Documentation: FoodieHub — Recipe Platform

PROJECT NAME :Cookbook-your virtual assistant

**NAME:**

TEAM LEADER: VINODHINI .Y

TEAM MEMBER: DIVYA .R

TEAM MEMBER: INISIYA .M

TEAM MEMBER: BHAVANA PRIYA .M

Table of Contents:

*1. Overview*

*2. Architecture & Tech Stack*

*3. User Roles & Permissions*

*4. Features & Flows*

*5. Data Models*

*6. UI / Pages Layout*

*7. API Endpoints*

*8. Integration Points*

*9. Testing & Quality Assurance*

*10. Deployment & Hosting*

*11. Maintenance & Roadmap*

*12. Contribution Guide (if open-source)*

**1.overview**:

Trend & seasonal features



**Name: FoodieHub**

**Purpose: A recipe platform designed to let users discover, browse, save, and rate recipes.**

**Target Users: Home cooks, recipe creators, food lovers.**

**Key Value Propositions:**

**Wide variety of recipes (breakfast, lunch, dinner, desserts, healthy, quick meals)**

**Ratings & Reviews**

**2. Architecture & Tech Stack**

**Frontend: (e.g.) Static site / Jamstack / React / Vue / etc.**

**Backend: (if applicable)**

**Database / Storage: (e.g.) MySQL / MongoDB / NoSQL / Cloud storage for images**

**Hosting & Delivery: GitHub Pages / Netlify / Vercel etc.**

**Third-party services: e.g. Authentication, Image CDN, Analytics**

**3. User Roles & Permissions**

**Role Permissions**

**Visitor / Guest Browse recipes; view ratings; trending / latest recipes**

**Registered User Save favorites; rate recipes; leave reviews; contribute new recipes**

**Admin / Moderator Approve/edit/delete recipes & reviews; manage user accounts; manage content categories**

**4. Features & User Flows**

****

****

**1. Recipe name, image(s), ingredients, preparation steps, cook time, servings**

**2. Ratings & Reviews**

**3. User Actions**

**4. SignUp / Login**

**5. Save / bookmark recipe**

**6. Rate & Review**

**7. Add new recipe (if allowed)**

**8. Admin Functionality**

**9. Content moderation**

**10. Manage categories**

**11. Manage users**

**5. Data Models**

**Here are example database-entities and their fields:**

1. **User**
2. **Id**
3. **Name**
4. **Email**
5. **password\_hash**
6. **role (guest, user, moderator, admin)**
7. **joined\_date**
8. **saved\_recipes (relation)**

**Recipe**

****

1. **id**
2. **author\_id (User)**
3. **title**
4. **category (Breakfast, Dessert, etc.)**
5. **image\_url(s)**
6. **description**
7. **ingredients (list)**
8. **steps (list)**
9. **prep\_time / cook\_time / total\_time**

