# COOK BOOK: Your Virtual Kitchen Assistant

# (React Application)

**INTRODUCTION:**

**Project Title:**

CookBook is a revolutionary web application designed to change the way you discover, organize, and create recipes. It caters to both novice and professional chef offering a user-friendly interface, robust features, and a vast collection of inspiring recipes.

**Team Members:**

|  |  |  |
| --- | --- | --- |
| **TEAM MEMBERS** | **E-MAIL ID** | **ROLE** |
| PAVITHRA N | npavithra08122004@gmail.com | Team Leader |
| HARINI S | harinisuji27@gmail.com | Team Member |
| LITHIKA S | lithika09061980@gmail.com | Team Member |
| SWATHI D | swathideena865@gmail.com | Team Member |

**Project Overview:**

**Purpose:**

The Cookbook Recipes App is designed to provide users with a comprehensive and user-friendly platform for discovering, saving, and cooking recipes. The app aims to simplify meal planning, grocery shopping, and cooking by offering a robust search functionality, personalized features, and a vast collection of recipes

**Features:**

1. Recipe Library: A comprehensive collection of recipes from around the world, including images, ingredients, instructions, and cooking times.

2.Search and Filter: Robust search functionality to help users find recipes based on ingredients, cooking time, and dietary preferences.

3. Meal Planning: Personalized meal planning feature to help users plan and organize their meals for the week.

4. Grocery Lists: Automatic generation of grocery lists based on selected recipes.

5. Recipe Bookmarking: Users can bookmark and save their favorite recipes for easy access.

6. Social Sharin: Users can share recipes and cooking achievements on social media platforms.

7.User Profile: Users can create a profile to save their preferences, bookmarks, and meal plans.

8. Recipe Submission: Users can submit their own recipes for moderation and inclusion in the recipe library.

9. Rating and Review: Users can rate and review recipes to help others make informed decisions.

10. Notification System: Users receive notifications for new recipe submissions, bookmarked recipes, and meal planning reminders.

**Architecture:**

**Components structure:**

The Cookbook Recipes App consists of the following major React components:

1. App: The top-level component that renders the entire app.

2. Header: The header component that contains the navigation bar and search bar.

3. Recipe List: The component that displays a list of recipes.

4. Recipe Detail: The component that displays detailed information about a single recipe.

5. Meal Plan: The component that allows users to plan and organize their meals.

6. Grocery List: The component that generates a grocery list based on selected recipes.

7. User Profile: The component that displays the user’s profile information and saved recipes.

**State Management:**

The Cookbook Recipes App uses the Context API for state management. The Context API provides a way to share data between components without passing props down manually.

**Routing:**

The Cookbook Recipes App uses React Router for client-side routing. The app has the following routes:

1. /: The homepage that displays a list of popular recipes.

2. recipes: The page that displays a list of all recipes.

3. recipes:id: The page that displays detailed information about a single recipe.

4. meal-plan: The page that allows users to plan and organize their meals.

5. grocery-list: The page that generates a grocery list based on selected recipes.

6. user-profile: The page that displays the user’s profile information and saved recipes.

**Routing Library:**

The Cookbook Recipes App uses React Router v6, which provides a powerful and flexible way to manage client-side routing.

**Setup Instruction**

**Prerequisites:**

Software Prerequisites:

This could mean the programming languages you need to know (like Python, HTML, CSS, JavaScript).

It might also include specific software or databases required (like MySQL or other database systems).

For example, if the cookbook project is a website, then knowledge of HTML, CSS and Javascript would be a prerequisite

Hardware Prerequisites:

This would involve the computer and any other devices needed for development.

Knowledge Prerequisites:

Understanding of measurements and culinary terms.

Perhaps knowledge of specific cuisines.

Material Prerequisites:

A computer or notebook for compiling recipes.

Access to recipe sources (cookbooks, websites, etc.).

the cookbook includes photos, a camera.

