**Frontend Development with React.js – Project Documentation**

**1. Introduction  
 Project Title:**

Cook Book – Your Virtual Kitchen Assistant.

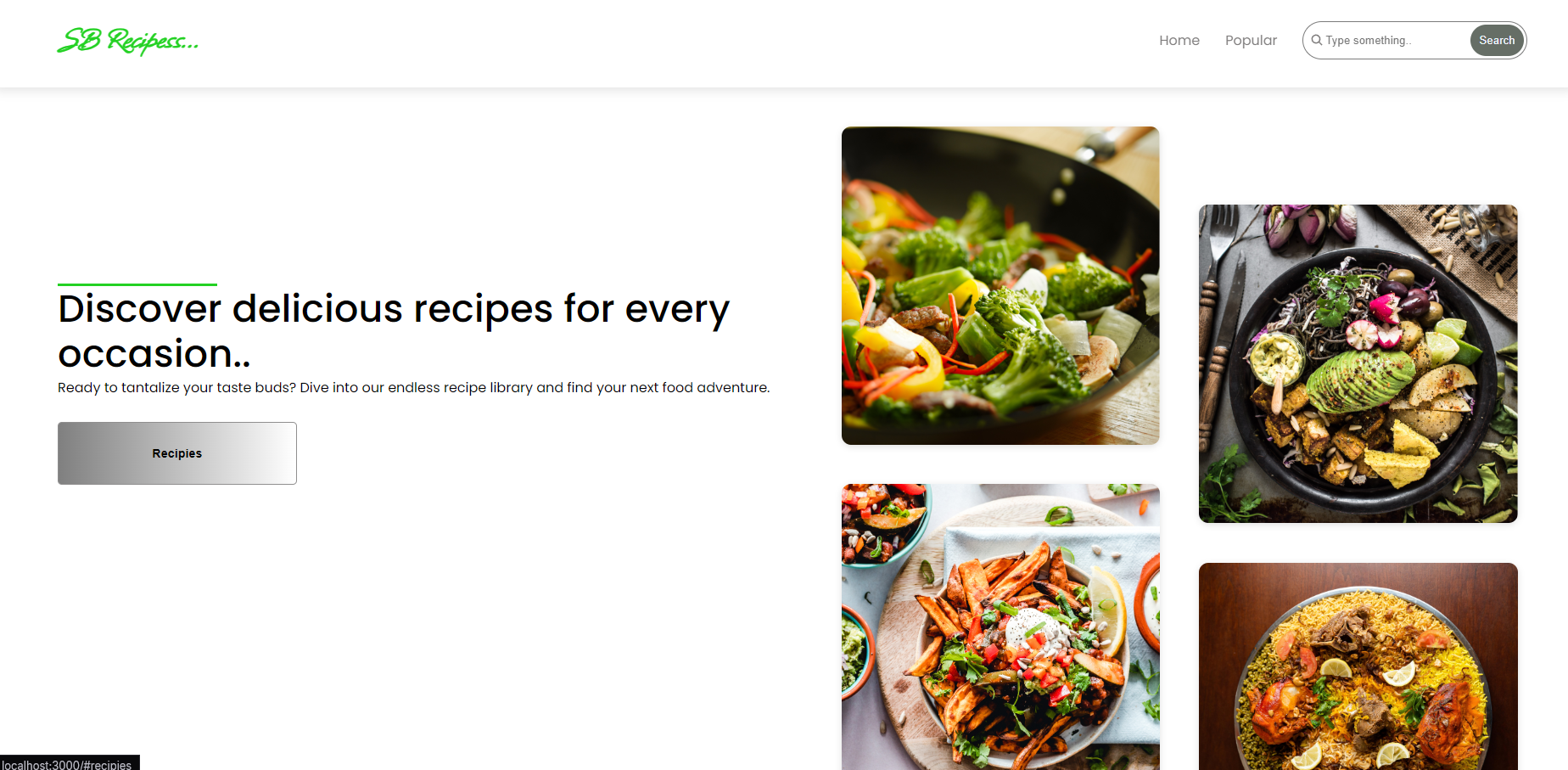
**Team Members:**  
 Akshya D – Frontend Developer  
 Dharsha R – UI/UX Designer  
 Padma Priya S – State Management & Integration  
 Enija Srri P S – Testing & Deployment  
  
