

Project Documentation

COOKBOOK-YOUR VIRTUAL KITCHEN ASSISTANT

1. Introduction

- **Project Title:** Cookbook
- **Team ID:** NM2025TMID29941
- **Team Leader:** PRIYADHARSHINI N
- **-202400661@sigc.edu**
- **Team Members:** LAVANYA E
- **-202400450@sigc.edu**
- **RAMYA V**
- **-202400660@sigc.edu**
- **DEEPA SRI AP**
- **-202400171@sigc.edu**
- **JEGATHEESHWARI A**
- **-202400357@sigc.edu**

2. Project Overview

- **Purpose:**
Cookbook is a recipe management and discovery platform. It allows users to post, edit, and explore recipes, filter by ingredients/cuisine, and generate shopping lists.
- **Features:**
 - Recipe posting and editing
 - Recipe search & filtering by ingredients, cuisine, or category
 - Favorites collection
 - Shopping list generator
 - Comment and review system
 - Admin dashboard for moderation

3. Architecture

- **Frontend:** React.js with Bootstrap and Material UI
- **Backend:** Node.js with Express.js (API endpoints, authentication, recipe management)
- **Database:** MongoDB with Mongoose (for storing recipes, users, categories, favorites, comments)
- **Optional Services:** Cloud storage for recipe images (AWS S3/Cloudinary)

4. Setup Instructions

- **Prerequisites:**
 - Git
 - React.js
 - Express.js
 - Mongoose
 - Visual Studio Code
 - Node.js & MongoDB
- **Installation Steps:**
 1. Clone the repository:
 2. `git clone https://github.com/your-repo/cookbook.git`
 3. `cd cookbook`
 4. Install client dependencies:
 5. `cd client`
 6. `npm install`
 7. Install server dependencies:
 8. `cd server`
 9. `npm install`

5. Folder Structure

```
Cookbook/
|-- client/                # React frontend
|   |-- components/
|   |-- pages/
|   |-- assets/
|-- server/                # Node.js backend
|   |-- routes/
|   |-- models/
|   |-- controllers/
|   |-- middleware/
|-- package.json
|-- README.md
```

6. Running the Application

- **Frontend:**
 - `cd client`
 - `npm start`
- **Backend:**
 - `cd server`
 - `npm start`
- **Access:**
Visit <http://localhost:3000>

7. API Documentation

- **User:**
 - `POST /api/user/register`
 - `POST /api/user/login`
- **Recipes:**
 - `POST /api/recipes/create`
 - `GET /api/recipes/:id`
 - `GET /api/recipes/search?ingredient=tomato`
- **Favorites:**
 - `POST /api/favorites/add`
 - `GET /api/favorites/:userId`
- **Comments:**
 - `POST /api/comments/add`
 - `GET /api/comments/:recipeId`

8. Authentication

- **JWT-based authentication** for secure login
- **Middleware** to protect private routes (recipe posting, favorites, reviews)

9. User Interface

- **Landing Page** – Recipe highlights & search
- **Recipe Details Page** – Full recipe with ingredients, steps, and reviews
- **User Dashboard** – Manage user's recipes, favorites, and shopping list
- **Admin Panel** – Moderate recipes and manage users

10. Testing

- **Manual testing** at each development milestone
- **Tools:** Postman (API testing), Chrome DevTools (frontend testing)

11. Screenshots or Demo

<https://drive.google.com/file/d/1-xd0bYO998lc4j99DmmgSOW2SqZCOcy3/view?usp=drivesdk>

12. Known Issues

- Image upload may fail on low bandwidth
- Search speed decreases with very large dataset
- No offline support yet

13. Future Enhancements

- AI-powered recipe recommendations
- Nutrition calculator per recipe
- Voice-assisted cooking mode
- Mobile app (iOS/Android)