# 📘 Recipe Social Media App – Unified Project Plan

## 1. 📌 Vision Statement & Value Proposition

🧑‍🍳 Recipe Social Media App – Vision, Personas & Value Proposition 📌 Vision Statement To create a vibrant and inclusive social platform where home cooks and food enthusiasts can share, discover, and enhance home-cooked recipes—powered by AI-driven personalization and community engagement. The app aims to make cooking more enjoyable, accessible, and creative by blending traditional recipe sharing with modern AI tools that help users customize recipes to their preferences, lifestyles, and dietary needs. 👥 Target User Personas 1. Home Cooks Enjoy preparing meals at home Seek inspiration and validation from others Interested in sharing their creations and exploring new ideas 2. Health-Conscious Individuals Focus on eating healthier or following specific diets (e.g., keto, vegan) Looking for recipe modifications to align with wellness goals Appreciate nutrition info and AI suggestions like “Make Healthier” 3. Beginner Cooks Just starting to cook for themselves Want easy-to-follow, confidence-building recipes Likely to benefit from community support and AI simplification tools 🌟 Unique Value Proposition Social-First Experience A recipe platform designed around community interaction, not just content consumption—users can follow others, engage with posts, and build their cooking identity. AI-Powered Enhancements Tools to modify recipes with one tap: make it healthier, quicker, more flavorful, or adapted to a different cuisine or portion size. Rich Discovery & Personalization Search, filter, and browse trending or personalized feeds. Get tailored suggestions using tags, dietary preferences, and AI. User-Controlled Privacy Users can keep recipes or cookbooks private while maintaining a public presence and engaging socially.

## 2. 🧩 Functional & Non-Functional Requirements

📘 Functional and Non-Functional Requirements Document Social Recipe App (Working Title) 🟢 1. Vision and Overview A social platform for sharing and discovering home-cooked recipes with a vibrant community, featuring AI-powered tools to modify and enhance user-created recipes. 🟢 2. Target Users Home cooks Health-conscious individuals Beginner cooks 🟢 3. Unique Value Proposition Social-first experience with community following and engagement AI-powered recipe recommendations and modifications Rich recipe discovery and sharing capabilities 🟢 4. Functional Requirements 🔹 4.1 User Accounts & Authentication Users must be able to sign up and sign in with: Google Single Sign-On Apple Single Sign-On No email/password login or anonymous guest browsing. 🔹 4.2 User Profiles Each user profile includes: Username Profile photo/avatar Bio/About Me Location Follower and following counts Posted recipes Personal cookbook (saved recipes) Profiles are public by default: Users can mark individual recipes as private (visible only to themselves). Users can set their cookbook to private. User roles: Standard users Admins/Moderators with abilities to delete content or suspend accounts Verified account badges are planned for a future phase, not included in MVP. 🔹 4.3 Recipe Creation Users can create recipes with: - Title - Description - Ingredients (list with quantities) - Preparation steps - Cooking time - Difficulty level - Cuisine/type tags (e.g., Italian, Vegan) - General tags - Nutrition information (potentially generated via AI) - Cover photo (optional, can be auto-generated using AI) 🔹 4.4 Recipe Interaction Users can: - Like/favorite recipes - Comment on recipes - Reply to comments (nested replies) - Like comments - Rate recipes (star rating) - Save recipes to their cookbook - Share recipes externally (social media or direct link) 🔹 4.5 Recipe Discovery & Search Users can: - Search recipes by keyword - Filter recipes by tags/cuisine - Filter by difficulty and cooking time - Browse trending recipes - Receive AI-powered personalized recommendations - View a feed of recipes from followed users 🔹 4.6 AI Features Users can generate AI-powered recipe modifications, including: - Make Healthier: Reduce calories or swap ingredients - Quick Version: Reduce cooking time - Change Cuisine: Adapt to another cuisine - Enhance Flavor: Suggest improvements - Scale Recipe: Adjust quantities for different servings 🔹 4.7 Social & Community Features Follows & Connections - Follow other users - See who follows you - Block other users Feeds - Personalized feed (followed users) - Trending feed - New Recipes feed - Feeds are selectable via tabs - Filter feeds by cuisine Notifications - Recipe liked - Recipe commented on - New follower - Recipe featured - Comment reply Reporting & Moderation - Report recipes and comments - Block/report users - Admin/moderator review tools Messaging - Messaging is not included in MVP 🔹 4.8 Recipe Privacy Users can mark recipes as private (visible only to themselves). 🟢 5. Non-Functional Requirements 🔹 5.1 Performance Recipe feed loads within 2 seconds on average connections. AI-powered recipe suggestions return within 10 seconds. Recipe uploads (including images) complete within 10 seconds. 🔹 5.2 Scalability Designed for up to 1,000 users initially. Firebase to handle auto-scaling beyond this if needed. 🔹 5.3 Availability & Reliability Standard availability target: ~99% uptime. 🔹 5.4 Security All connections encrypted via HTTPS. Tokens securely stored: iOS: Keychain Android: EncryptedSharedPreferences Firestore security rules: Users can edit only their own recipes and profiles. Private recipes visible only to their owner. Two-factor authentication is not required. 🔹 5.5 Privacy & Compliance Minimal privacy requirements for portfolio use. GDPR/CCPA and other compliance to be evaluated later if published. 🔹 5.6 Accessibility Accessibility support (screen readers, color contrast, text scaling) planned for a later phase. 🔹 5.7 Compatibility iOS minimum supported version: iOS 13+ Android minimum supported version: Android 8+ 🟢 6. Monetization (Future Phases) Potential monetization options to be considered after MVP: - In-app Ads - Purchasable credits for premium AI features and recipe library - Affiliate links for ingredients or tools - Sponsored content (featured recipes) - Donations via Patreon or in-app support Timeline: - Monetization is not planned at launch and will be evaluated in later phases if the app is published and gains adoption.

