**Project documentation**

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

**Team Members:**

**R . Magathi** – Project lead &Frontend Developer

**H .Mubina Hasin** – UI/UX Designer

**M .Naseha Begum –** Backend Developer(API integration)

**A.Nasima –** Project Manager

**A. Ojeetha Begum –** Documentation & Research

**Project Overview**

**Purpose**:

Cookbook is a virtual kitchen assistant designed to simplify cooking by providing step-by-step recipes, ingredient management, and personalized suggestions. It helps users discover recipes, track ingredients, and improve their cooking experience.

**Features:**

* Recipe search and filter (by cuisine, ingredients, or cooking time).
* Interactive recipe steps with timers.
* Favorite and save recipes for quick access.
* Ingredient list management (pantry tracking).
* Responsive UI for mobile and desktop.

**Architecture**

**Component Structure:**

* App Component – Root of the application, manages routing.
* Header & Footer – Navigation and branding.
* RecipeList – Displays recipes with search and filter.
* RecipeDetails – Shows ingredients, instructions, and media.
* PantryManager – Handles ingredient tracking.
* Favorites – Displays saved recipes.
* Reusable UI Components – Buttons, Modals, Cards, Input fields.

**State Management:**

* Implemented using React Context API for global states like user favorites and pantry items.
* Local component states for UI interactions (form inputs, modals).

**Routing**:

Implemented using React Router v6.

Routes:

* / → Home (Recipe list)
* /recipe/:id → Recipe detail page
* /favorites → Saved recipes
* /pantry → Ingredient manager

**Setup Instructions**

**Prerequisites:**

**-**Node.js (>= 16.x)

**-**Npm or yarn

**-**Git

**Installation:**

1.*Clone the repository :*

Git clone <https://github.com/your-repo/cookbook.git>

2.*Navigate into Project directory:*

Cd cookbook/client

3.*Install dependencies*

Npm install

4.Create .env file for API keys (if recipe API is used).

**Folder Structure**

Client/

├── public/

├── src/

│ ├── assets/ # Images, icons, styles

│ ├── components/ # Reusable UI components

│ ├── pages/ # RecipeList, RecipeDetail, Pantry, Favorites

│ ├── context/ # Global state providers

│ ├── hooks/ # Custom hooks (useFetch, usePantry)

│ ├── utils/ # Helper functions (formatting, filtering)

│ ├── App.js

│ ├── index.js

├── package.json

**Utilities:**

* useFetch – Custom hook for API requests.
* filterRecipes – Helper for searching/filtering recipes.

**Running the Application**

**Frontend Server:**

* Cd client
* Npm start

**Component Documentation**

**Key Components:**

* RecipeCard – Displays recipe thumbnail, name, and quick info. Props: title, image, time.
* RecipeDetail – Full recipe details with step-by-step instructions. Props: id.
* PantryManager – Adds/removes ingredients from pantry list.
* FavoritesList – Renders all saved recipes.

**Reusable Components:**

* Button- Custom styled button Component.
* Modal- For displaying popups(e.g.,recipe instructions).
* Loader- Loading spinner for API calls.

**State Management**

**Global State:**

* Context API for favorites and pantry items.
* Example: FavoritesContext, PantryContext.

**Local State:**

* Input fields, modal visibility, active step in recipe instructions.

**User Interface**

* Screenshots (example placeholders):
* Home Page (recipe browsing)
* Recipe Detail with instructions
* Pantry Manager (ingredient tracking)
* Favorites Page

**Styling**

* CSS Frameworks/Libraries:
* Tailwind CSS for styling and responsive design.
* React Icons for visuals.

**Theming**:

* Light/Dark mode toggle implemented with Tailwind classes.
* Custom color palette for branding.

**Testing**

**Testing Strategy:**

* Unit tests with Jest and React Testing Library.
* Integration tests for context providers.
* End-to-end tests with Cypress (recipe flow simulation).