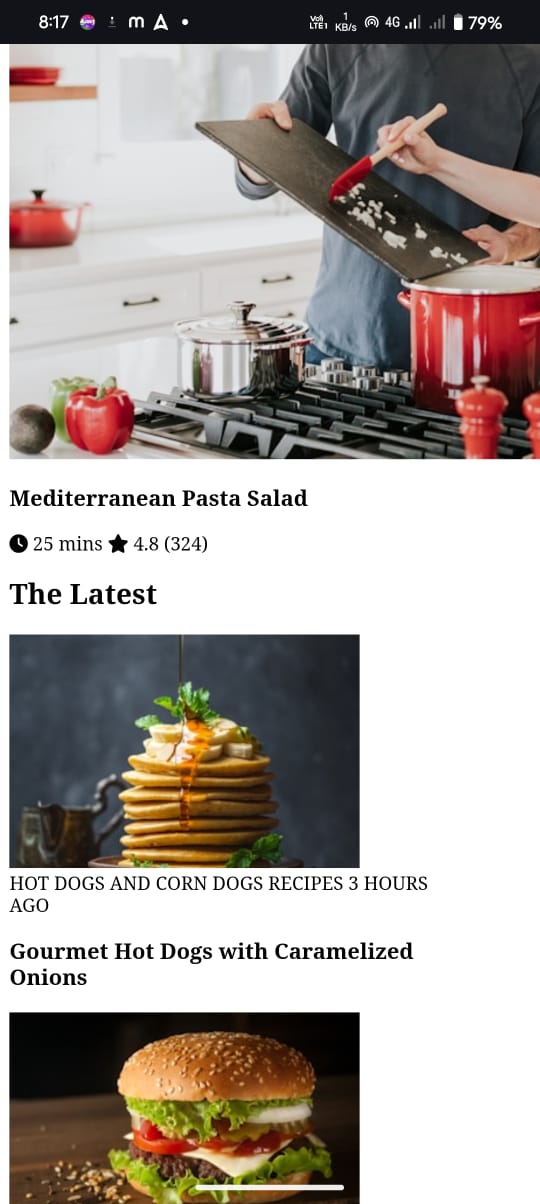
**Review**

1. **Id**
2. **recipe\_id**
3. **user\_id**
4. **rating (e.g. 1-5)**
5. **comment**
6. **date\_posted**

****

****

1. **id**
2. **name**
3. **description**
4. **UI / Pages Layout**

****

**List of pages / screens, with layout details:**

**Page Components / Regions**

**Home / Landing Hero image, featured recipe, trending / latest recipes, categories navigation**

**Category Listing Filters, recipe cards (image, title, rating), pagination/infinite scroll**

**Recipe Detail Large image; recipe title; author; prep/cook time; ingredients; steps; reviews section**

**Login / Signup Form fields; social login (if applicable)**

**Profile / Saved Recipes List of saved recipes; option to edit profile**

**Admin Dashboard Manage content/users; review pending submissions**

**7. API Endpoints**

**(If there is backend API) Sample endpoints might include:**

**Method Endpoint Purpose Parameters / Request Body Response**

**GET /recipes List all recipes (with filters) category, sort, page, search term List of recipes**

**GET /recipes/{id} Get recipe details recipe id Recipe object with all fields**

**POST /recipes Create a new recipe recipe data JSON Created recipe / status**

**PUT / PATCH /recipes/{id} Update recipe recipe id + data Updated recipe**

**DELETE /recipes/{id} Remove recipe recipe id Success/failure**

**POST /recipes/{id}/review Add a review rating, comment, user auth Created review**

**POST /users/login Authenticate user email, password Auth token or session**

**POST /users/signup Register user name, email, password User object / token**

**8. Integration Points**

**Authentication / User Management — possibly via OAuth or custom**

**Image Hosting / CDN — for recipe images**

**Analytics — Google Analytics or similar to track usage, popular recipes**

**Email Service — for welcome emails, password reset, etc.**

**Search Engine / Full-text Search — to enable recipe searching**

**9.Testing & Quality Assurance**

**Unit Tests — for individual functions/models**

**Integration Tests — for API endpoints**

**UI / End-to-end Tests — simulate user flows (browsing, saving, reviewing)**

**Performance Testing — page load times, image sizes, mobile responsiveness**

**Accessibility — alt tags, ARIA, keyboard navigation**

**10. Deployment & Hosting**

**Hosting Platform: e.g. GitHub Pages (if static), or another cloud provider if dynamic**

**CI/CD Pipeline: automated build, test, deployment**

**Environments: Development, Staging, Production**

**Backup / Data Persistence: how to store and backup user data, recipes, images**

**11. Maintenance & Roadmap**

**Regular Tasks: content moderation; updating dependencies; backups**

**Feature Roadmap Ideas:**

**Social sharing of recipes**

**Personalized recipe recommendations**

**Mobile App**

**Multilingual support**

**Video instructions**

**Shopping list generation**

**12. Contribution Guide**

**If this is open to collaboration, include:**

**How to set up local dev environment**

**Coding conventions / style guide**

**Pull request process**

**Branching model**

**THANKYOU**