Web-Based Cooking & Recipes Application

1. Executive Summary

The Daily Meal is a progressive web application (PWA) that empowers home cooks and food enthusiasts to discover, manage, and share recipes with the guidance of an AI-powered virtual chef, "Chef Rania." Built with React, Node.js, Express, and MongoDB, The Daily Meal combines traditional community features—favorites, ratings, user submissions—with advanced AI search via Perplexity and computer vision for image-based ingredient recognition. A data-seeding process preloads the database with popular recipes, ensuring a rich out-of-the-box experience.

3. Target Audience & Personas

1. **Novice Cook Nora:** Seeks step-by-step guidance; uses AIChat for ingredient substitutions.
2. **Enthusiast Evan:** Browses trending recipes; rates and favorites frequently.
3. **Content Creator Carla:** Submits original recipes; monitors review status; engages with comments.

4. Feature List & User Stories

Feature	User Story
Recipe Discovery	As Nora, I browse "Recipe of the Day" and filter by dietary preferences.
Search & Filter	As Evan, I live-search recipes by ingredient and prep time.
Recipe Detail & Interaction	As Evan, I view full instructions, rate recipes, and add to favorites.
User Submissions	As Carla, I submit a new recipe and edit after approval.
AIChat with Chef Rania	As Nora, I ask Chef Rania for recipe ideas based on ingredients I upload from my fridge.
Image-Based Suggestions	As Nora, I snap a photo of my vegetables and get recipe recommendations.
Social & Sharing	As Evan, I share a recipe link to social media and copy it to clipboard.
Admin Moderation	As Admin, I review, approve, or reject user-submitted recipes in a dashboard.

5. Technical Architecture

- **Frontend (PWA):** React with React Router, Axios, styled-components, react-dropzone.
- **Backend:** Node.js, Express.js, JWT auth, Multer for uploads.
- **Database:** MongoDB with Mongoose ODM; Redis for caching "Recipe of the Day."
- **AI & Vision:** Perplexity HTTP API for text queries; Google Vision API for image analysis; YouTube Data API for video links.
- **Seeding & Scraping:** Node.js script using Cheerio/Puppeteer to collect initial recipe and category data.

6. Data Models & API Contracts

- **User:** `{ name, email, passwordHash, avatarUrl, favorites: [ObjectId], role }`
- **Recipe:** `{ title, description, ingredients, steps, imageUrl, youtubeUrl, categoryId, kitchenId, authorId, ratings, favoritesCount, status }`
- **AIInteraction:** `{ userId, type: "text"|"image", query, response, timestamp }`

Key Endpoints

- Auth: `POST /api/auth/signup`, `POST /api/auth/login`, `POST /api/auth/reset`
- Recipes: `GET /api/recipes`, `GET /api/recipes/:id`, `POST /api/recipes`, `PUT /api/recipes/:id`, `DELETE /api/recipes/:id`
- Actions: `POST /api/recipes/:id/rate`, `POST /api/recipes/:id/favorite`
- AI: `POST /api/ai/chat`, `POST /api/ai/image`

7. AI & Vision Integration Details

1. **Text Chat:** Client sends message to `/api/ai/chat`; server proxies to Perplexity, formats response in Chef Rania's voice, returns list of recipe links and tips.
2. **Image Chat:** Client uploads image to `/api/ai/image`; server uses Vision API to detect ingredients, then queries Perplexity + YouTube Data API, returns curated suggestions.

8. Seeding Strategy & Data Sources

- **Blogs:** Top 10 public recipe blogs scraped weekly.
- **YouTube:** Playlists from 5 popular cooking channels.
- **Seeding Script:** Run `npm run seed` populates collections with ~500 recipes across 10 categories and 5 kitchens.