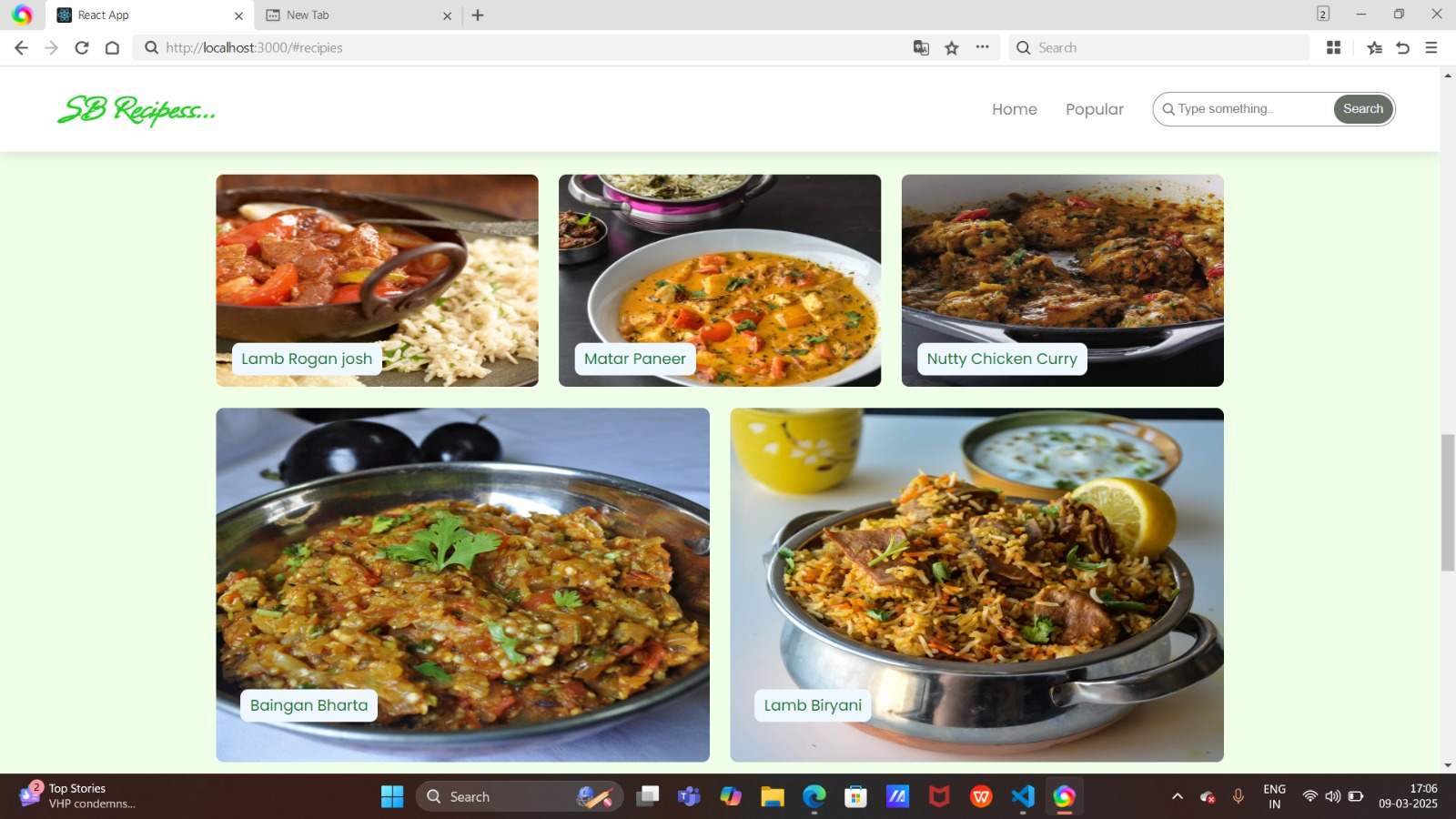
* Unit Testing using Jest
* Component Testing