Cookbook:

Your virtual kitchen Assistant



**1. Introduction**

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

***Team Members*:**

* Abinaya C – Team Leader (Project Coordination & Backend Oversight)
* Afreen D – Frontend Developer (React, UI/UX Design
* Afrina Shirin B – Backend Developer (API Integration, Database)
* Aysha M – QA Engineer (Testing, Bug Tracking)
* Aysha R – Project Manager (Documentation, Deployment)

---

**2. Project Overview**

***Purpose*:**

* CookBook is a virtual kitchen assistant designed to help users explore, organize, and manage recipes effortlessly. It simplifies meal planning, cooking steps, and ingredient tracking through an interactive and user-friendly interface.

***Goals***:

* Provide a collection of recipes with step-by-step instructions.
* Enable users to create personalized recipe collections.
* Assist users with ingredient lists and preparation time.
* Offer an intuitive and visually engaging cooking experience.

***Features:***

* Recipe search and filter (by cuisine, diet, or preparation time)
* Recipe detail page with ingredients, instructions, and nutrition info
* Personalized recipe collection (Favorites & Saved Recipes)
* Ingredient checklist with progress tracking
* Responsive design for mobile and desktop

---

**3. Architecture**

***Component Structure***:

* *App*: Root component handling routing
* *Header*: Navigation bar (Home, Recipes, Favorites)
* *HomePage*: Landing page with featured recipes
* *RecipeList*: Displays recipe cards in a grid layout
* *RecipeCard*: Reusable card for each recipe (title, image, prep time)
* *RecipeDetail*: Full recipe details (steps, ingredients, instructions)
* *FavoritesPage*: Displays user’s saved recipes
* *SearchBar*: Input component for recipe search/filter
* *Footer*: App footer with copyright

***State Management***:

* *Global*: Context API for user favorites and search filters
* *Local*: useState/useReducer for component-level states like inputs and toggles

***Routing:***

* react-router-dom used

***Routes***:

* / → HomePage
* /recipes → RecipeList
* /recipe/:id → RecipeDetail
* /favorites → FavoritesPage

---

**4. Setup Instructions**

***Prerequisites*:**

Node.js v16 or later

npm or yarn

***Installation***:

*1. Clone the repository*:

git clone https://github.com/your-username/cookbook-virtual-assistant.git

*2. Navigate into the project directory*:

cd cookbook-virtual-assistant

*3. Install dependencies*:

npm install

4. Configure environment variables (API keys, backend URLs) in .env

---

**5. Folder Structure**

cookbook-virtual-assistant/

│── public/

│── src/

│ ├── assets/ # Images, icons, static files

│ ├── components/ # Reusable components (Header, Footer, RecipeCard, SearchBar)

│ ├── pages/ # Page-level components (HomePage, RecipeList, RecipeDetail, FavoritesPage)

│ ├── context/ # Context API (Global state management)

│ ├── hooks/ # Custom hooks (e.g., useFetchRecipes, useFavorites)

│ ├── utils/ # Utility functions (API calls, formatters)

│ ├── styles/ # Global CSS, theme files

│ └── App.js # Root application file

└── package.json

***Utilities***:

* *api.js*: API request handler (fetch recipes, recipe details)
* *helpers.js*: Utility functions for formatting data (e.g., cooking time)
* *useFavorites.js*: Custom hook for managing user favorites

---

**6. Running the Application**

***Frontend*:**

* cd cookbook-virtual-assistant

npm start

---

**7. Component Documentation**

***Key Components***:

* *RecipeCard*: Props → title, image, prepTime, onFavorite
* *RecipeDetail*: Props → recipeId (fetches full details)
* *SearchBar*: Props → onSearch, onFilterChange

***Reusable Components***:

* *Button*: Styled button used across app
* *Modal*: Reusable modal for popups
* *Loader*: Loading spinner for API calls

---

**8. State Management**

***Global State (Context API):***

* *Favorites*: Manages user’s favorite recipes across the app
* *Search Query:* Stores search/filter criteria for global access

***Local State***:

* Form inputs (search bar text)
* Toggle states (like expanding ingredient list)

---

**9. User Interface**

* *HomePage*: Featured recipes carousel
* *RecipeList*: Grid of recipe cards with search filters
* *RecipeDetail*: Full-screen recipe with step-by-step cooking guide
* *FavoritesPage*: User’s saved recipes

(Screenshots or GIFs to be inserted after deployment)

---

**10. Styling**

***CSS Frameworks/Libraries:***

* Tailwind CSS for global styling
* Styled-Components for scoped component styling