## 3. 🧭 App Navigation & Screen Flows

📱 Screen Flows – Recipe Social Media App This document describes the primary screen flows for the Recipe Social Media App, including onboarding, profile setup, core navigation, manual recipe creation, guided AI recipe creation, social interactions, and cookbook browsing. 1️⃣ Onboarding & Authentication Flow Purpose: Guide users through sign-in and initial setup. Screens: - Welcome Screen - App branding - “Continue with Google” - “Continue with Apple” - SSO Authentication - OAuth flow via provider - Profile Setup (First-Time Only) - Upload profile photo - Choose username - Optional: Bio and Location - Onboarding Carousel (optional) - Short introduction slides explaining: - Feed for browsing - Create tab for posting recipes - FAB for AI-generated recipes - Cookbook for saved recipes - Profile and settings - Landing on Feed tab Flow Summary: Welcome ➡️ SSO Authentication ➡️ Profile Setup ➡️ Onboarding Carousel ➡️ Feed Tab 2️⃣ First-Time Profile Edit Flow Purpose: Allow new users to complete their profile right after onboarding. Screens: - My Profile (initial view) - Prompt to complete profile (banner or modal) - Edit Profile button - Edit Profile - Update username, bio, location - Change profile photo - Save changes Flow Summary: Profile Tab ➡️ Edit Profile ➡️ Save 3️⃣ Tab Bar Navigation Overview After onboarding, users see the bottom tab bar, which is persistent across all screens: Tabs: 1. Feed - Browse and discover recipes. 2. Cookbook - View saved recipes. 3. + (Create) - Manually create new recipes. 4. Chefbot - Explore AI tools and recommendations. 5. Profile - Manage profile and settings. Behavior: - Tapping Create (+) always opens the manual recipe editor. - Tapping other tabs maintains navigation context. ✨ FAB Behavior Floating Action Button (FAB): - Visible only on the Feed screen. - Purpose: Launch the Guided AI Recipe Assistant. - Behavior: - Appears when scrolling up. - Hides when scrolling down. - Tapping starts a multi-step guided creation flow. 4️⃣ Home Feed & Discovery Flow (Feed Tab) Purpose: Browse and discover recipes. Screens: - Home Feed - Tabs: Trending | Following | New - Filters: Cuisine, Cooking Time, Difficulty - Search bar - Recipe cards - FAB: “AI Recipe Assistant” (appears when scrolling up) - Recipe Details - Title and description - Ingredients and instructions - Ratings and comments - Like, Save, Share - “Enhance with AI” button - Search & Filters - Keyword search - Tag selection - Cuisine filter - External Share - OS-native share sheet Flow Summary: Feed Tab ↔️ Recipe Details ↔️ Search/Filters ↔️ Share 5️⃣ Manual Recipe Creation Flow (+ Tab) Purpose: Enable users to create and publish recipes manually. Access: Tapping the + (Create) tab in the navigation bar. Screens: - Recipe Editor - Title and Description - Ingredients (add list) - Preparation steps - Cuisine and Tags - Difficulty, Cooking Time, Servings - Cover Photo (upload or AI-generated) - Nutrition Info (optional) - Privacy Setting - Public / Private toggle - Confirmation & Publish - Return to Feed or Cookbook Flow Summary: Create Tab ➡️ Editor ➡️ Privacy ➡️ Publish 6️⃣ Guided AI Recipe Assistant Flow (Feed FAB) Purpose: Help users generate a personalized AI recipe via guided questions. Access: Tapping the FAB in the Feed screen. Screens: Meal Type Breakfast, Lunch, or Dinner? Dietary Choices Vegan, Vegetarian, Keto, Paleo, No preference Regional Cuisine Italian, Mexican, Indian, Chinese, etc. Kid Friendly Yes / No Portion Quantity 1–10 servings Portion Size Small, Medium, Large Ingredient Exclusions Multi-select or free text input Spice Level Mild, Medium, Spicy Confirmation Review selections “Generate Recipe” button Generating Recipe Loading animation AI Recipe Preview Title, Description Ingredients and Instructions Nutrition info Optional image Actions: Save to Cookbook Publish Share externally Flow Summary: Feed FAB ➡️ Guided Questions (1–8) ➡️ Confirmation ➡️ Generate ➡️ Preview ➡️ Save 7️⃣ AI Exploration Flow (Chefbot Tab) Purpose: Provide broader AI tools and recipe enhancement options. Screens: - Chefbot Home - Predefined AI suggestions - Recent enhancements - Quick actions: - Make Healthier - Quick Version - Change Cuisine - Enhance Flavor - Scale Recipe - Enhancement Options - Select enhancement type - Customization Parameters - Generating Enhancement - Enhanced Recipe Preview - Save to Cookbook - Share externally Flow Summary: Chefbot Tab ➡️ Enhancement Options ➡️ Customize ➡️ Generate ➡️ Preview ➡️ Save 8️⃣ Cookbook Flow (Cookbook Tab) Purpose: View and manage saved recipes. Screens: - Cookbook Home - List of saved recipes - Filters (Tags, Cuisine) - Privacy toggle (Public/Private) - Recipe Details (from saved list) - Remove from Cookbook confirmation Flow Summary: Cookbook Tab ↔️ Recipe Details ↔️ Remove or Share 9️⃣ Social & Community Flow Purpose: Support following, community engagement, and notifications. Screens: - User Profiles - Profile info - Follow / Unfollow - Followers / Following counts - Posted recipes - Followers List - Following List - Notifications - Likes - Comments - Follows - Enhancements - Report / Block Modal Flow Summary: Feed / Profile ↔️ User Profiles ↔️ Followers/Following ↔️ Notifications 🔟 Ratings & Comments Flow Purpose: Enable recipe feedback and discussion. Screens: - Recipe Details - Ratings summary - Comments section - Add Rating Modal - Star selection (1–5) - Add Comment Modal - Text input - Reply to Comment - Nested replies - Like Comment - Inline action Flow Summary: Recipe Details ↔️ Add Rating ↔️ Comments ↔️ Replies 🔔 Error & Empty States Key Screens: - No Internet Connection - Empty Feeds or Cookbook - No Search Results - Profile Incomplete - Recipe Upload Failed

