Project Documentation

Project Title 1.

Introduction Project Title: Cook Book: Your Virtual Kitchen Assistant

Team ID: NM2025TMID42339

Team Leader:

VANISRI.K& vanisrikumar2312@gmail.com

Team Members:

VIJAYAKUMARI.C& chakkrai1156@gmail.com

SWETHA.V& swethavswa@gamil.com

2. Project Overview

•Purpose:

To compile recipes, often with a specific theme, for an intended audience. Purpose vary widely, from sharing family recipes and documenting cultural heritage to teaching cooking skills, promoting a business, or even generating income.

• Features:

Span content, design, and user experience, including a central theme, well-organized recipes with clear instructions, engaging visuals like high-quality photos and illustrations, a user-friendly layout with an intuitive table of contents and index, and elements that enhance usability such as meal planning tools, search functionality, and interactive feedback systems, as seen in digital cookbook applications.

3. Architecture

• Frontend:

Consider an architecture using a JavaScript framework like React or Angular for building the user interface.

• Backend:

A Node.js server with an Express.js framework, communicating with a frontend (e.g., React) via RESTful APIs, and a database (e.g., PostgreSQL, MongoDB, or a cloud-based solution like Firebase) for storing user data, recipes, and ingredients.

• Database:

A cookbook project's database architecture typically uses a relational database model like (MySQL) to manage users, recipes, and ingredients efficiently. MongoDB stores user data, project information, applications, and chat messages

4. Setup Instructions

• Prerequisites:

To start a cookbook project, define its concept and target audience, then gather and test recipes to ensure they are accurate. You must then organize the recipes, write introductions, create a consistent design with high-quality photos, and thoroughly proofread. Finally, decide on a publishing option and begin promoting the cookbook.

• Installation Steps:

Git: A version control system for cloning the project repository.

Node.js and npm: A JavaScript runtime and package manager needed to install the project's dependencies. 5. Folder Structure For a full-stack recipe book project, the folder structure you provided is a strong starting point that clearly separates the frontend and backend. To make the structure more robust and scalable, you can add and organize subfolders within your client and server directories following standard best practices for MERN-stack applications.

5. Folder Structure

For a full-stack recipe book project, the folder structure you provided is a strong starting point that clearly separates the frontend and backend. To make the structure more robust and scalable, you can add and organize subfolders within your client and server directories following standard best practices for MERN-stack applications.

6. Running the Application

• Frontend:

*To run the frontend for a "Cook Book" project, follow these steps in your terminal: * Navigate to the client directory. The cd client command changes your current location in the terminal to the client folder.

*Start the development server. The npm start command runs the script that starts the application, typically a development server.

*This is a common procedure for projects with a client-server structure, where the frontend is in a client or frontend folder and the backend is in a separate directory.

• Backend:

1.Open your terminal or command prompt.

2.Navigate to the project's backend directory. Based on common project structures, this is likely a folder named server or backend inside your main project folder.

3.Install the necessary dependencies. If this is your first time setting up the project, or if new dependencies have been added, run the npm install command to download all required packages.

4.Start the server using the npm start command. This will execute the script defined as "start" in your backend's package.json file.

5.If npm start does not work, it is possible the start script is named differently. Look inside the package.json file in the server directory for the "scripts" section.

• **Access:** