**Project Documentation – CookBook: Virtual Kitchen Assistant**

**1. Introduction**

* **Project Title**: CookBook – Virtual Kitchen Assistant
* **Team ID**: NAM2025TMID32306
* **Team Leader**: Kanchana c
* **Team Members**: Kamalasri S

Ishwarya B

Inthumathi M

**2. Project Overview**

* **Purpose**:  
  CookBook is a virtual kitchen assistant application that helps users manage cooking by providing guided recipes, ingredient management, and a simple interface.
* **Key Features**:
  + Recipe browsing & guided cooking flow
  + Ingredient list management
  + Real-time interaction with the app
  + Admin control panel for managing content

**3. Architecture**

* **Frontend**: React.js with Bootstrap and Material UI
* **Backend**: Node.js & Express.js (server logic + API endpoints)
* **Database**: MongoDB (stores user data, recipes, ingredients, chat/interaction history)

frontend\_Sample

**4. Setup Instructions**

**Pre-requisites**

CookBook\_Virtual\_Kitchen\_Assist…

:

* Node.js (LTS version)
* MongoDB
* Git
* Visual Studio Code

**Steps**:

1. Download project code from SmartInternz Portal → Code Drive Link
2. Extract the ZIP folder
3. Open folder in VS Code
4. Install dependencies:

bash

Copy code

npm install

1. Start the project:

bash

Copy code

npm start

1. Open the app in browser → [**http://localhost:3000**](http://localhost:3000/)

**5. Folder Structure**

bash

Copy code

CookBook/

│-- client/ # React frontend

│ └── components/

│ └── pages/

│-- server/ # Node.js backend

│ └── routes/

│ └── models/

│ └── controllers/

**6. Running the Application**

* **Frontend**:

bash

Copy code

cd client

npm start

* **Backend**:

bash

Copy code

cd server

npm start

* **Access**: [http://localhost:3000](http://localhost:3000/)

**7. API Documentation**

* **User**:
  + /api/user/register
  + /api/user/login
* **Projects/Recipes**:
  + /api/projects/create
  + /api/projects/:id
* **Applications**: /api/apply
* **Chats/Assistant**:
  + /api/chat/send
  + /api/chat/:userId

frontend\_Sample

**8. Authentication**

* **JWT-based authentication** for secure login
* Middleware protects private routes

**9. User Interface**

* Landing Page
* User Dashboard (recipes, assistant)
* Admin Panel
* Recipe/Project Details Page

**10. Testing**

* Manual testing at different milestones
* Tools used: **Postman**, **Chrome Dev Tools**

**11. Known Issues**

* (To be filled during development/testing phase)

**12. Future Enhancements**

* Integration with voice assistant for hands-free cooking
* Adding more cuisines & advanced recipe suggestions
* Nutrition tracking & shopping list generator

.