***Theming***:

* Light/Dark mode toggle implemented
* Custom warm-toned color palette for food-friendly design

---

**11. Testing**

***Testing Strategy***:

* *Unit Testing*: Jest + React Testing Library
* *Integration Testing*: Ensure components interact correctly
* *End-to-End Testing*: Cypress for user flows (searching, saving recipes)

***Code Coverage****:*

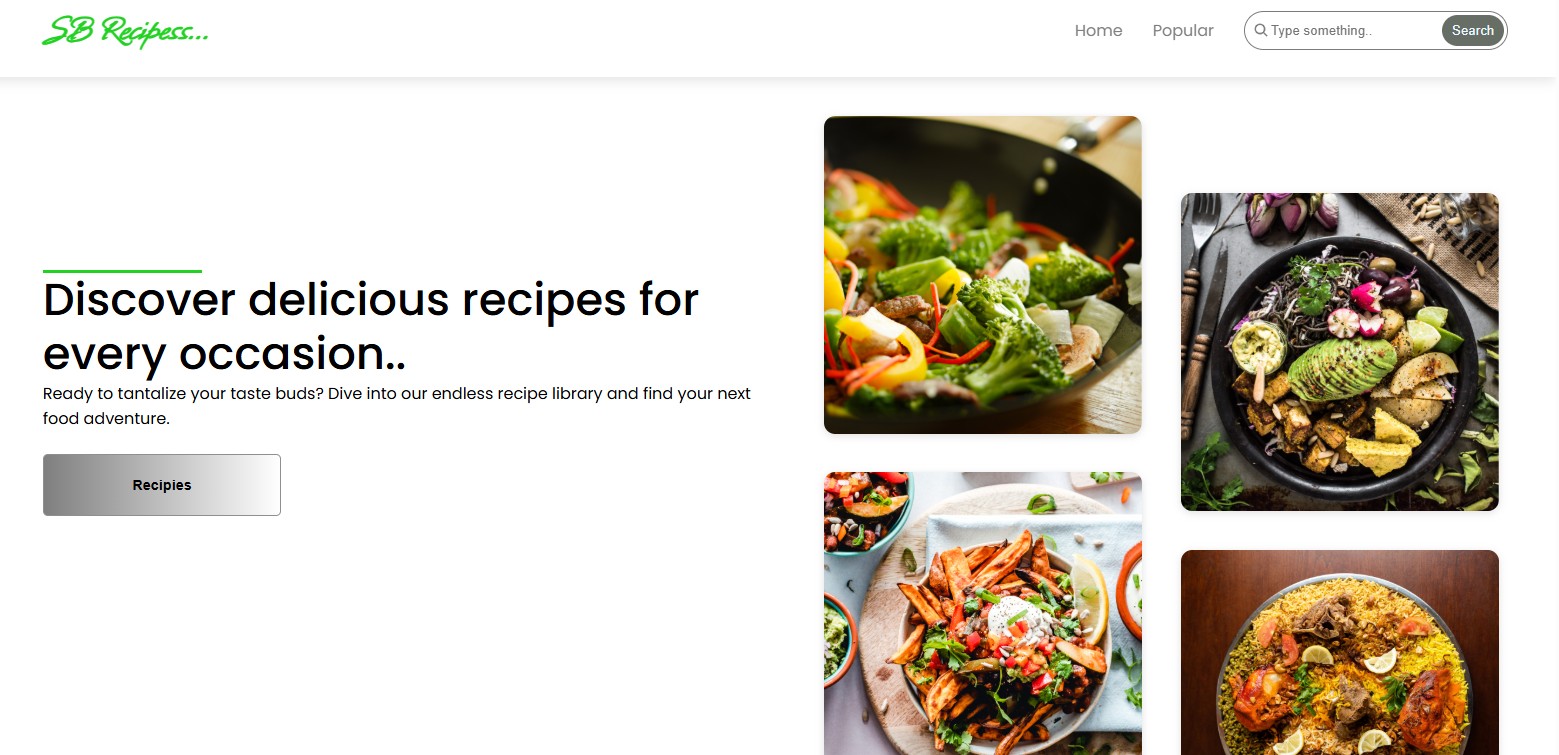
* Tracked with jest --coverage

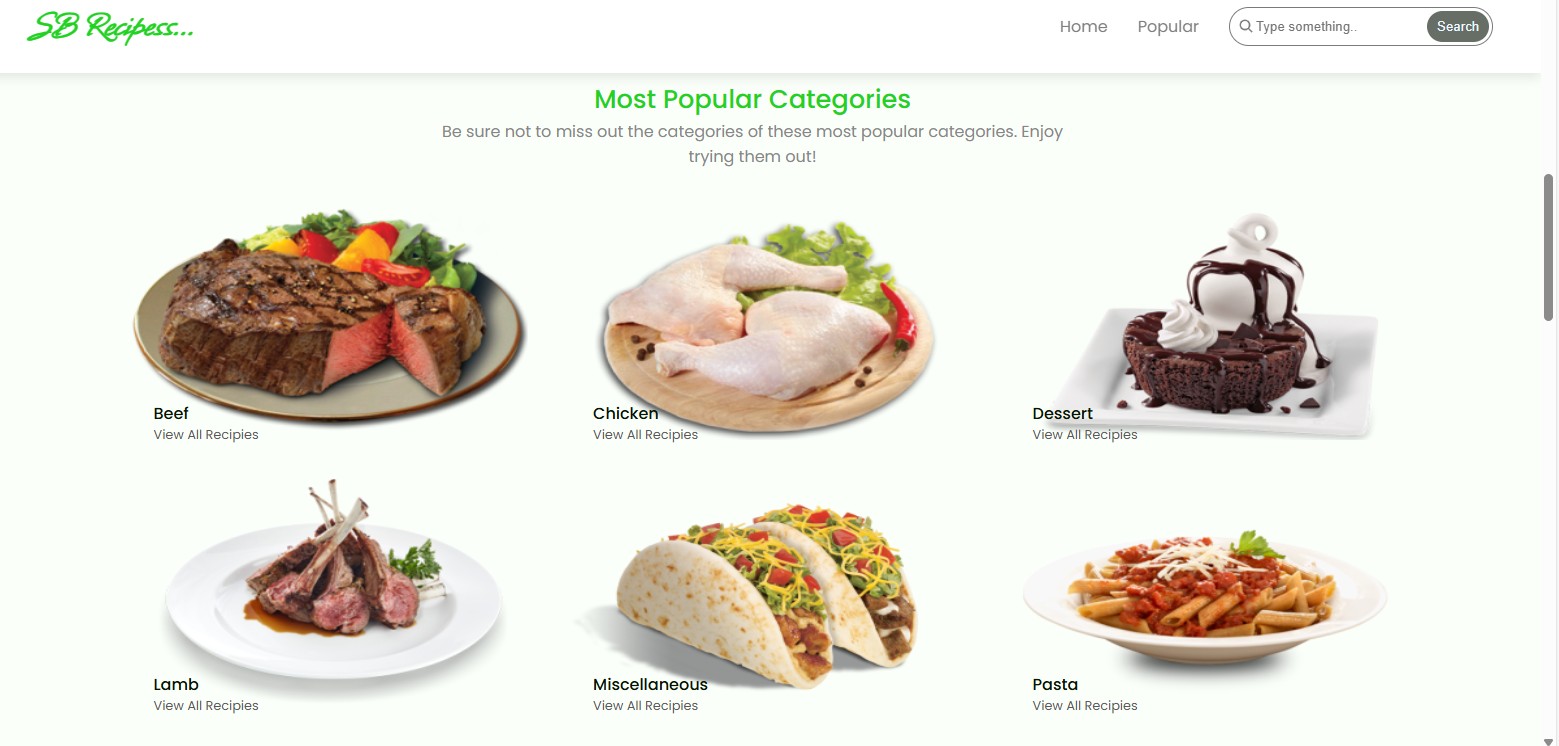
---

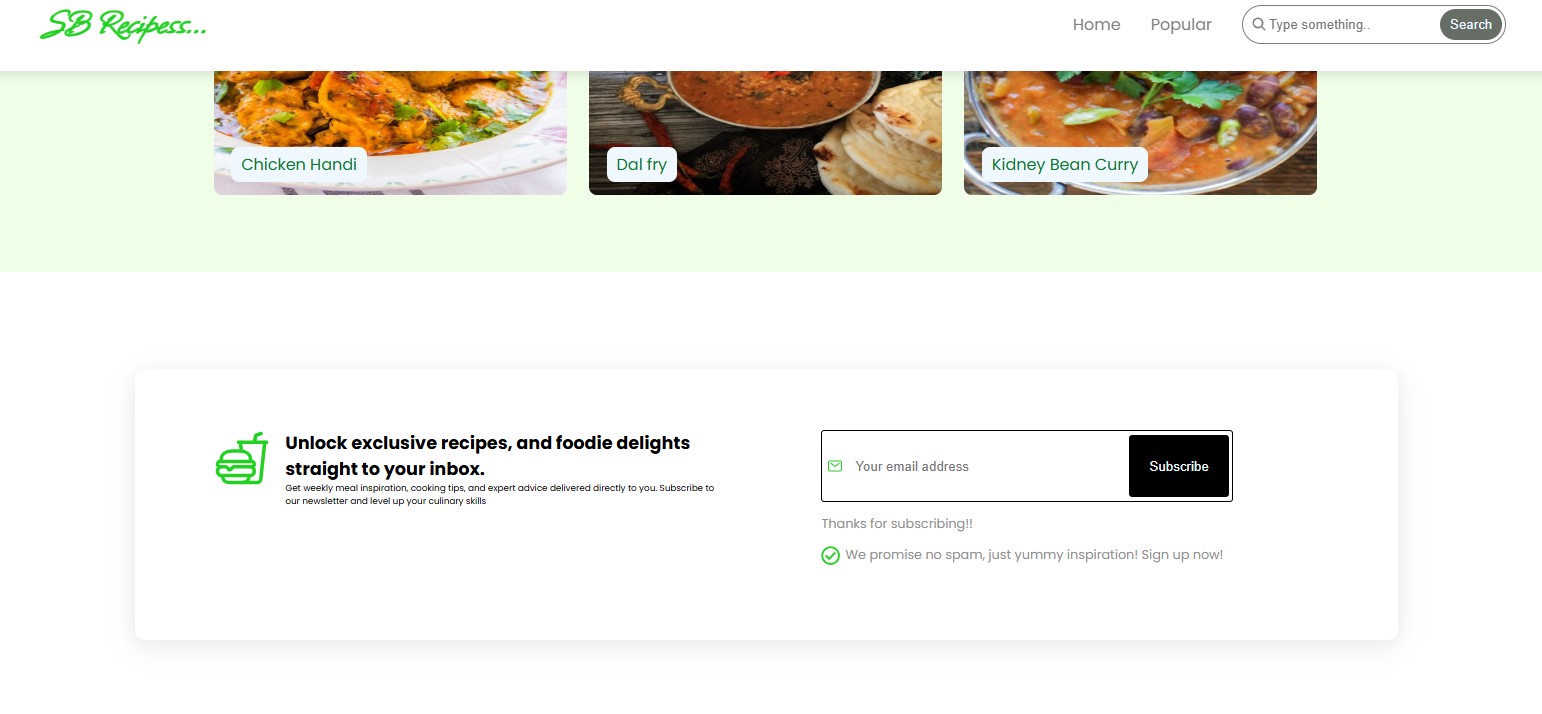
**12. Screenshots or Demo**

