Project Documentation

# 1. Introduction

• Project Title: Cookbook – Your Virtual Kitchen Assistant

• Team ID: NM2025TMID48212

• Team Leader: PRIYADHARSHINI M (dharshnip659@gmail.com)

• Team Members:

– SOWNDHARYA E (sowndharya6465@gmail.com)

– SATHIYA M (singkongsine90@gmail.com)

– GOMATHI M (indhumathi250505@gmail.com)

# 2. Project Overview

• Purpose: Cookbook is a virtual kitchen assistant that helps users explore, manage, and cook recipes with ease. It provides personalized recipe recommendations, ingredient management, nutrition tracking, and a virtual assistant for cooking guidance.

• Features:

– Recipe search and recommendations  
 – Ingredient-based suggestions  
 – Grocery list generation  
 – Nutrition analysis  
 – Virtual cooking assistant (chatbot/AI guidance)  
 – User authentication and profiles

# 3. Architecture

• Frontend: React.js (with Material UI / Bootstrap for styling)  
• Backend: Python (Flask/Django) or Node.js  
• Database: MySQL / MongoDB  
• APIs: RESTful APIs for recipe search, authentication, and nutrition info

# 4. Setup Instructions

• Prerequisites: Node.js, Python/Flask (or Django), Database (MySQL/MongoDB), Git, VS Code

• Installation Steps:

# Clone repository  
 git clone <your\_repo\_link>  
  
 # Frontend setup  
 cd client  
 npm install  
 npm start  
  
 # Backend setup  
 cd server  
 pip install -r requirements.txt  
 python app.py # or flask run

# 5. Folder Structure

Cookbook/  
|-- client/ # React frontend  
| |-- components/  
| |-- pages/  
|-- server/ # Backend (Flask/Django/Node.js)  
| |-- routes/  
| |-- models/  
| |-- controllers/  
|-- database/ # SQL or MongoDB schemas

# 6. Running the Application

• Frontend:  
 cd client  
 npm start  
  
• Backend:  
 cd server  
 python app.py # or flask run  
  
• Access: Visit http://localhost:3000

# 7. API Documentation

• User:  
 – POST /api/user/register  
 – POST /api/user/login  
• Recipes:  
 – GET /api/recipes/search  
 – GET /api/recipes/:id  
• Grocery & Ingredients:  
 – POST /api/grocery/add  
 – GET /api/grocery/list

# 8. Authentication

• JWT-based authentication for secure login  
• Middleware to protect user-specific routes

# 9. User Interface

• Landing Page  
• Recipe Search Page  
• Nutrition Dashboard  
• Grocery List Page  
• Virtual Assistant Chatbot  
• Admin Panel

# 10. Testing

• Manual testing during milestones  
• Tools: Postman, Chrome DevTools

# 11. Screenshots / Demo

(Add your app screenshots here)

# 12. Known Issues

• Limited recipe dataset  
• Ingredient substitution not fully optimized

# 13. Future Enhancements

• Voice assistant integration  
• Multi-language recipe support  
• Smart kitchen IoT integration  
• AI-based meal planning