**Installation:**

Step 1: Clone the Repository

Git clone

Step 2: Install Dependencies

npm install

Step 3: Configure Environment Variables

Create a new file named .env:

Step 4: Start the App\*

npm start

Step 5: Start the MongoDB Server

Step 6: Start the Backend Server

Navigate into the `backend` directory:

Node server.js

**Folder Structure**

**Client**

components: Reusable React components, such as buttons, forms, and cards.

pages: Top-level React components that represent individual pages, such as the homepage, recipe list, and recipe detail.

assets: Static assets, such as images, fonts, and CSS files.

utils: Helper functions, utility classes, and custom hooks used throughout the app.

hooks: Custom React hooks used to manage state and side effects.

context: React Context API providers and consumers used to share data between components.

routes: React Router configuration and route definitions.

**Utilities**

Helper Functions

- api.js: A helper function to make API requests to the backend server.

- format.js: A helper function to format dates, times, and numbers.

- string.js: A helper function to manipulate strings.

Utility Classes

- Recipe.js: A utility class to represent a recipe, including its ingredients, instructions, and metadata.

- User.js: A utility class to represent a user, including their profile information and saved recipes.

Custom Hooks

- useRecipe.js: A custom hook to fetch and manage recipe data.

- useUser.js: A custom hook to fetch and manage user data.

- useAPI.js: A custom hook to make API requests and handle errors.

Context API

- RecipeContext.js: A React Context API provider and consumer to share recipe data between components.

- UserContext.js: A React Context API provider and consumer to share user data between components.

**Running the Application:**

Frontend:

To start the frontend server locally, navigate to the client directory and run:

npm start:

This will start the React development server, and you can access the Cookbook Recipes App at http://localhost:3000.

Accessing the App:

Once the frontend server is running, you can access the Cookbook Recipes App by navigating to

webpage

<http://localhost:3000>

This will take you to the app's homepage, where you can browse and search for recipes.

**Component Documentation**

Key Components:

1. App Component:

Purpose: The top-level component that renders the entire app

2. Recipe List Component:

Purpose: Displays a list of recipes.

On Delete: A callback function to handle recipe deletion.

Description: The Recipe List component renders a list of recipes, including their names, descriptions, and images.

3. Recipe Detail Component:

Purpose: Displays detailed information about a single recipe.

On Edit: A callback function to handle recipe editing.

Description: The Recipe Detail component renders detailed information about a single recipe, including its ingredients, instructions, and cooking time.

4. Search Bar Component:

Purpose: Allows users to search for recipes

On Search: A callback function to handle search queries.

Description: The Search Bar component renders a search input field and a submit button.

Reusable Components:

1. Button Component:

Purpose: A reusable button component.

Type: The type of button (e.g., "submit", "reset").

On Click: A callback function to handle button clicks.

2. Card Component:

Purpose: A reusable card component.

Title: The card title.

Image: The card image.

Description: The Card component is a reusable card component that can be customized with different titles, descriptions, and images.

3. Input Component:

Purpose: A reusable input component.

Type: The type of input (e.g., "text", "email").

OnChange: A callback function to handle input changes.

Value: The input value.

**Statement** **Management:**

**Global State:**

**Overview:**The Cookbook Recipes App uses the Context API for global state management. The global state is stored in the AppContext component, which wraps the entire app.

Global State Properties:

The global state includes the following properties:

- Recipes: An array of recipe objects.

- Current User: The currently logged-in user object.

- SelectedRecipe: The currently selected recipe object.

**Local** **State:**

**Overview:** Components that require local state management use the useState hook to create and manage their own local state.

Handling Local State

Components handle local state by:

- Creating a local state variable using the useState hook.

- Updating the local state variable using the setState function.

- Using the local state variable to render component content.

Local State Examples:

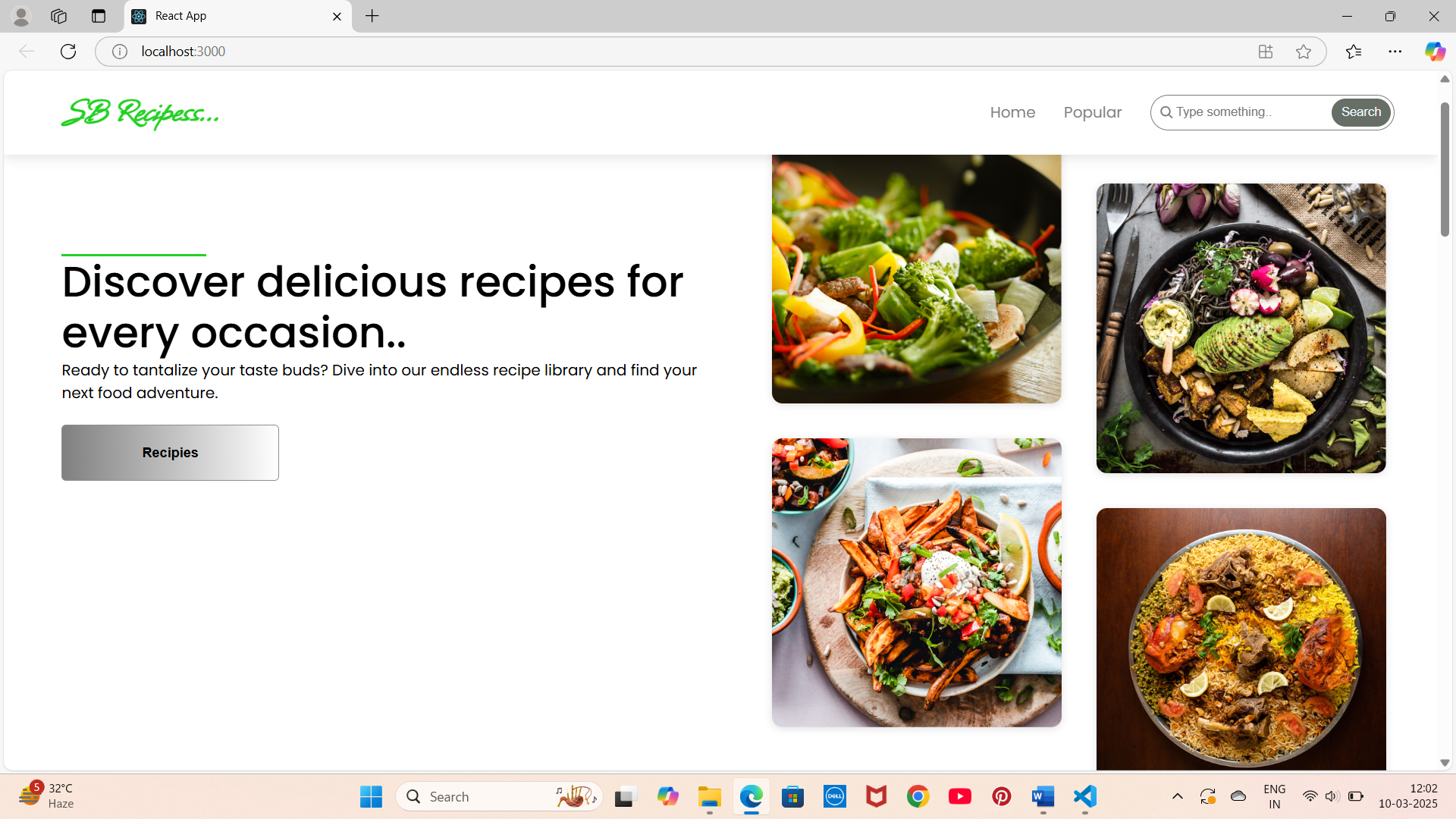
Search Bar: Uses local state to store the search query input.

Recipe Form: Uses local state to store the recipe form data.

Comment Form: Uses local state to store the comment form data.