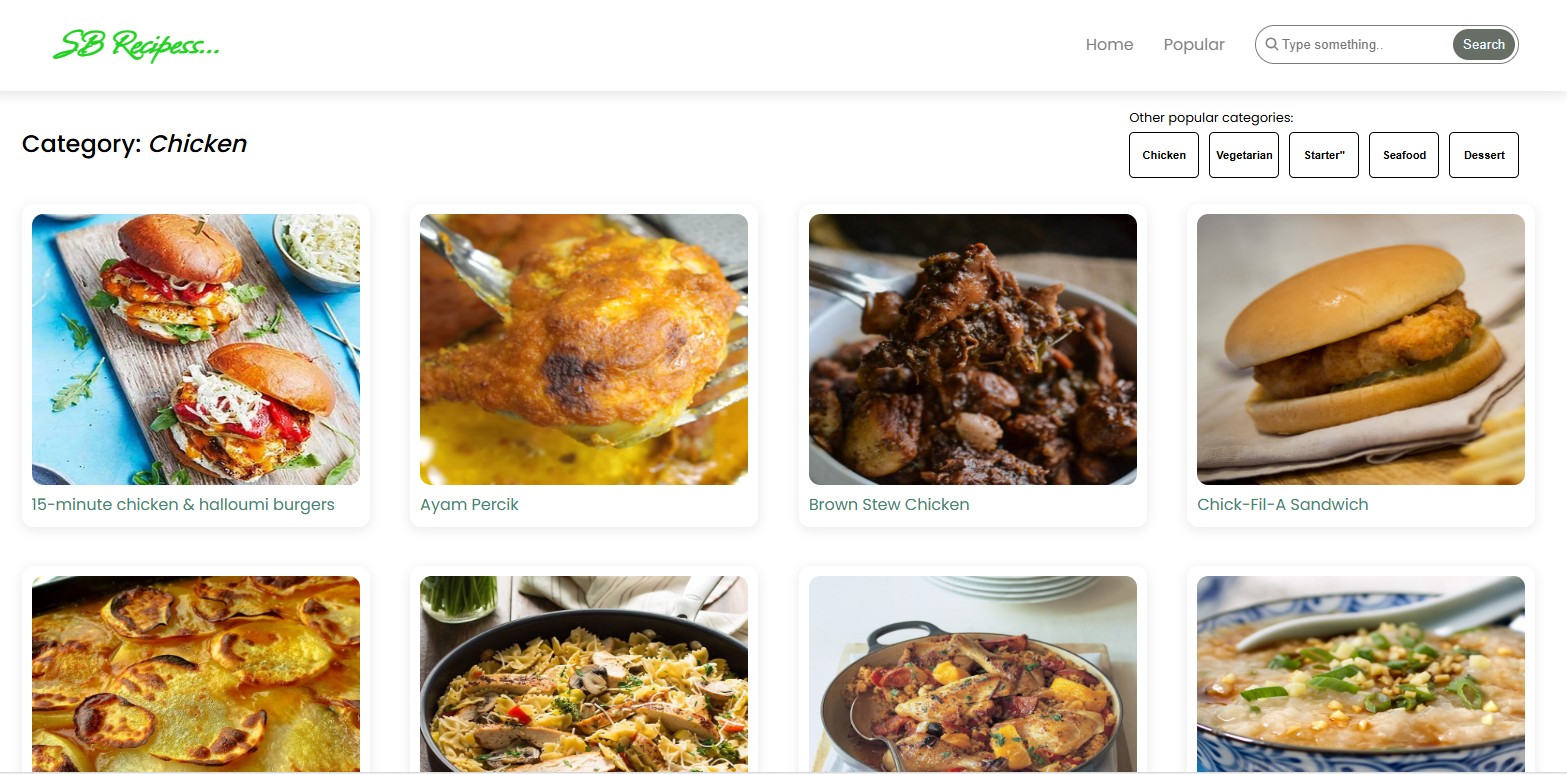
***Demo Link****:* CookBook Demo

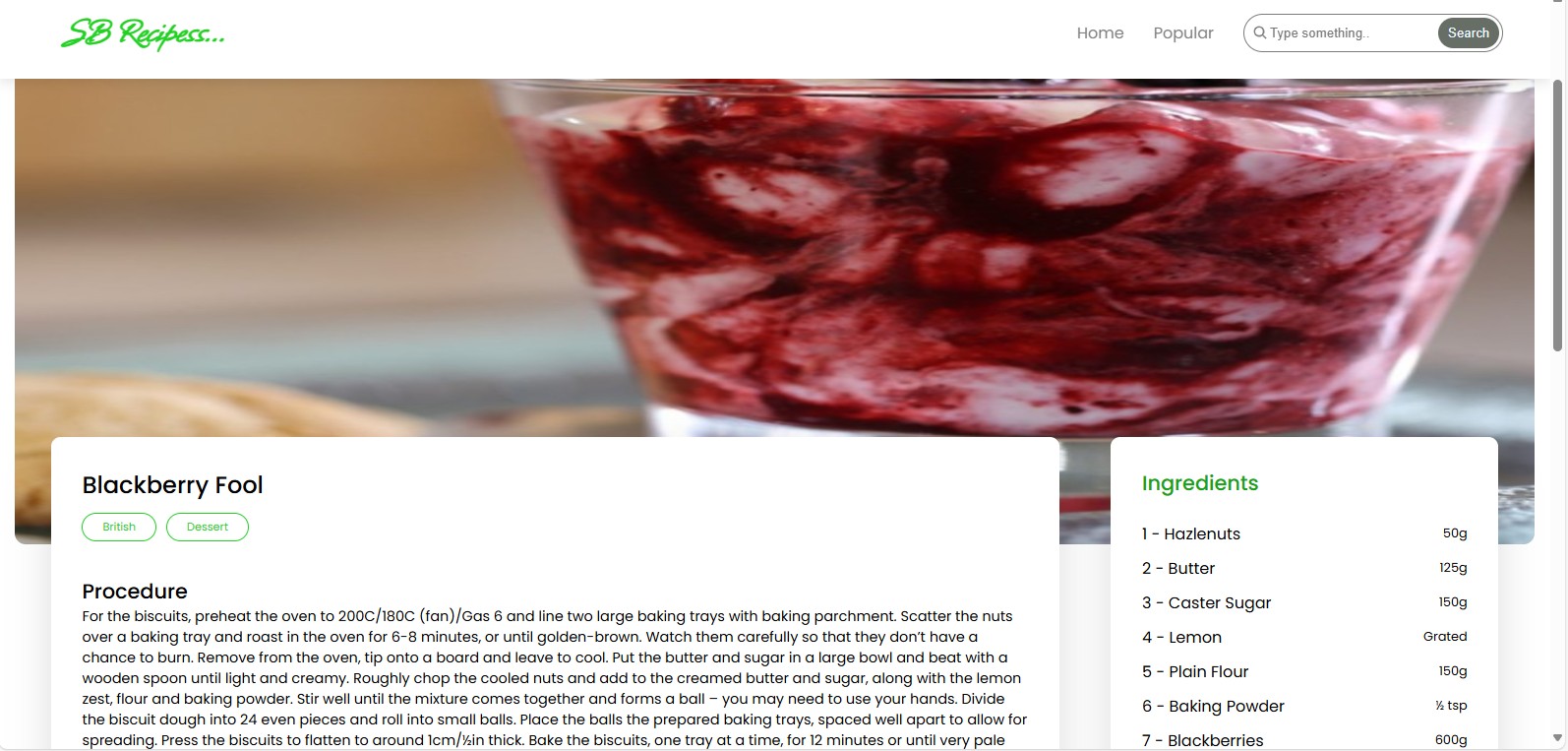
***Screenshots****:*

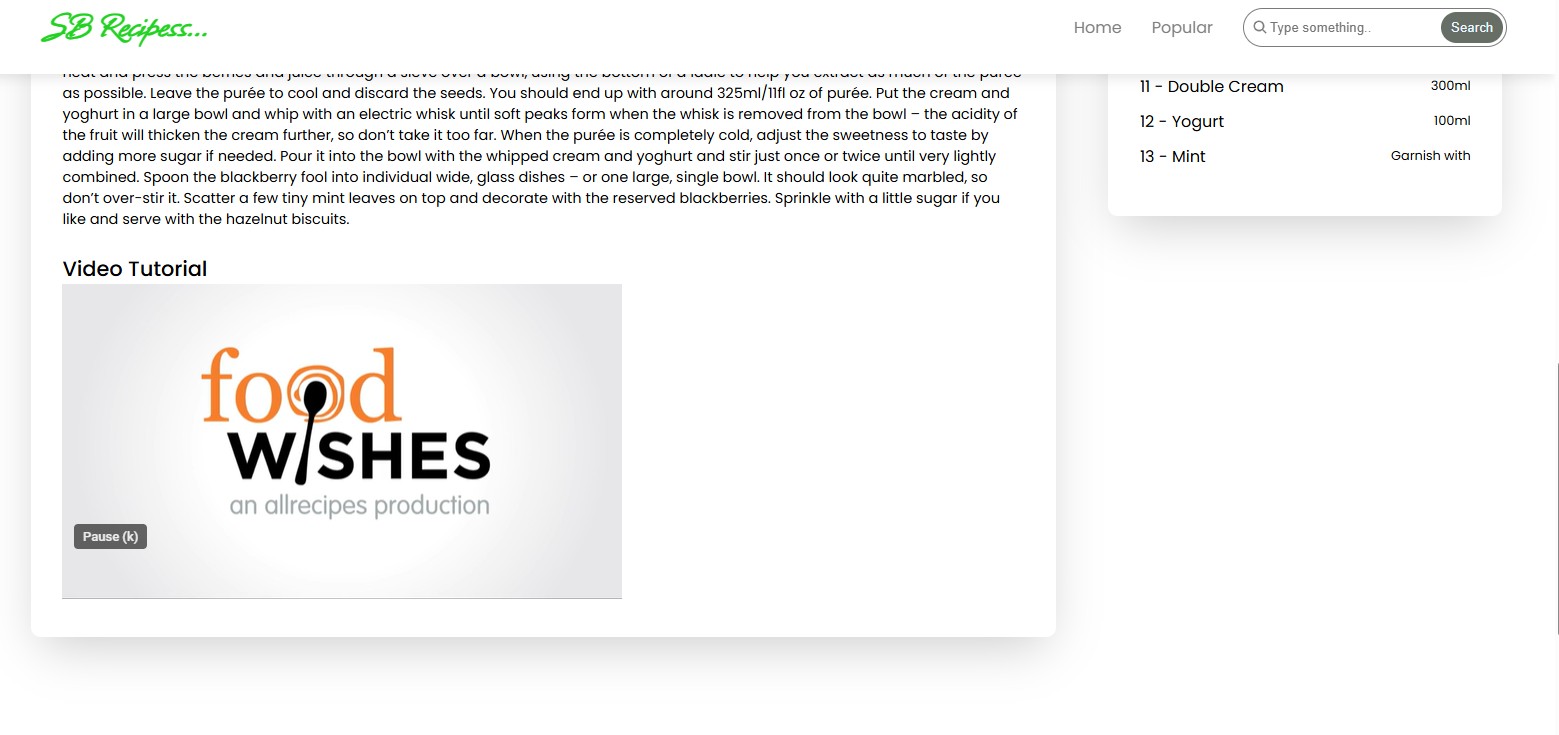