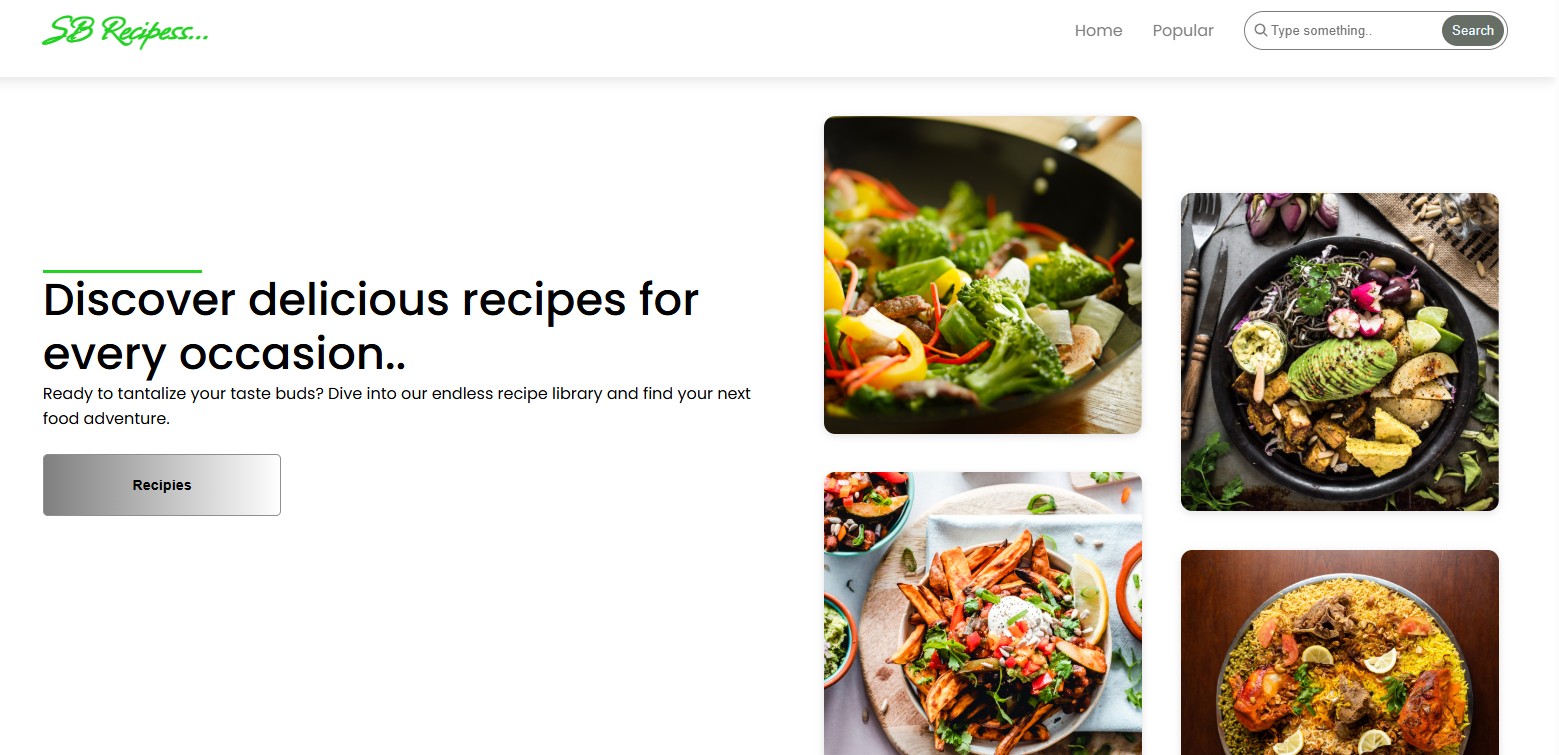
**Code Coverage:**