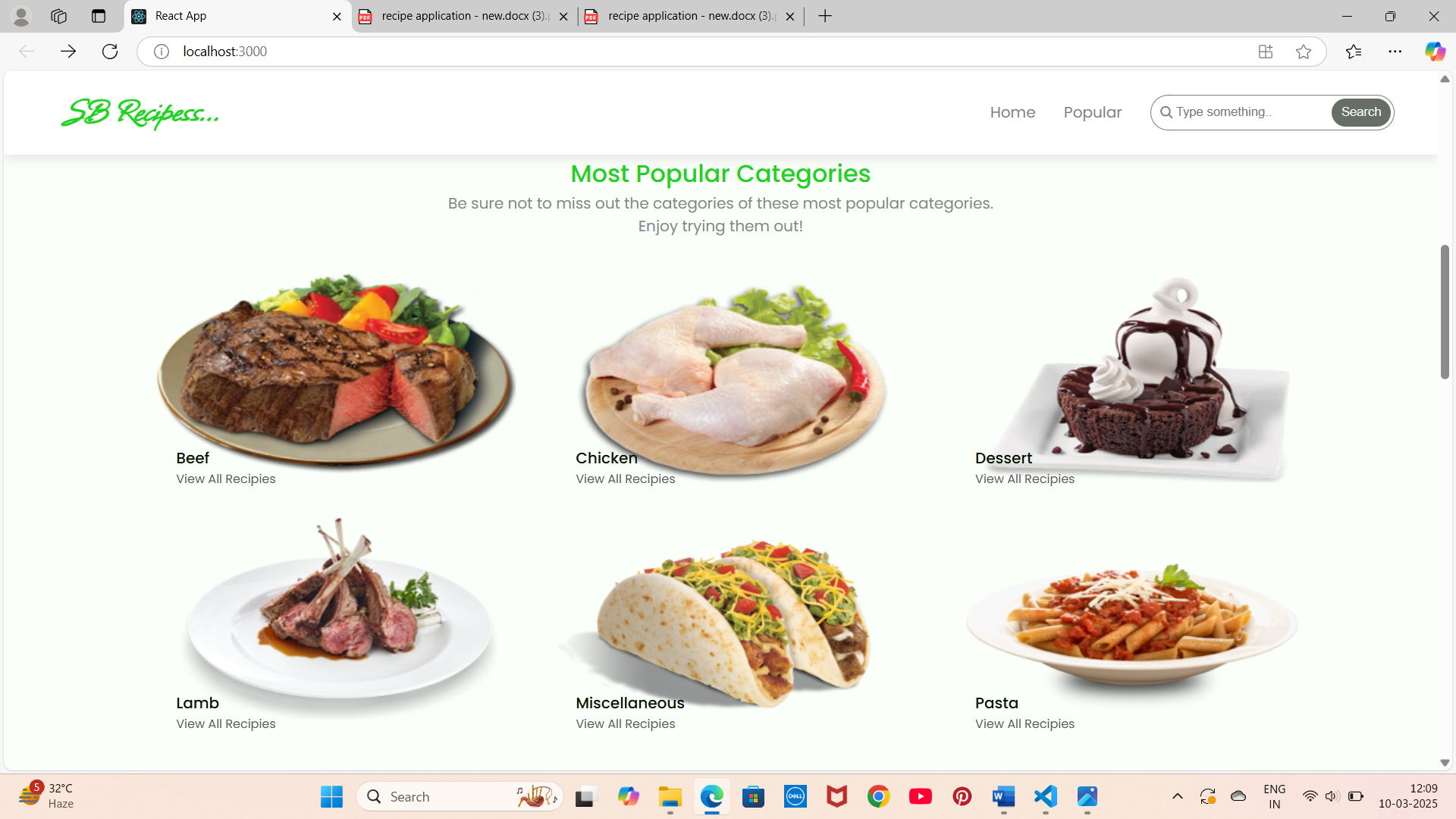
**User** **Interface:**

Hero components The hero component of the application provides a brief description about our application and a button to view more recipes.