with React Testing Library

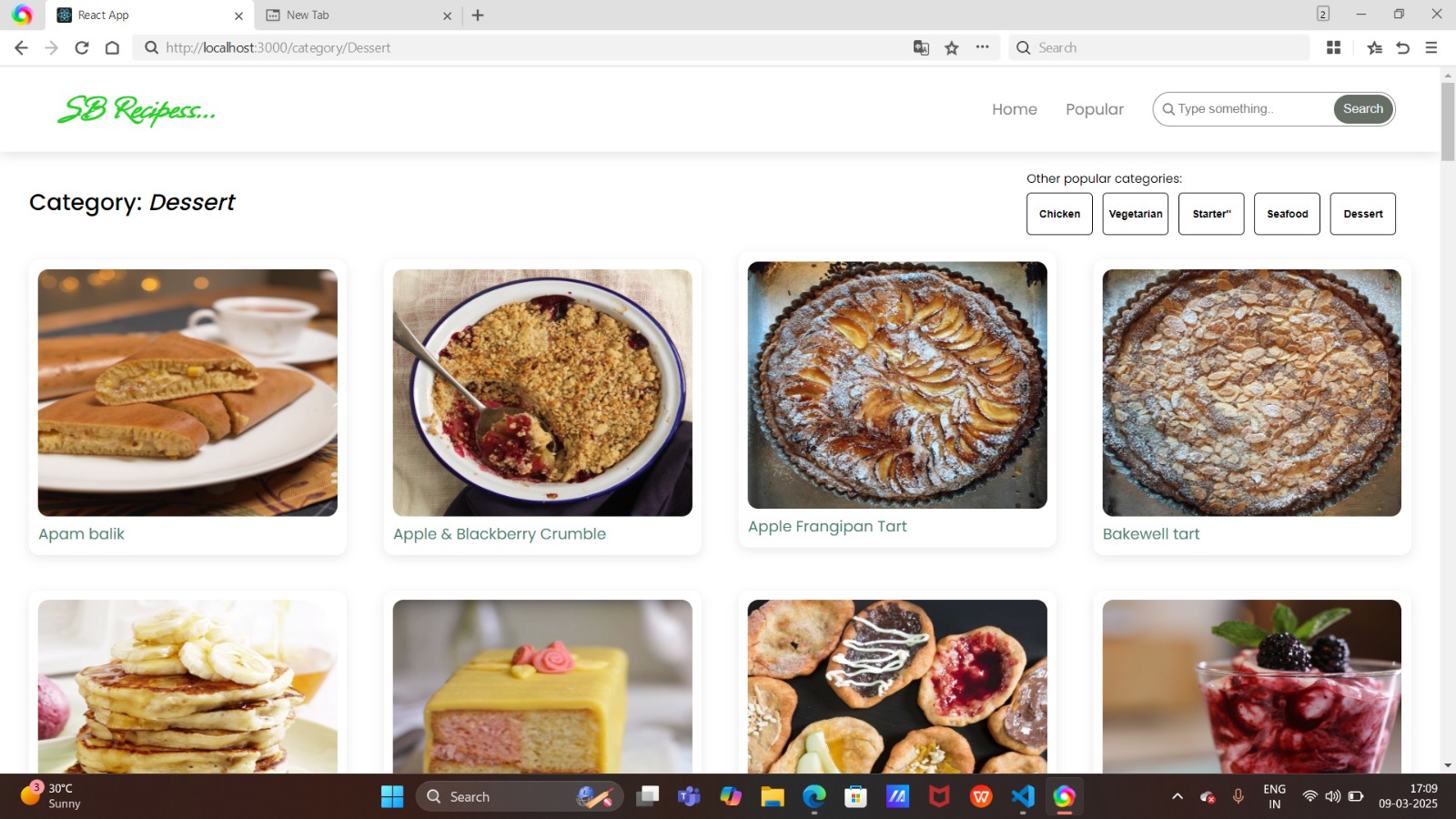
**Code Coverage:**