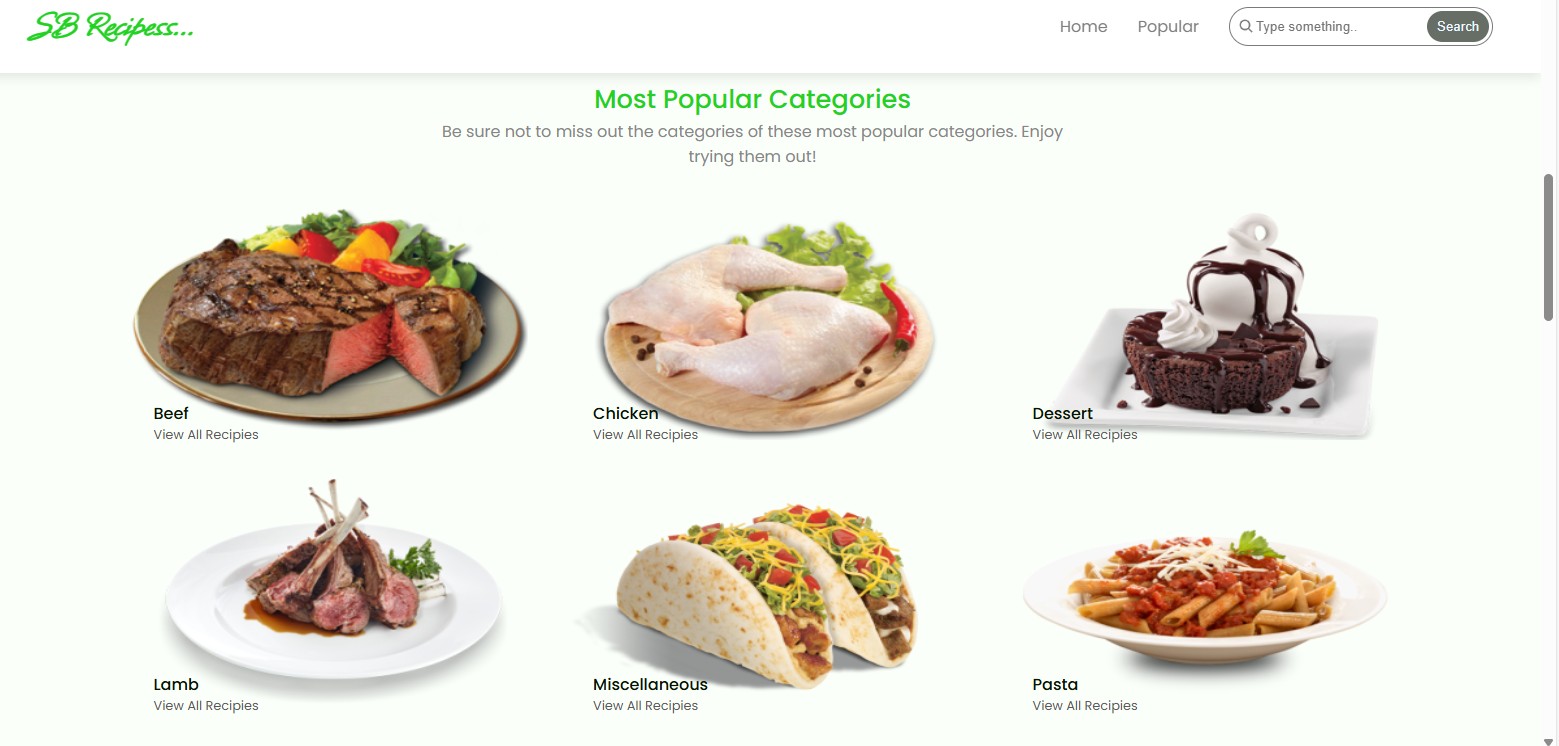
* Jest coverage reports (npm run test -- --coverage).