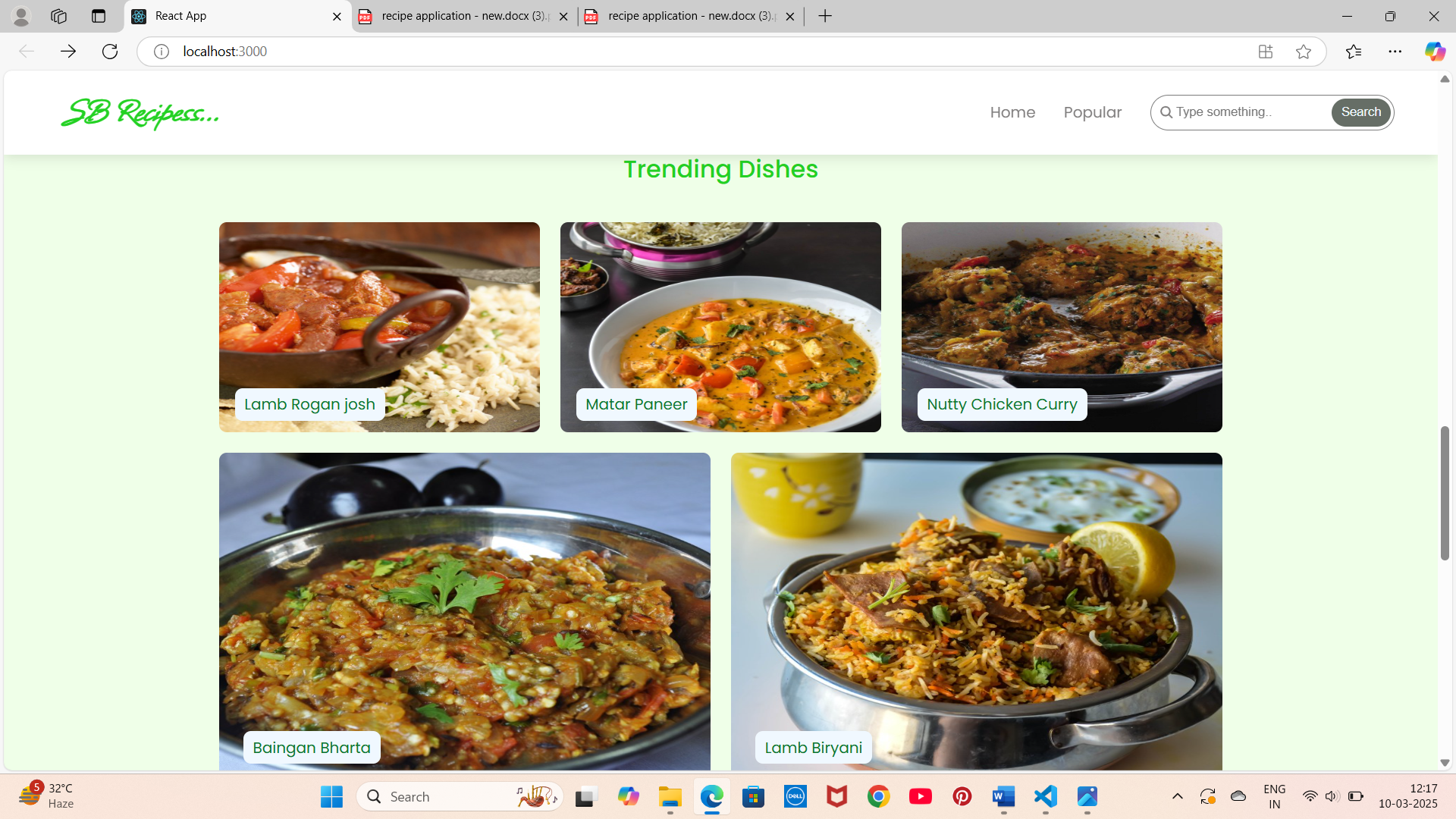
Popular categories:

This component contains all the popular categories of recipes.



