**Product Requirements Document (PRD)**

**Project: AI-Powered Recipe Generator (MVP)**

**Date: July 31, 2025**

**Owner: Rafay Adeel**

**1. Objective**

To build an MVP of an AI-powered recipe generation web application that enables users to input ingredients and receive a complete recipe generated by AI, with functionality to save and view past recipes.

**2. Key Features (MVP Scope)**

* **Authentication**: Magic link email login via Supabase.
* **Recipe Generation**:
  + Input ingredients
  + Output includes:
    - Title
    - List of ingredients with quantities
    - Step-by-step instructions
    - Estimated prep and cook time
* **Save Recipes**:
  + Auto-save generated recipe to database
  + View list of saved recipes (linked to logged-in user)

**3. User Stories**

1. **As a user**, I want to log in using my email so that I can use the app securely.
2. **As a user**, I want to input ingredients and receive a recipe with full details.
3. **As a user**, I want the generated recipe to be automatically saved to my account.
4. **As a user**, I want to view a list of all my saved recipes.

**4. System Architecture**

**Frontend**

* **Framework**: Next.js 14 (App Router)
* **Styling/UI**: TailwindCSS, ShadCN/UI
* **Hosting**: Vercel

**Authentication**

* **Tool**: Supabase Magic Link

**AI Logic**

* **Platform**: n8n
* **Model**: OpenAI GPT-3.5 or GPT-4

**Databases**

* **Supabase**: For authentication and saving recipe references (IDs)
* **MongoDB Atlas**: For storing full recipe objects

**5. API Endpoints**

**POST /api/generate-recipe**

* Verifies user authentication
* Sends ingredients to n8n webhook
* Receives and parses recipe JSON
* Saves recipe to MongoDB
* Stores mongo\_recipe\_id in Supabase
* Returns the recipe object

**GET /api/saved-recipes**

* Verifies user authentication
* Fetches mongo\_recipe\_ids from Supabase
* Retrieves recipes from MongoDB using those IDs
* Returns a list of saved recipes

**6. Database Schema**

**Supabase**

**users (auto-managed by Supabase)**

**saved\_recipes**

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| id | UUID | Primary key |
| user\_id | UUID | Foreign key to users table |
| mongo\_recipe\_id | String | ID of recipe in MongoDB |

**MongoDB**

**recipes Collection**

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| \_id | ObjectId | Primary key |
| title | String | Recipe title |
| ingredients | Array of Objects | Ingredients + quantities |
| instructions | Array of Strings | Step-by-step instructions |
| prepTime | String | Estimated prep time |
| cookTime | String | Estimated cook time |
| userId | String | Reference to Supabase user |

**7. AI Workflow (n8n)**

1. Webhook trigger receives ingredients
2. Builds a prompt:

arduino

CopyEdit

"Generate a recipe using the following ingredients: [ingredients]. Include title, ingredients with quantity, instructions, prep time, and cook time."

1. Sends to OpenAI API
2. Parses response into structured JSON
3. Returns JSON to Next.js API route

**8. File Structure**

recipe-generator/

├── app/

│ ├── (auth)/login/page.tsx

│ ├── (auth)/callback/page.tsx

│ ├── api/generate-recipe/route.ts

│ ├── api/saved-recipes/route.ts

│ ├── recipes/generate/page.tsx

│ ├── recipes/saved/page.tsx

│ ├── layout.tsx

│ └── globals.css

├── components/

│ ├── LoginForm.tsx

│ ├── RecipeGeneratorForm.tsx

│ ├── RecipeCard.tsx

│ └── SavedRecipeItem.tsx

├── lib/

│ ├── supabase/client.ts

│ ├── mongodb.ts

│ └── types.ts

├── services/

│ ├── recipeService.ts

│ └── authService.ts

├── styles/index.css

├── .env.local

├── next.config.mjs

├── tailwind.config.ts

├── tsconfig.json

└── README.md

**9. Task Breakdown**

**Phase 1: Auth and Layout**

* Set up Supabase project
* Enable magic link login
* Build login + callback pages
* Add navbar with login/logout logic

**Phase 2: Recipe Generator**

* Create form for ingredient input
* Create API route: /api/generate-recipe
* Connect to n8n
* Display result with card UI
* Auto-save generated recipe

**Phase 3: Saved Recipes**

* Create API route: /api/saved-recipes
* Retrieve saved recipe IDs from Supabase
* Fetch recipes from MongoDB
* Render recipe list

**Phase 4: Polish and Deploy**

* Add loading and error states
* Ensure responsive UI
* Set up environment variables
* Deploy to Vercel with GitHub

**10. Tools & Setup Checklist**

* Supabase project created with email magic link auth
* MongoDB Atlas cluster setup
* n8n workflow deployed and connected to OpenAI
* OpenAI API key secured
* Vercel project connected to GitHub repo