**2. Project Overview**  
 **Purpose:** The project aims to provide a virtual kitchen assistant that helps users explore, save, and manage recipes with ease. It focuses on delivering an intuitive UI, responsive design, and a seamless cooking experience.   
 **Features:** Recipe browsing with categories & filters  
 Search functionality with live suggestions  
 User authentication (login/signup with validation)  
 Recipe saving and favorites list  
 Responsive UI for mobile and desktop  
  
**3. Architecture  
 Component Structure:** Node.js – Root component with routing  
 Header, Footer, Navbar – Layout components  
 RecipeList, RecipeDetail – Feature components  
 Auth – Login/Signup formsFavorites – User’s saved recipes  
 **State Management:**  
 Global state handled using Context API (AuthContext, RecipeContext)  
 Local state managed using useState & useReducer  
 Routing (react-router-dom):  
 / → Home  
 /recipes → Recipe list  
 /recipe/:id → Recipe details  
 /login → Login/Signup  
 /favorites → User’s saved recipes

**4. Setup Instructions  
 Prerequisites:** Node.js (v16 or later)  
 npm or yarn  
 **Installation:** git clone https://github.com/your-repo/cook-book.git  
 cd cook-book/client  
 npm install  
 Environment Variables (.env):

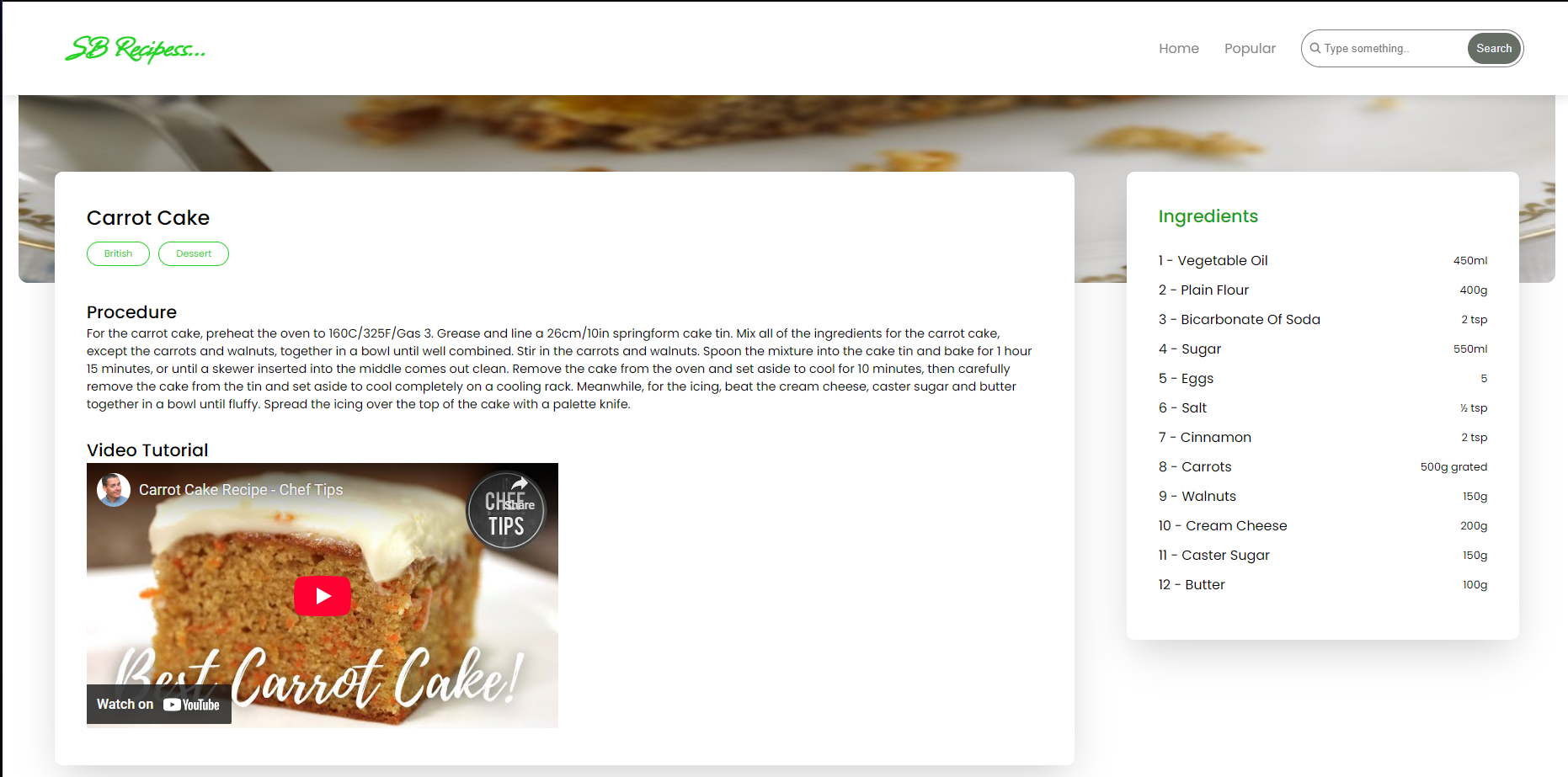
REACT\_APP\_API\_URL=http://localhost:5000/api  
 REACT\_APP\_AUTH\_KEY=your-secret-key  
  