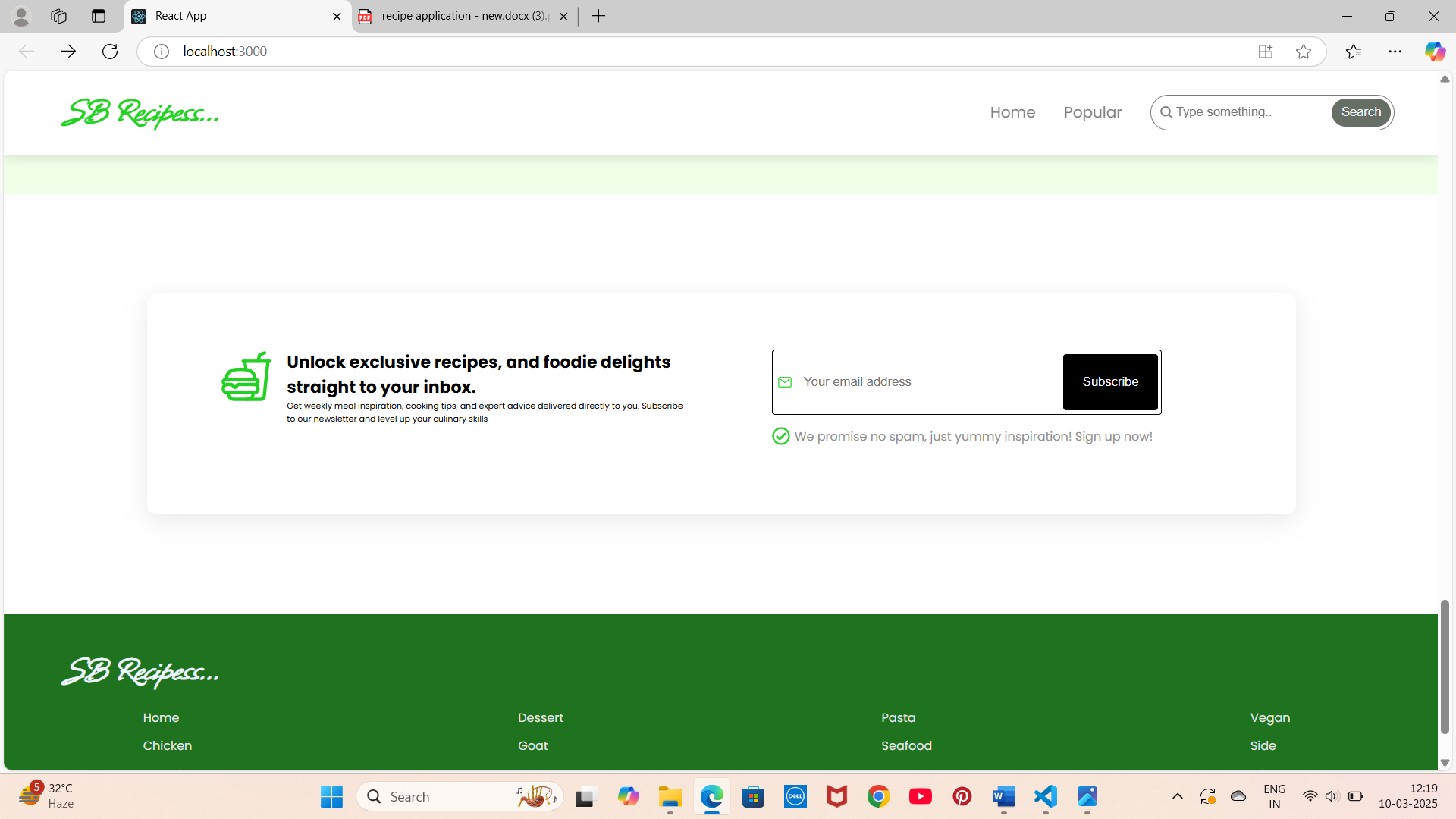
**Trending** **Dishes**:

This component contains some of the trending dishes in this application.



