

## 1.Introduction

- Project title:  
Cookbook: your virtual kitchen assistant.
- Team ID:  
NM2025TM1D42251
- Team Leader:  
MANISHA. R  
(manisha8610700@gmail.com)
- Team members:  
MONIKA.G  
(karthigokul515@gmail.com)
- MYTHILI.J  
([jmythili2007@gmail.com](mailto:jmythili2007@gmail.com))
- NANDHINI. S  
(rajeshwarisakthi7@gmail.com)

## 2. Project Overview

- Purpose:

To create a user - friendly , Digital platform that helps users discover,organise and image receipts while offering hands- free, interactive guidance to simply meal preparation

Features:

- 1.Recipe Discovery & Management
2. Hands-Free Cooking Assistance
3. Personalized Experience
4. Step-by-Step Guidance
5. Technical Innovation
6. Enhanced User Experience

## 3. Architecture

Frontend:

To display recipes, allow users to search, save, and interact with features like ingredient lists or cooking instructions.

Backend:

The backend handles the "behind-the-scenes" operations of the application.

## Database:

The database stores all the application's data, such as recipes, users, and their preferences.

## 4. Setup instructions

- Prerequisites.

front-end web technologies like HTML, CSS, and JavaScript for the user interface, a suitable Integrated Development Environment (IDE) like VS Code, and a backend framework like React or Python (if using a backend like Flask) for handling logic and data.

- Installation steps:

To install a "Cookbook" for a "Virtual Kitchen Assistant" project, you likely need to install software like Chef Workstation for Chef-based infrastructure automation and its associated tools like Test Kitchen

## 5. Folder structure:

The structure simple but scalable to accommodate project growth.

## 6. Running the application.

- Frontend:

set up a server with its backend and a client with its frontend, connect them via an API, and access the frontend URL in your web browser to view and interact with the application

- Backend:

Creating the server- side logic, database, and that power a cookbook your virtual kitchen assistant

- Access:

Open your web browser and enter the URL of the deployed frontend or the local development URL.

## 7. API Documentation:

- User:

This often includes examples for authentication, creating data, retrieving data and managing user information

- Chats:

JWT - based authentication for secure login

- Middleware protects private routes:

Redirect the user to a login page or send an “access denied” parts of the application

## 9. User interface

- Landing page:  
This page introduces the project, highlighting key features and a clear value position
- Freelancer Dashboard:  
A user interface for freelancer to manage clients, tasks and finances

## 10. Screenshots or Demo:



## 12. Known Issues:

Data resets on page refresh (no persistent storage)  
No authentication system implemented

## 13. Future Enhancements:

Connect backend with [Node.js](#)+Express+mongodb.  
Implement user login & authentication  
Add checkout and payment system  
Create sales tracking & reports