## 4. 🗃️ Data Models & 🔐 Security Rules

📘 Recipe Social Media App – Data Models & Firestore Design (Updated) Table of Contents User Model Saved Recipes Subcollection Followers and Following Subcollections Recipe Model Recipe Subcollections Comments Ratings Likes Enhancements Security Rules (Examples) 1️⃣ User Model Collection Path: users/{uid} (uid is the Firebase Authentication UID) Fields: Example Document: { “uid”: “user\_123”, “email”: “user@example.com”, “username”: “foodie101”, “profileImageUrl”: “https://example.com/avatar.jpg”, “bio”: “Passionate about healthy cooking.”, “location”: “Toronto, Canada”, “createdAt”: “2025-07-14T12:00:00Z”, “lastLoginAt”: “2025-07-14T19:00:00Z”, “preferences”: { “dietary”: “vegetarian”, “notifications”: true }, “isAdmin”: false, “verified”: false, “isCookbookPrivate”: false } 2️⃣ Saved Recipes Subcollection Path: users/{uid}/savedRecipes/{recipeId} Fields: Example Document: { “recipeId”: “recipe\_456”, “savedAt”: “2025-07-14T20:00:00Z”, “notes”: “Try with almond milk.”, “tags”: [“Favorites”, “Breakfast”] } 3️⃣ Followers and Following Subcollections Followers Path: users/{uid}/followers/{followerUid} Following Path: users/{uid}/following/{followingUid} Fields: Example Document: { “followedAt”: “2025-07-14T18:30:00Z” } 4️⃣ Recipe Model Collection Path: recipes/{recipeId} Fields: Example Document: { “id”: “recipe\_456”, “authorId”: “user\_123”, “title”: “Classic Pancakes”, “description”: “Fluffy pancakes perfect for breakfast.”, “ingredients”: [ {“name”: “Flour”, “quantity”: “2 cups”}, {“name”: “Milk”, “quantity”: “1.5 cups”}, {“name”: “Eggs”, “quantity”: “2”} ], “instructions”: [ “Mix dry ingredients.”, “Add wet ingredients.”, “Cook on griddle.” ], “coverImageUrl”: “https://example.com/pancakes.jpg”, “additionalImages”: [], “cuisine”: “American”, “tags”: [“breakfast”, “easy”], “nutritionInfo”: { “calories”: 300, “protein”: “8g”, “carbs”: “40g”, “fat”: “10g” }, “cookingTime”: 20, “difficulty”: “Easy”, “servings”: 4, “isPublic”: true, “likesCount”: 15, “commentsCount”: 3, “rating”: { “average”: 4.5, “count”: 10 }, “createdAt”: “2025-07-14T15:00:00Z”, “updatedAt”: “2025-07-14T15:00:00Z” } 5️⃣ Recipe Subcollections 5.1 Comments Path: recipes/{recipeId}/comments/{commentId} Fields: Example Document: { “id”: “comment\_123”, “authorId”: “user\_456”, “text”: “I love this recipe!”, “createdAt”: “2025-07-14T21:00:00Z”, “likesCount”: 2, “parentCommentId”: null, “isDeleted”: false } Comment Likes Subcollection Path: recipes/{recipeId}/comments/{commentId}/likes/{userId} 5.2 Ratings Path: recipes/{recipeId}/ratings/{userId} 5.3 Likes Path: recipes/{recipeId}/likes/{userId} 5.4 Enhancements Path: recipes/{recipeId}/enhancements/{enhancementId} Fields: Example Document: { “id”: “enhancement\_123”, “recipeId”: “recipe\_456”, “createdBy”: “user\_789”, “enhancementType”: “MAKE\_HEALTHIER”, “title”: “Classic Pancakes (Healthier)”, “description”: “Reduced sugar and added whole wheat flour.”, “ingredients”: […], “instructions”: […], “nutritionInfo”: {…}, “coverImageUrl”: “https://example.com/healthier-pancakes.jpg”, “cuisine”: “American”, “createdAt”: “2025-07-14T22:00:00Z”, “notes”: “Wanted to lower calories.” } 6️⃣ Security Rules (Examples) User Profile & Saved Recipes: match /users/{userId} { allow read: if true; allow write: if request.auth.uid == userId;

match /savedRecipes/{recipeId} { allow read: if !get(/databases/(userId)).data.isCookbookPrivate || request.auth.uid == userId; allow write: if request.auth.uid == userId; }

match /followers/{followerId} { allow read, write: if request.auth.uid == userId; }

match /following/{followingId} { allow read, write: if request.auth.uid == userId; } } Recipes & Subcollections: match /recipes/{recipeId} { allow read: if resource.data.isPublic == true || request.auth.uid == resource.data.authorId;

allow write: if request.auth.uid == resource.data.authorId;

match /comments/{commentId} { allow read: if true; allow write: if request.auth.uid != null;

match /likes/{userId} {  
 allow read: if true;  
 allow write: if request.auth.uid == userId;  
}

}

match /ratings/{userId} { allow read: if true; allow write: if request.auth.uid == userId; }

match /likes/{userId} { allow read: if true; allow write: if request.auth.uid == userId; }

match /enhancements/{enhancementId} { allow read: if true; allow write: if request.auth.uid != null; } }