Software Requirements Specification – Munchora

This document defines the functional, non-functional, and technical requirements for Munchora, a web application designed to help users search for recipes, generate AI-powered recipes, manage grocery shareable lists across devices.

ID	Functional Requirements
FR001	Allow users to create an account using email and password.
FR002	Support signing in with email and password.
FR003	Support social login - sign in with Google SSO.
FR004	Users can logout.
FR005	Users can view and update profile information (name, bio, profile image).
FR006	Users can delete their account.
FR007	Allow users to search recipes using text queries and filters.
FR008	Users can view recipe details.
FR009	Users can generate a recipe using AI by providing a prompt.
FR010	Users can edit a recipe using AI by providing a prompt.
FR011	Users can edit their generated recipes.
FR012	Users can either share their recipes or keep them private.
FR013	Recipes should be able to like and comment.
FR014	Users can create and delete grocery lists.
FR015	Grocery lists can have items added, updated and removed.
FR016	Ingredients from a recipe should be possible to add directly to a grocery list.
FR017	An grocery list item can be marked as completed.
FR018	Users can share and un-share grocery lists with each other.
FR019	Users can leave a grocery list they don’t own.
FR020	

ID	Usability Requirements

UR001	Intuitive navigation and layout for easy browsing.
UR002	Typography, spacing, and icons must follow a consistent design system (ShadCN + Tailwind).
UR003	The website should be mobile-friendly and accessible on various devices.

ID	Reliability Requirements
RR001	Regular backups of user data and product information to prevent data loss.
RR002	User-friendly error messages and logging for system errors.
RR003	Ensure high availability with minimal downtime, ideally 99.9% uptime.
RP004	Must provide comprehensive auditing for monitoring and error tracking.

ID	Performance Requirements
PR001	Pages should load within 2 seconds under normal conditions.
PR002	Support for at least 1000 concurrent users without performance degradation.
PR003	

ID	Technical Requirements
TR001	Frontend Built using React, TypeScript, Tailwind CSS, React Router v7, ShadCN UI.
TR002	Built using Ruby on Rails (API-only mode).
TR003	Uses MySQL as database.
TR004	System must support efficient React Query caching and pagination.
TR005	Maintain user sessions with JWT or cookie-based authentication.
TR006	Support real-time communication for instant updating grocery lists.
TR007	Maintain a persistent connection with reconnection logic.