**Project Documentation**

**1. Introduction**

Project Title: CookBook – Your Virtual Kitchen Assistant

Team ID: NM2025TMID36470

Team Leader: [ SONA S & [24cssonas95@gmail.com](mailto:24cssonas95@gmail.com) ]

Team Members:

[ MONISHRI V & [24csmonishri059@gmail.com](mailto:24csmonishri059@gmail.com) ]

[ SANTHOSHINI S & [24cssanthoshini087@gmail.com](mailto:24cssanthoshini087@gmail.com) ]

[ NATHIYA DEVI S & [24csnathiyadevi060@gmail.com](mailto:24csnathiyadevi060@gmail.com) ]

**2. Project Overview**

**Purpose:** CookBook is an intelligent digital recipe and kitchen assistant that helps users discover, manage, and cook recipes. It simplifies meal planning, suggests ingredients, and guides cooking with step-by-step instructions.

**Features:**

* Recipe search and recommendation
* Ingredient-based recipe suggestions
* Step-by-step cooking guidance (voice/text)
* Meal planning and shopping list generation
* User authentication and personal recipe storage
* Admin dashboard for recipe management

**3. Architecture**

**Frontend:** React.js with Tailwind CSS / Material UI

**Backend:** Node.js & Express.js for APIs and business logic

**Database:** MongoDB for recipes, user data, preferences

**Optional Integration:** External APIs for nutrition or recipe search

**4. Setup Instructions**

**Prerequisites:**

* Node.js
* MongoDB
* Git
* React.js
* Express.js, Mongoose
* Visual Studio Code

**Installation Steps:**

# Clone repository

git clone <repo-link>

# 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/

|-- database/ # MongoDB setup scripts

**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

POST /api/recipes/create

GET /api/recipes/:id

**Shopping List:**

POST /api/shopping-list

GET /api/shopping-list/:userId

**8. Authentication**

* JWT-based authentication for secure login
* Middleware to protect user dashboards and recipe management

**9. User Interface**

* Landing Page with recipe highlights
* Recipe search and filter page
* Cooking assistant interface (step-by-step guide)
* Meal planner and shopping list page
* Admin Panel for managing recipes

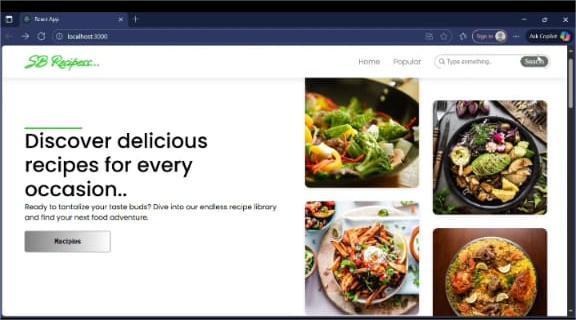
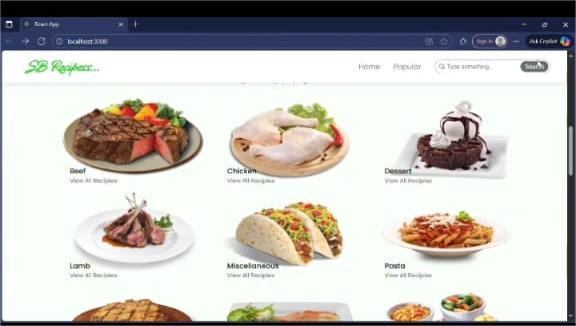
**10. Testing**

**Manual Testing:** During each milestone

**Tools:** Postman, Chrome Dev Tools, MongoDB Compass

**11. Screenshots or Demo**

****

** **

**12. Known Issues**

* Limited offline support
* Dependency on internet for external API recipes

**13. Future Enhancements**

* Voice-enabled cooking instructions
* AI-powered personalized meal plans
* Integration with smart kitchen devices