**5. Folder Structure**client/  
public/  
src/  
assets/ # Images, icons, styles  
components/ # Reusable UI components  
pages/ # Page-level components  
context/ # Context API files  
 hooks/ # Custom hooks  
 utils/ # Helper functions  
 Node.js  
 index.js  
 package.json  
 **Utilities**:  
 api.js – API call handler  
 useAuth.js – Custom hook for authentication  
 useFetch.js – Data fetching hook  
  
**6. Running the Application  
 Frontend (React):**  
 cd client  
 npm start  
  
**7. Component Documentation**  
 **Key Components**:  
 RecipeCard – Displays recipe preview (Props: title, image, time)  
 RecipeDetail – Shows recipe details (Props: id)  
 AuthForm – Handles login & signup  
 **Reusable Components:** Button – Configurable button with variant prop.

Input – Reusable input with validation  
  
8. **State Management  
 Global State:** Managed with Context API (AuthContext, RecipeContext)  
 **Local State:**  
 useState for modal/toggle state  
 useReducer for complex form handling  
  
**9. User Interface**  
 **Screenshots Placeholder:**  
 Home Page  
 Recipe Detail Page  
 Login/Signup  
 Favorites Page  
  
**10. Styling  
 CSS Frameworks/Libraries:**  
 Tailwind CSS – Utility-first styling  
 Styled Components – Dynamic styles  
 **Theming:** Light/Dark mode toggle (via Context)  
  
**11. Testing  
 Testing Strategy:** Unit Testing – Jest + React Testing Library  
 Snapshot Testing – Components UI  
 End-to-End Testing – Cypress  
 **Code Coverage:**  
 Configured with Jest coverage reports  
  
**12. Screenshots or Demo**   
 Live Demo (Netlify/Vercel): https://cook-book-demo.netlify.app  
 (Insert screenshots of Home, Recipe Details, and Favorites here)