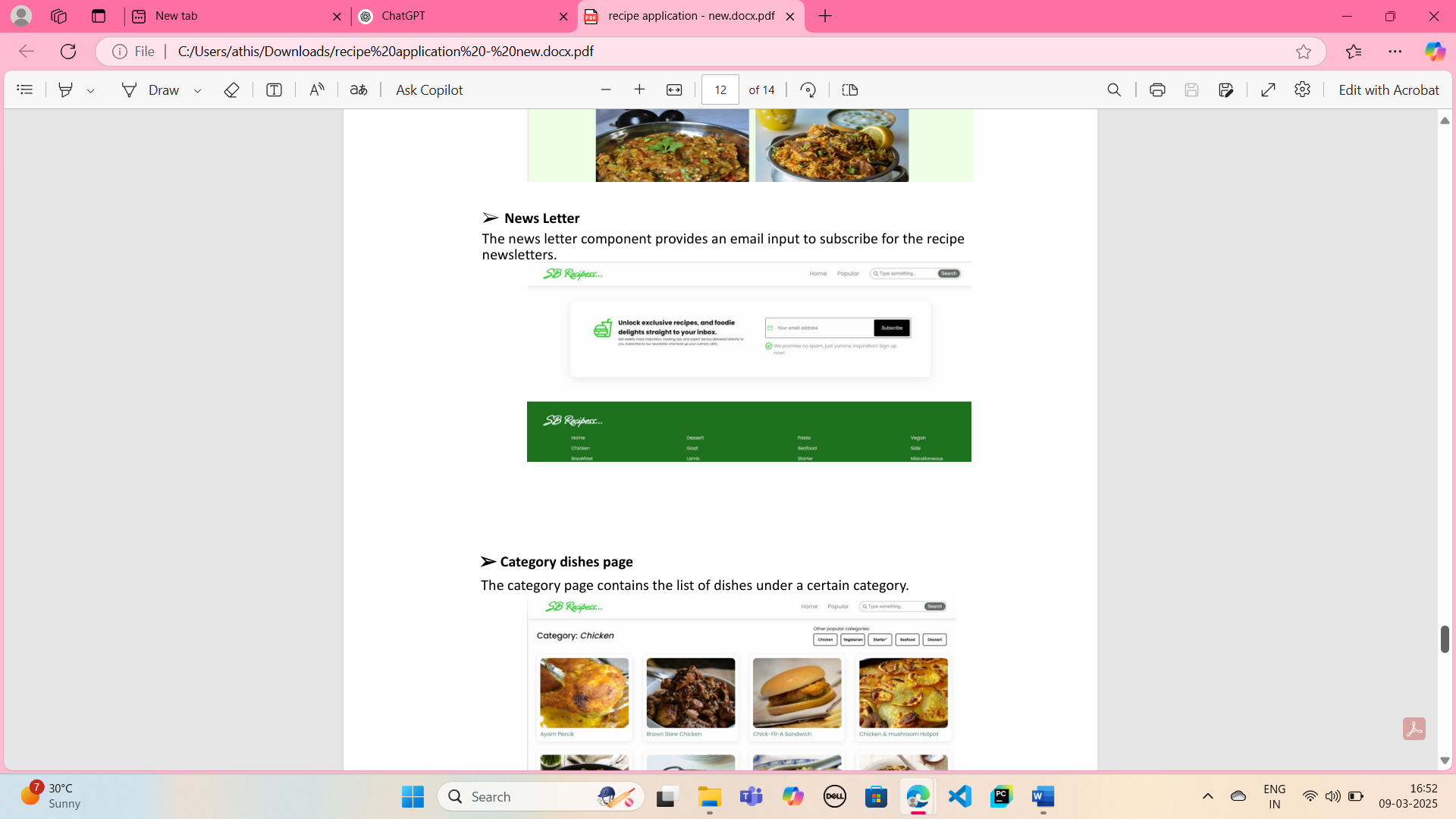
* Coverage reports generated using Jest

