

Project Documentation

1. Project Title

Project Title: CookBook: Your Virtual Kitchen Assistant

Team ID: NM2025TMID47275

Team Leader:[Name: Monisha.A (Mail ID:monialagu111@gmail.com)]

Team Members:

[Name:Manoran jitham.M (Mail ID:mano532ranjitham@gmail.com)]

[Name:Mohanapriya.M (Mail ID:mohanpriya2199@gmail.com)]

[Name:Mugila.M (Mail ID:mugilamurugan2007@gmail.com)]

[Name:Monisha.C (Mail ID:monishachennappan2311@gmail.com)]

2. Project Overview

Purpose:

CookBook is a virtual kitchen assistant web application designed to help users discover, manage, and cook recipes efficiently. It provides personalized recipe suggestions, ingredient tracking, and step-by-step cooking guidance.

Features:

- Recipe search and filtering by ingredients, cuisine, or dietary needs
- Smart pantry management (track available ingredients)
- Step-by-step cooking mode with timers
- Favorites & shopping list
- User reviews and ratings
- Admin panel for recipe management

3. Architecture

- Frontend: React.js with Material UI and Bootstrap
- Backend: Node.js + Express.js (API handling, recipe logic)
- Database: MongoDB (stores recipes, users, ingredients, favorites, shopping lists)

4. Setup Instructions

Prerequisites:

- Node.js
- MongoDB
- Git
- React.js
- Express.js
- Visual Studio Code

Installation Steps:

```
# Clone the repository
git clone
```

```
# Install client dependencies
cd client
```

```
npm install
```

```
# Install server dependencies  
cd ../server  
npm install
```

5. Folder Structure

```
CookBook/  
|-- client/ # React frontend  
| |-- components/  
| |-- pages/  
|  
|-- server/ # Node.js backend  
| |-- routes/  
| |-- models/  
| |-- controllers/
```

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:
– GET /api/recipes (fetch all recipes)
– GET /api/recipes/:id (fetch recipe by ID)
– POST /api/recipes/create (admin only)

Pantry & Shopping List:
– GET /api/pantry/:userId
– POST /api/pantry/add
– POST /api/shopping-list/add

Favorites:
– POST /api/favorites/add
– GET /api/favorites/:userId

8. Authentication

- JWT-based authentication for secure login
- Middleware for protecting private routes

9. User Interface

- Landing Page (search recipes)
- User Dashboard (pantry, shopping list, favorites)
- Recipe Details Page (step-by-step instructions)
- Admin Panel (recipe management)

10. Testing

- Manual Testing during milestones
- Tools: Postman, Chrome Dev Tools

11. Screenshots or Demo

Landing Page – Recipe search bar and trending recipes
User Dashboard – Pantry items, shopping list, saved favorites
Recipe Details Page – Ingredients, cooking steps with timers
Admin Panel – Add / Edit / Delete recipes

(Note: Wireframes or screenshots can be attached here after development)

12. Known Issues

- AI-based personalized recipe recommendation not fully optimized
- Limited offline support

13. Future Enhancements

- AI-based meal planner (weekly schedule)
- Voice-enabled cooking assistant
- Nutrition tracker and calorie calculator
- Mobile app version