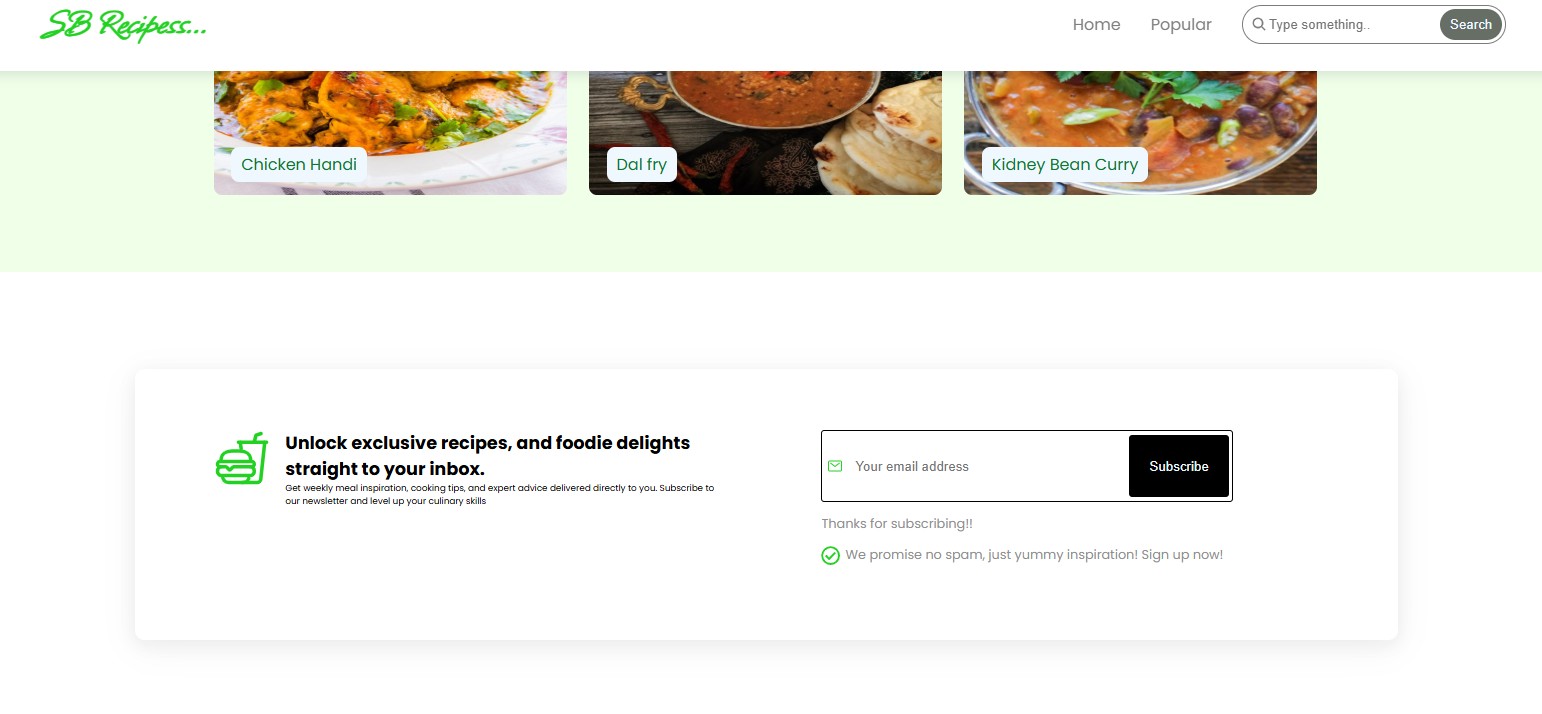
**Screenshots or Demo**

