BUSINESS REQUIREMENTS DOCUMENT

Project Overview

The **Smart Recipe Recommender** is a web application designed to help users discover personalized recipes based on their available ingredients, dietary preferences, and cooking time. The platform allows users to manage their pantry inventory, receive smart recipe suggestions, and generate meal ideas quickly and intuitively.

Purpose

The purpose of the **Smart Recipe Recommender** is to provide users with a centralized, easy-to-use system to:

- Get tailored recipe suggestions based on real-time inventory.
- Manage ingredients and track pantry items easily.
- Discover new recipes that match dietary goals, cuisines, and preferences.
- Save time and reduce food waste through smart planning.

Key Features and Requirements

1. Authentication

- Users must be able to **sign up** using email and password.
- Users must be able to **log in securely** to access personal features.
- Forms must handle **input validation** and display appropriate error messages.
- Authentication **tokens** must be securely managed for session maintenance.

2. Inventory Management

- Users can **add**, **edit**, or **delete** ingredients in their pantry.
- Inventory must store details like ingredient name, quantity, and expiration date.
- The system should automatically **flag** ingredients nearing expiry.

3. Smart Recipe Suggestions

- Users receive **recipe recommendations** based on available ingredients.
- Option to **filter recipes** by dietary restrictions (vegan, keto, gluten-free, etc.).
- Recipes should indicate missing ingredients (if any) and suggest quick substitutes.

4. Recipe Browsing and Saving

- Users can **browse recipes** from a curated list or personalized feed.
- Users can **save favorite recipes** for quick access later.
- Users can **rate** and **comment** on recipes if desired.

5. Chatbot Integration

AI-powered **chatbot** assists with ingredient substitutions, meal planning, or cooking tips. The chatbot can suggest quick recipes or answer basic cooking-related queries.

6. Responsive Design

The entire application must be **fully responsive** (desktop, tablet, mobile). Navigation, recipe cards, popups, and forms must function smoothly across all devices.

Technical Requirements

- **Frontend**: Built using **React.js** with responsive UI frameworks.
- **Backend**: REST APIs for Authentication, Inventory, Recipe Management. Tested using Postman.
- **Database**: MongoDB to store user profiles, ingredients, saved recipes, and history.
- **Authentication**: Secure **token-based** authentication for all protected routes.

Non-Functional Requirements

- **Fast loading** pages and recipe suggestions.
- **Minimal downtime** and robust **error handling** for network/API failures. Clean, intuitive, and **accessible** UI/UX. Informative error messages for failed form submissions.

Success Criteria

- Users can **sign up and log in** without issues.
- Users can **manage inventory** and track ingredients easily
- Users receive **relevant recipe recommendations** based on real-time inventory.
- **Chatbot assistant** provides useful, context-aware suggestions.
- All screens are **responsive and visually consistent**.
- APIs correctly handle valid and invalid cases, returning proper responses.

Conclusion

The **Smart Recipe Recommender** aims to make cooking easier, faster, and more sustainable by offering **personalized**, **smart**, and **user-friendly** recipe suggestions. The application should deliver reliability, ease of use, and a modern experience that empowers users to cook more and waste less