**Project Documentation**

**1. Introduction**

* **Project Title**: Cook Book – Your Virtual Kitchen Assistant
* **Team ID**: NM2025TMID42629
* **Team Leader**:
  + M. Komathi – [komathimadhappan@gmail.com](mailto:komathimadhappan@gmail.com)
* **Team Members**:
  + D. Gopika – [gopika535212@gmail.com](mailto:gopika535212@gmail.com)
  + N. Kalpana – [kalpananarayanan588@gmail.com](mailto:kalpananarayanan588@gmail.com)
  + M. Priyanka – [priyankamurugan1710@gmail.com](mailto:priyankamurugan1710@gmail.com)

**2. Project Overview**

* **Purpose**:  
  Cook Book is a virtual kitchen assistant that helps users discover, organize, and manage recipes. It offers step-by-step cooking guidance, ingredient management, and personalized suggestions to simplify the cooking experience.
* **Features**:
  + Recipe search and categorization
  + Step-by-step cooking assistant
  + Ingredient checklist and shopping list
  + Personalized recommendations
  + Admin panel for recipe management

**3. Architecture**

* **Frontend**: React.js with Bootstrap and Material UI
* **Backend**: Node.js and Express.js
* **Database**: MongoDB (recipes, ingredients, user preferences)

**4. Setup Instructions**

**Prerequisites**

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

**Installation Steps**

# Clone the repository

git clone <repository-url>

# 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 Application**: [http://localhost:3000](http://localhost:3000/)

**7. API Documentation**

* **User APIs**:
  + POST /api/user/register
  + POST /api/user/login
* **Recipe APIs**:
  + POST /api/recipes/create
  + GET /api/recipes/:id
  + GET /api/recipes/search
* **Shopping List APIs**:
  + POST /api/shoppinglist/add
  + GET /api/shoppinglist/:userId

**8. Authentication**

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

**9. User Interface**

* Landing Page
* Recipe Dashboard
* Cooking Assistant Page
* Shopping List Manager
* Admin Panel

**10. Testing**

* **Approach**: Manual testing during milestones
* **Tools**: Postman, Chrome Dev Tools

**11. Screenshots or Demo**

[Add screenshots of recipe search, cooking assistant, and shopping list features]

**12. Known Issues**

[List any existing bugs or limitations]

**13. Future Enhancements**

* Voice-enabled cooking assistant
* AI-powered meal planner
* Mobile app integration