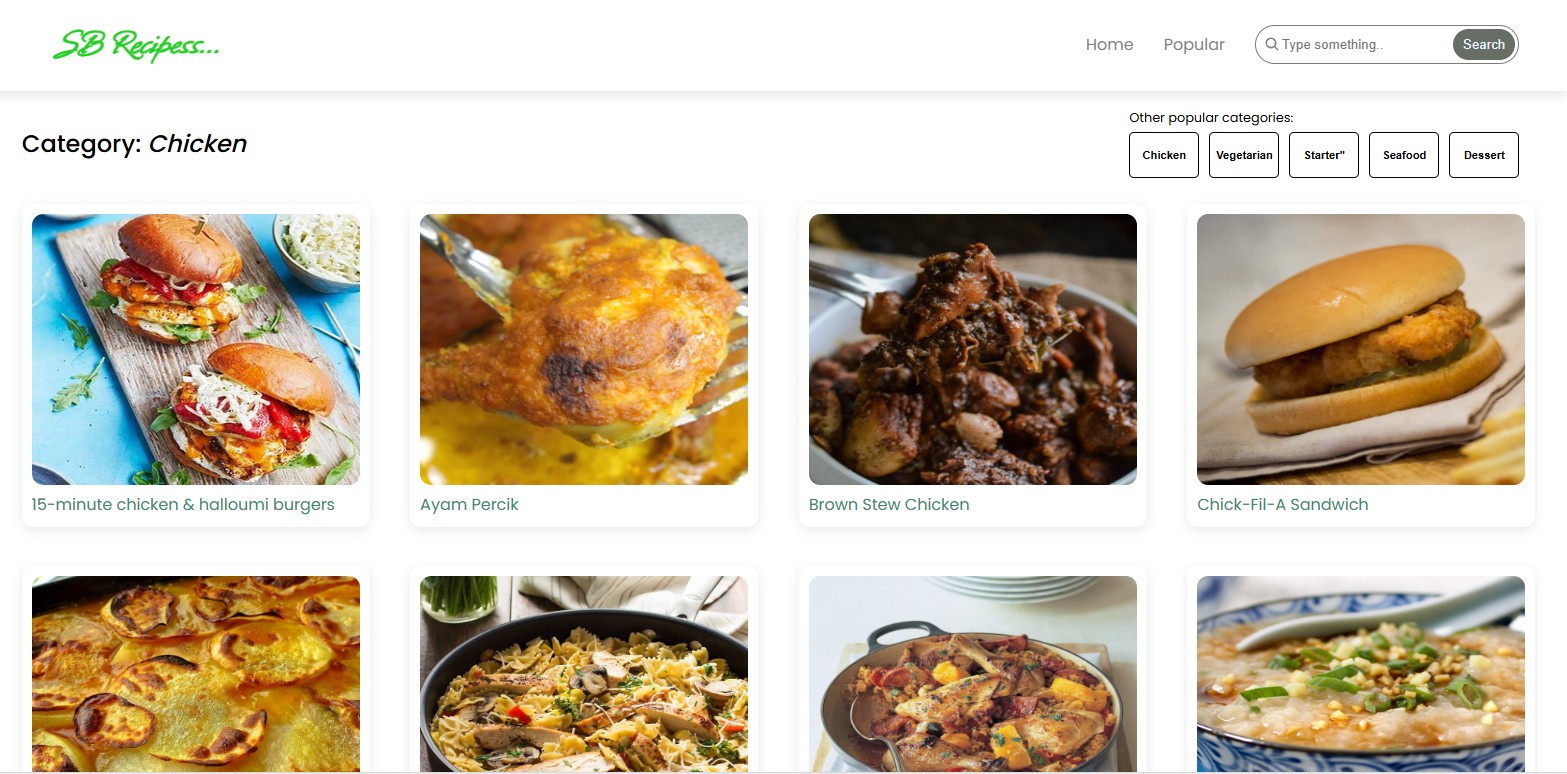
**Demo link:** [Insert deployment link, e.g., Netlify or Vercel]