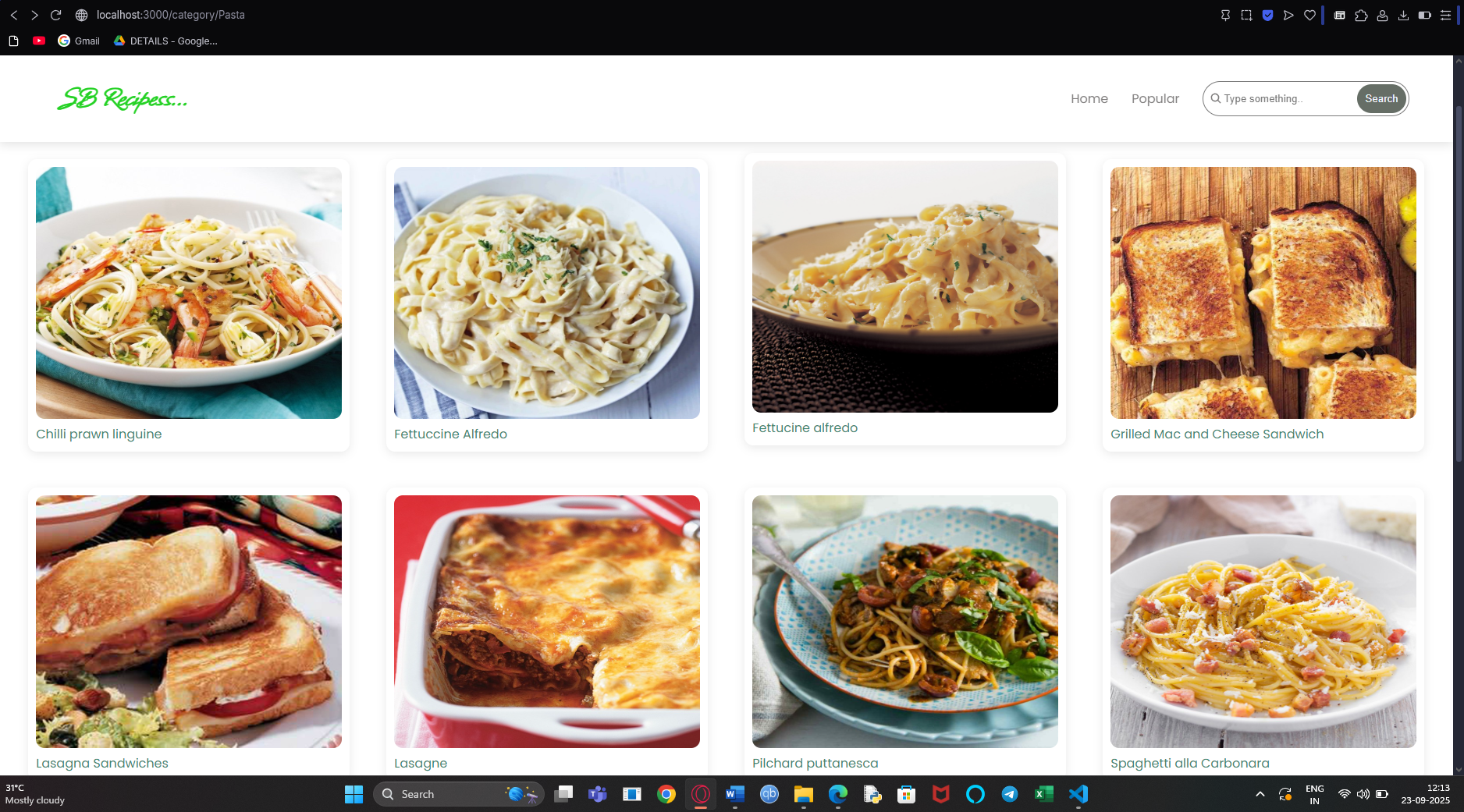
1. **HOME**



1. **RECIPE DETAILS**

****

1. **FAVORITES**



**13. Known Issues:**

Recipe API occasionally returns incomplete data  
 Favorites list does not persist after refresh (without backend integration)  
  
**14. Future Enhancements**  
 Voice-based recipe assistant  
 Meal planner & shopping list integration  
 Push notifications for saved recipes  
 Advanced filtering (calories, cuisine, etc.)