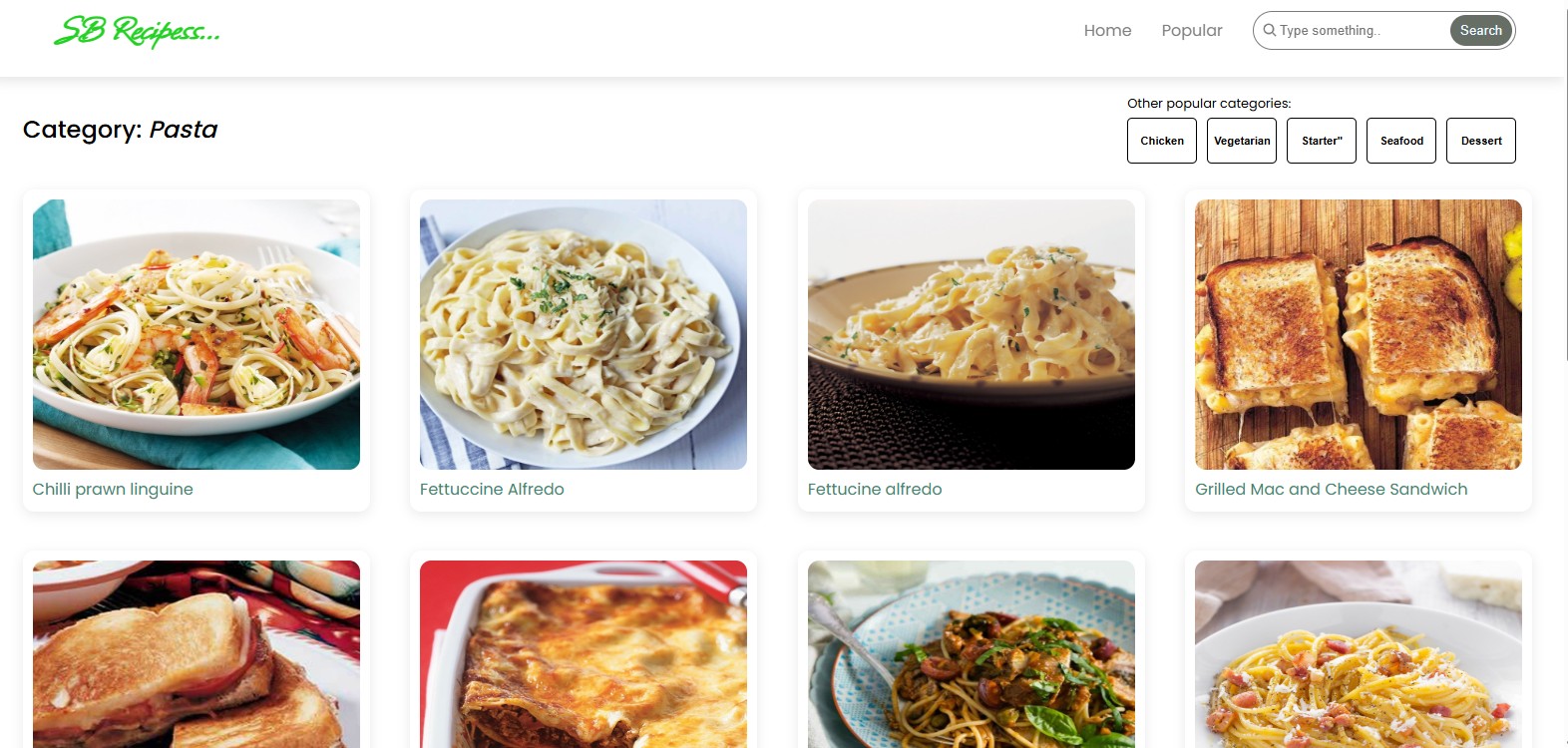


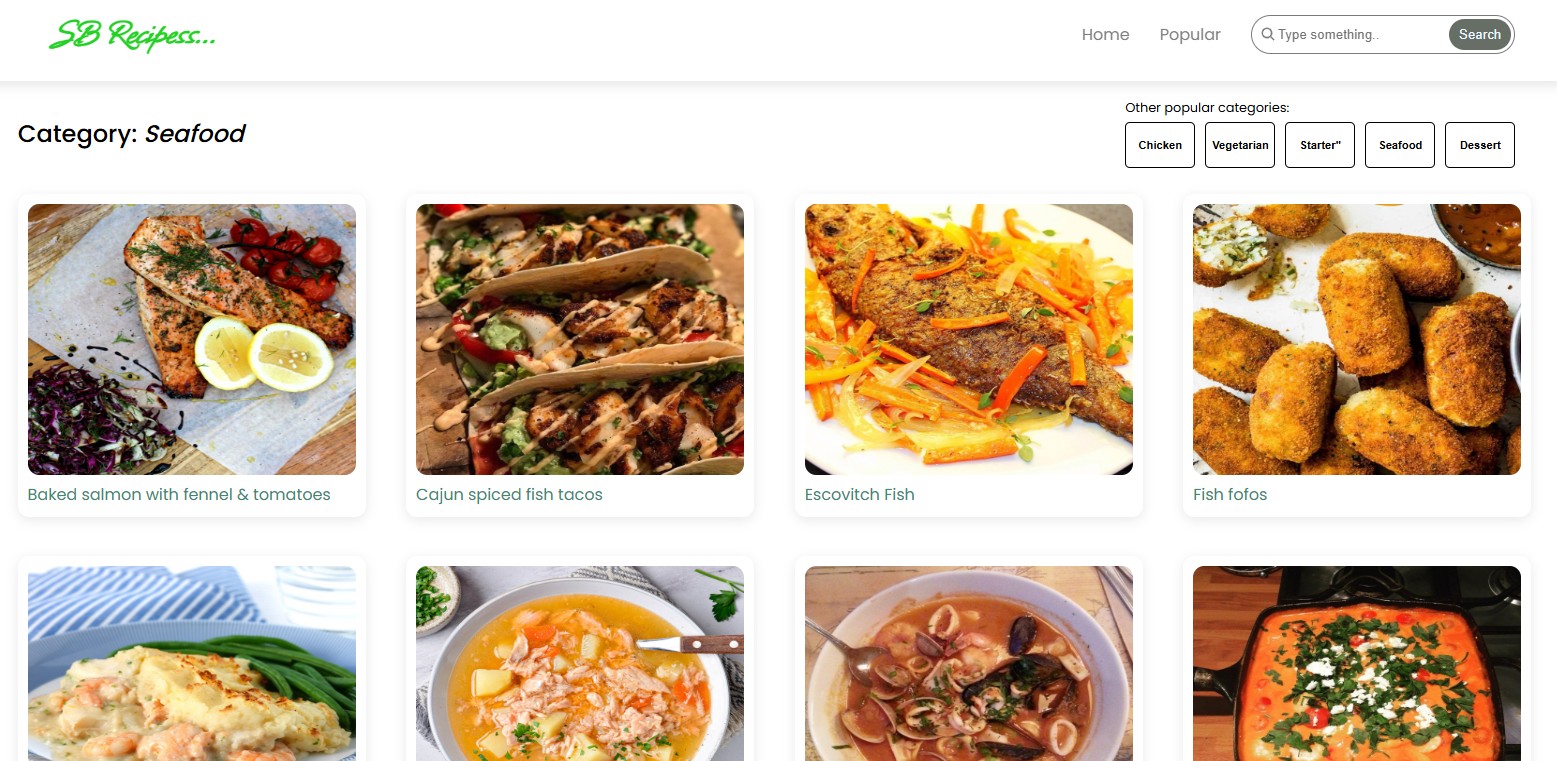












**13. Known Issues**

* API rate limits may restrict recipe searches
* Dark mode styling incomplete for some pages
* Favorites reset on page reload without backend persistence

---

**14. Future Enhancements**

* User authentication for saving recipes across devices
* Shopping list generation based on selected recipes
* Voice assistant integration for step-by-step cooking guidance
* Offline support with service workers
* Advanced filters (calorie count, allergen-free options)

—