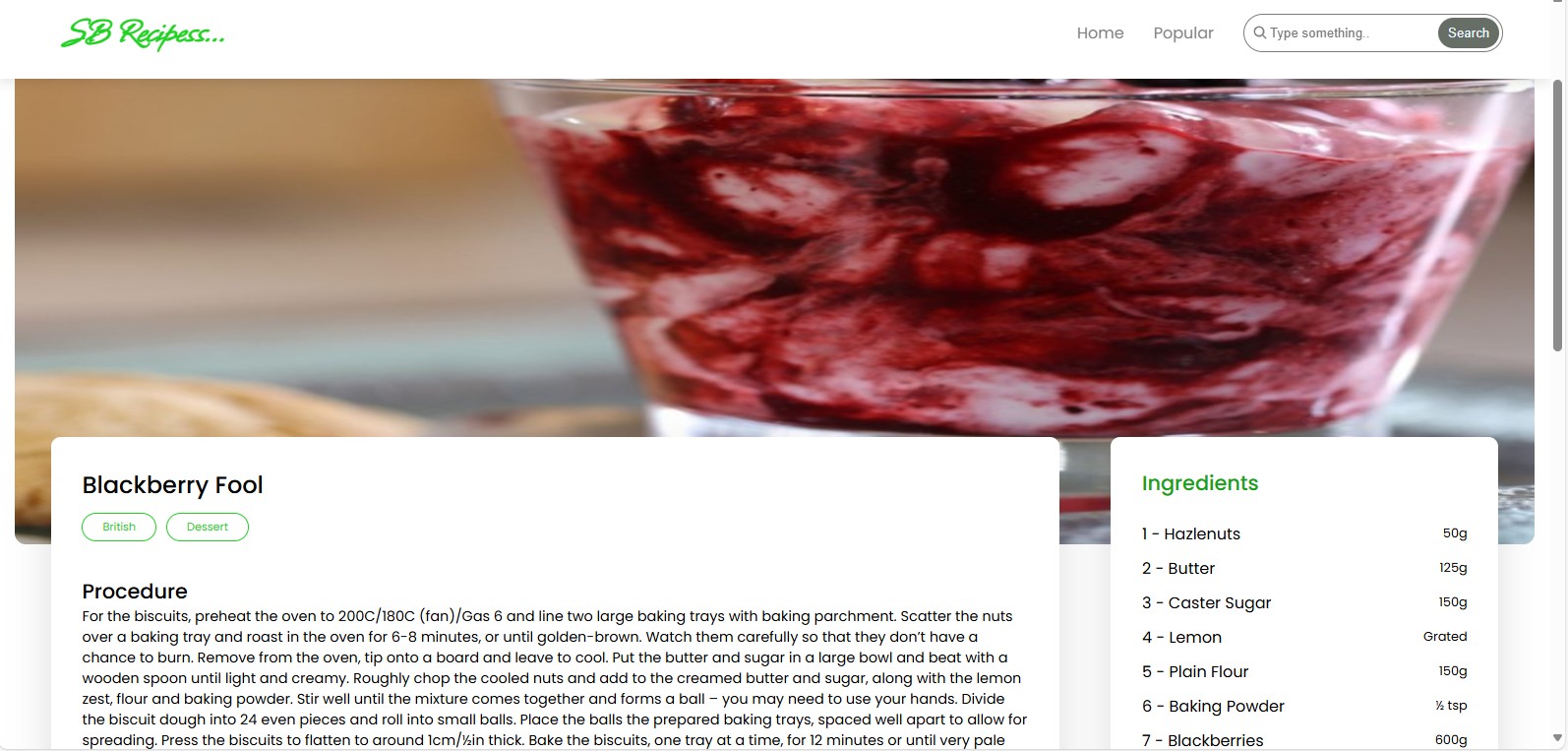
**Screenshots** :