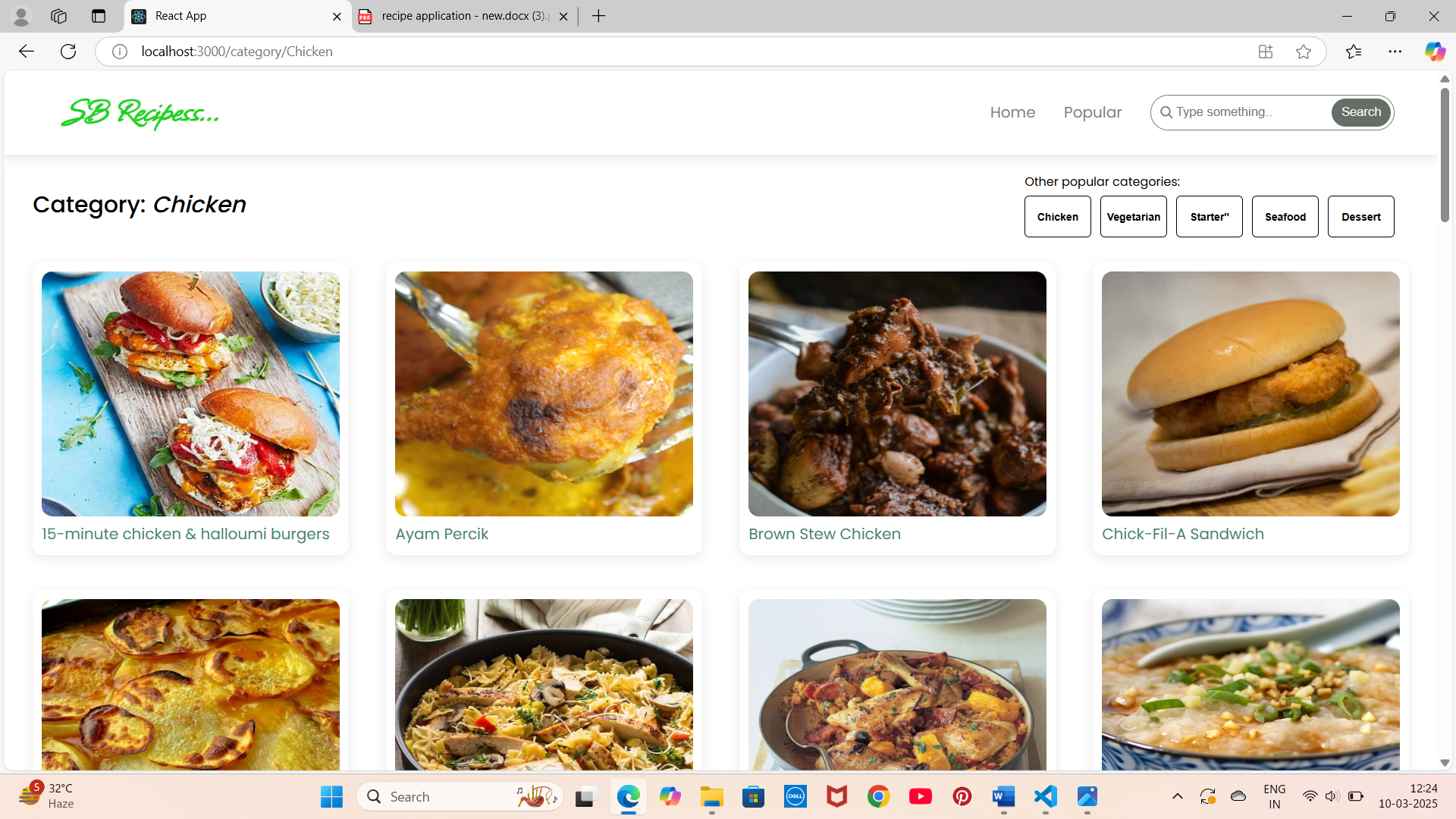
**News** **Letter**:

The news letter component provides an email input to subscribe for the recipe newsletters.



**Category** **dishes** **page**:

The category page contains the list of dishes under a certain category.



**Recipe** **page**:

The images provided below shows the recipe page, that includes images, recipe instructions, ingredients and even a tutorial video.

**Styling**:

CSS Frameworks/Libraries:

Tailwind CSS: A utility-first CSS framework for building custom user interfaces.

Styled Components: A library for styling React components using a CSS-in-JS approach.

**Pre**-**processors:** Sass: A CSS pre-processor used for writing more modular and reusable CSS code.

**Theming**: Colour palette: A custom colour scheme used throughout the app.

Typography: Custom font families and typography styles.

Spacing and sizing: Consistent spacing and sizing rules for layout and design elements.

Component library: A collection of reusable UI components, such as buttons, forms, and cards.

Custom Design System: The custom design system is implemented using a combination of Tailwind CSS and Styled Components. The design system is highly customizable, allowing for easy modifications to the app's visual design and layout.

**Testing**:

The testing approach for the Cookbook Recipes App includes:

- Unit testing: Testing individual components and functions in isolation using Jest and React Testing Library.