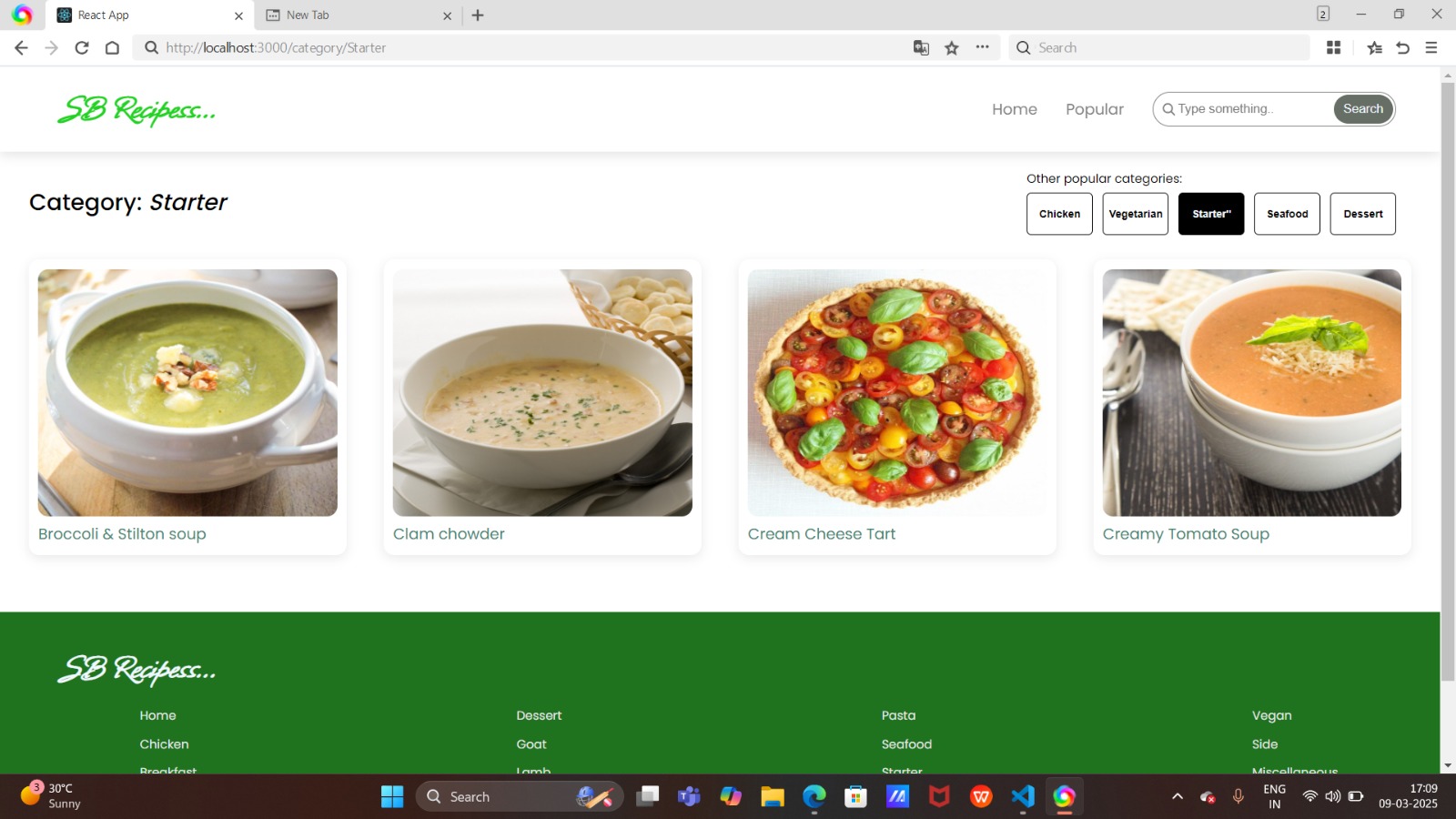
**Screenshots or Demo**

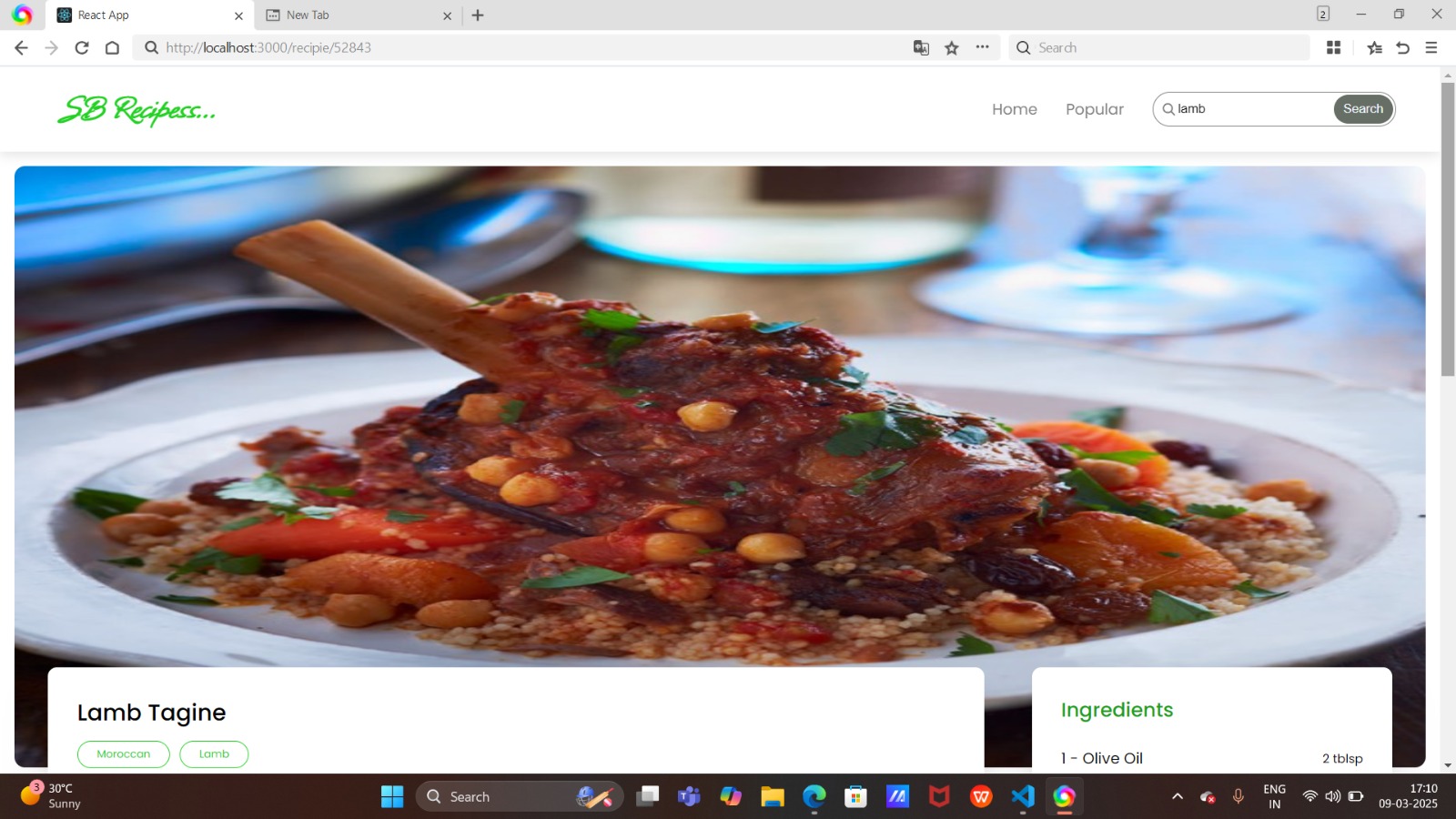








****



**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