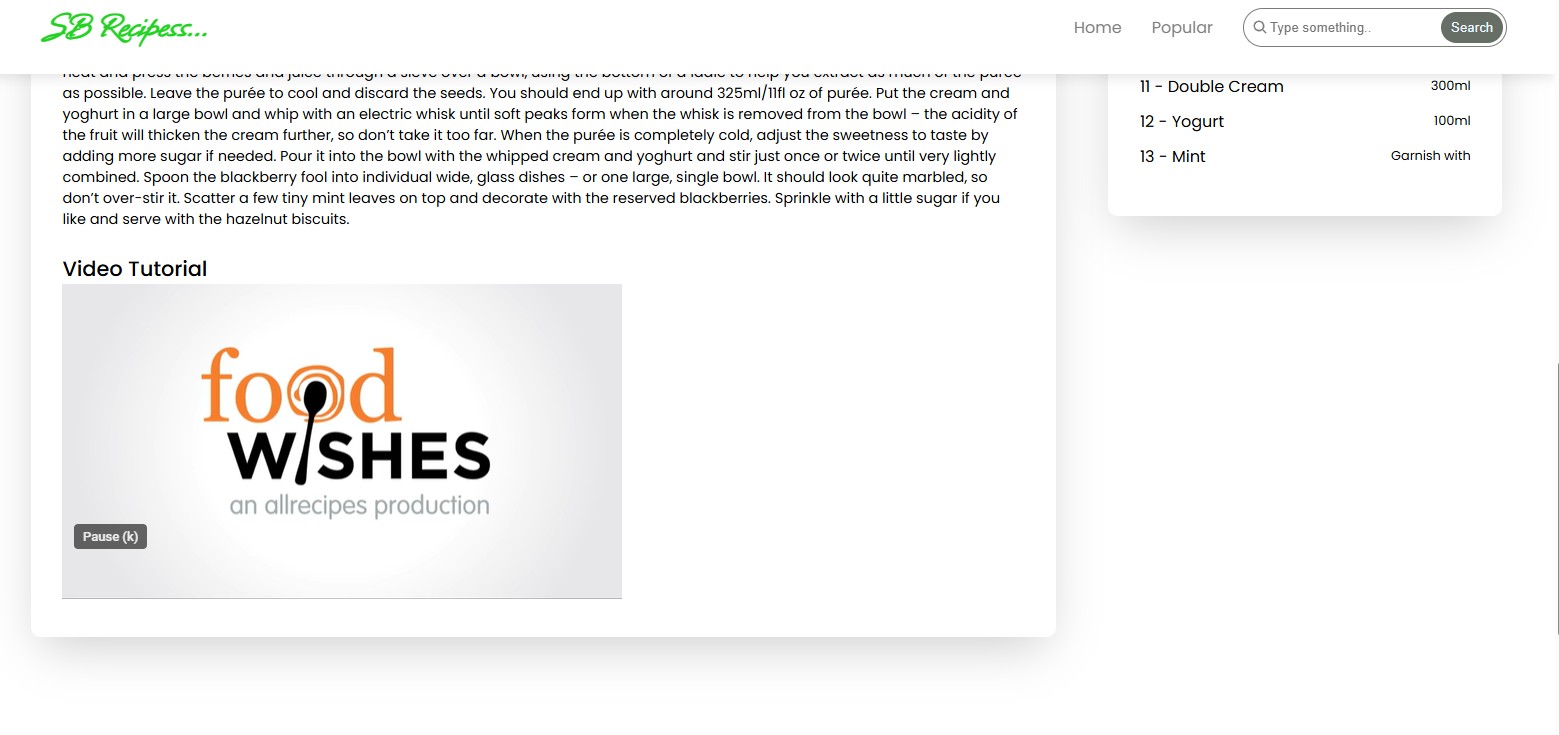


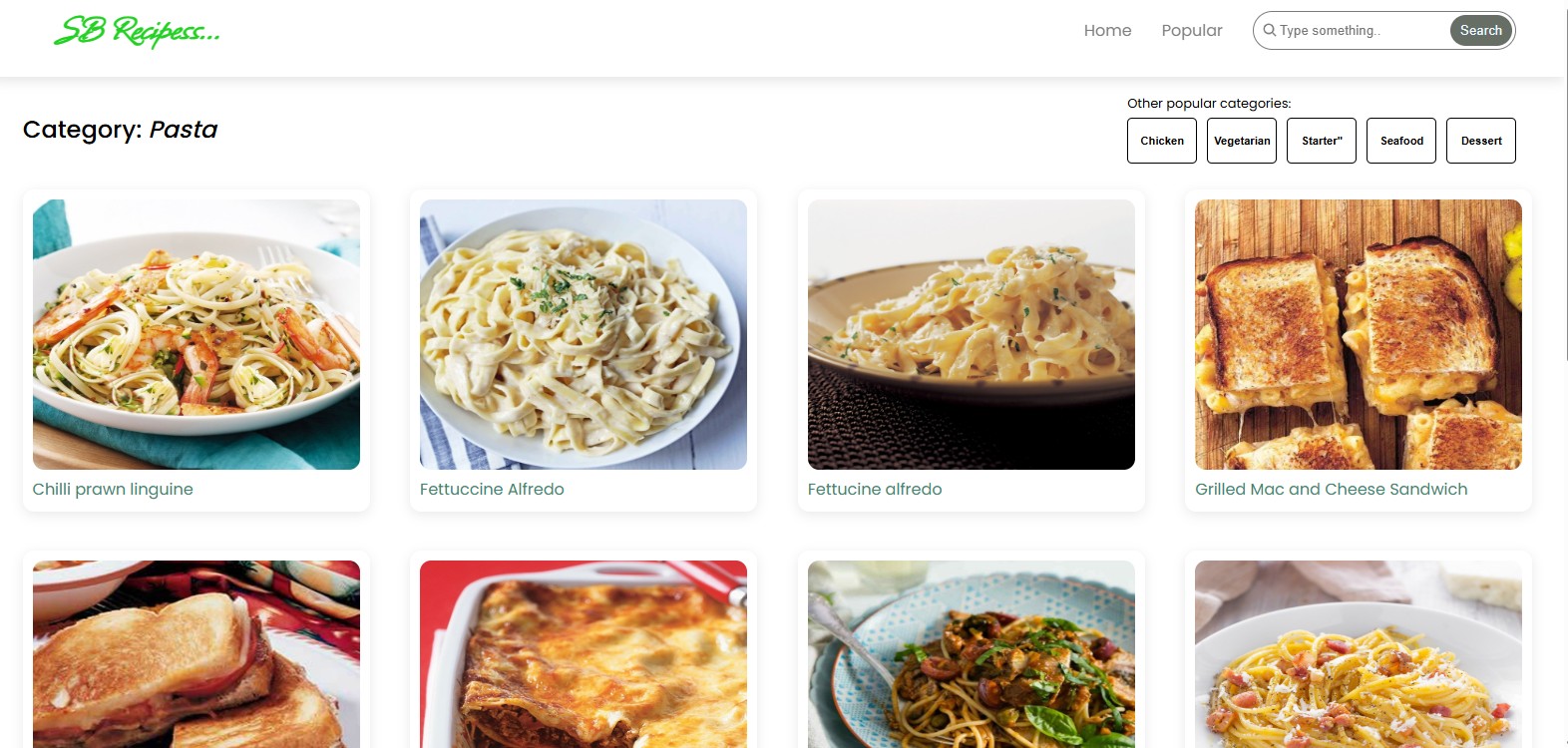


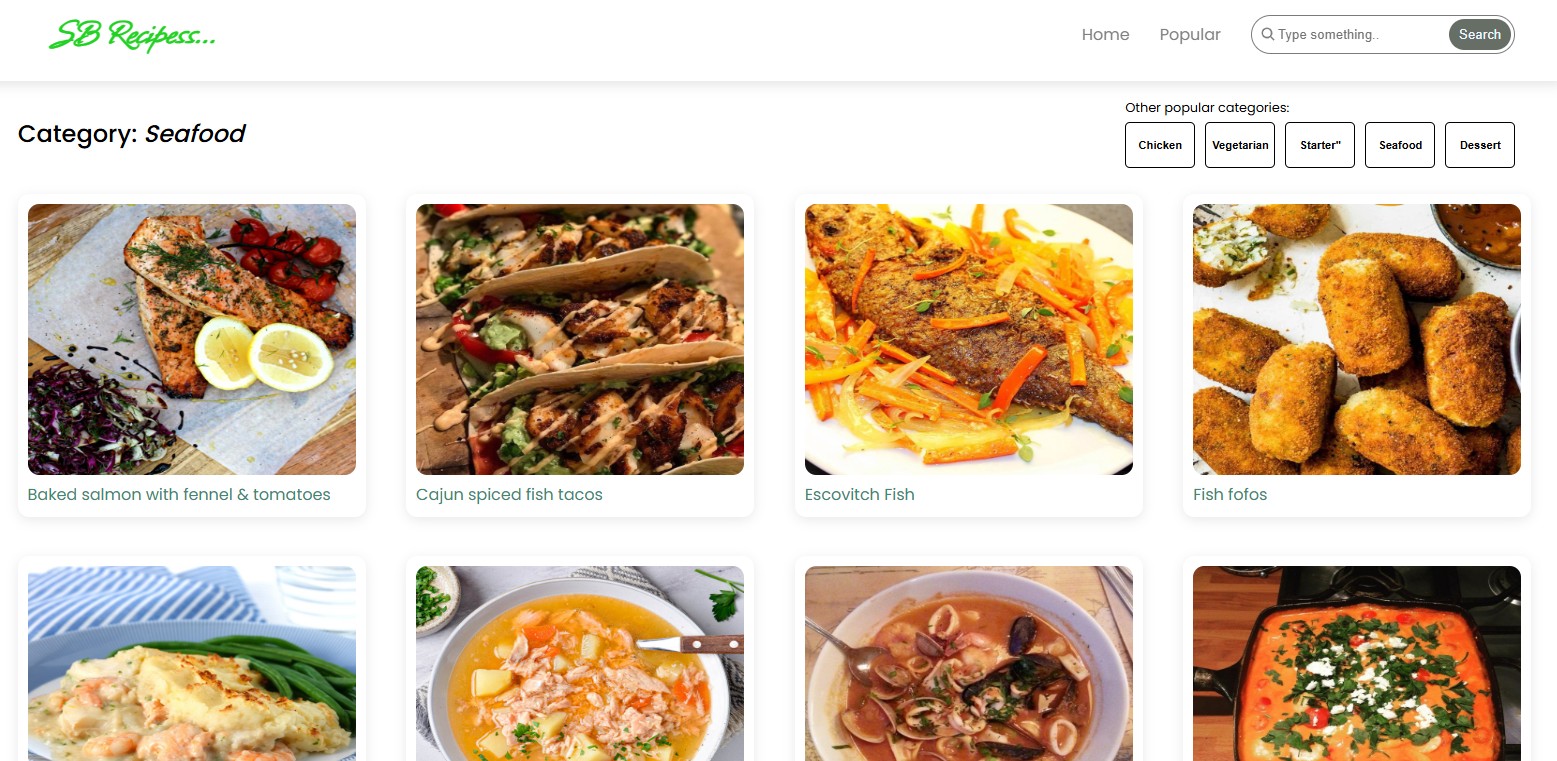












**Known Issues**

* Limited offline support (requires internet for recipe data).
* Pantry sync not persistent without backend/database.
* Mobile performance optimization still in progress.

**Future Enhancements**

* Add backend integration for persistent pantry and user profiles.
* AI-powered recipe suggestions based on pantry ingredients.
* Voice-controlled recipe navigation.
* Advanced filtering (dietary preferences, nutrition info)
* Animated transactions for smoother user experience.