- Integration testing: Testing how multiple components interact with each other using Jest and React Testing Library.

- End-to-end testing: Testing the entire app from a user's perspective using Cypress.

**Testing Tools:**

- Jest: A JavaScript testing framework for unit and integration testing.

- React Testing Library: A testing library for React components that provides a more intuitive API for testing.

- Cypress: An end-to-end testing framework for testing the entire app.

**Code Coverage:**

To ensure adequate test coverage, the following tools and techniques are used:

- Jest coverage report: Generates a coverage report that shows which lines of code are covered by tests.

- Code coverage thresholds: Sets a minimum code coverage threshold to ensure that all critical code paths are covered.

- Test-driven development (TDD): Writes tests before writing code to ensure that all code is test-driven.

**Demo:**

<https://drive.google.com/file/d/1LD7Jyi4ciAsLhh_oxOKXKUmcbjiKkyKp/view?usp=sharing>

**Known Issues:**

1. Recipe Image Upload:

- Issue: Recipe images are not uploading correctly, resulting in a broken image link.

- Workaround: Use a third-party image hosting service and link to the image instead of uploading it directly.

2. Search Functionality

- Issue: The search functionality is not returning accurate results, especially for recipes with multiple ingredients.

- Workaround: Use the filter functionality instead of search to narrow down recipes by ingredient or category.

3. Mobile Responsiveness

- Issue: The app is not fully responsive on mobile devices, resulting in layout issues and difficulty navigating.

- Workaround: Use a desktop or tablet device to access the app until mobile responsiveness issues are resolved.

4. User Authentication

- Issue: User authentication is not working correctly, resulting in users being unable to log in or save recipes.

- Workaround: Use a guest account to access the app until user authentication issues are resolved.

**Developer Notes:**

- The app uses a custom API for recipe data, which may require additional setup and configuration.

- The app uses a third-party library for image uploading, which may require additional dependencies and configuration.

- The app's search functionality uses a custom algorithm, which may require additional optimization and refinement.

**Reporting Issues:**

If you encounter any issues with the Cookbook Recipes App, please report them to our development team using the following contact information

**Future Enhancements:**

**1. Social Sharing Features:**

- Description: Allow users to share recipes on social media platforms like Facebook, Twitter, and Pinterest.

- Benefits: Increase user engagement and drive traffic to the app.

**2. Meal Planning and Grocery Lists:**

- Description: Introduce a meal planning feature that allows users to plan and organize meals for the week.

- Benefits: Enhance user experience and provide a more comprehensive cooking solution.

**3. Recipe Customization and Variations:**

- Description: Allow users to customize recipes based on dietary preferences, ingredient availability, and cooking methods.

- Benefits: Increase user engagement and provide a more personalized cooking experience.

**4. Augmented Reality (AR) Cooking Companion:**

- Description: Develop an AR cooking companion that provides users with step-by-step cooking instructions and real-time guidance.

- Benefits: Enhance user experience and provide a more immersive cooking experience.

**5. Enhanced Recipe Search and Filtering:**

- Description: Improve the recipe search and filtering functionality to include features like natural language processing, ingredient-based searching, and advanced filtering options.

- Benefits: Increase user engagement and provide a more efficient recipe discovery experience.

**6. User-Generated Recipe Contributions:**

- Description: Allow users to contribute their own recipes to the app, including photos, ingredients, and cooking instructions.

- Benefits: Increase user engagement, drive community growth, and provide a more diverse recipe collection.

**7. Enhanced Styling and Design:**

- Description: Improve the app's styling and design to provide a more visually appealing and user-friendly experience.

- Benefits: Increase user engagement, drive retention, and provide a more enjoyable cooking experience.

**8. Voice Assistant Integration:**

- Description: Integrate the app with popular voice assistants like Alexa, Google Assistant, and Siri to provide users with a more convenient cooking experience.

- Benefits: Increase user engagement, drive retention, and provide a more hands-free cooking experience.

**9. Multi-Language Support:**

- Description: Add support for multiple languages to cater to a broader user base and increase the app's global reach.

- Benefits: Increase user engagement, drive retention, and provide a more inclusive cooking experience.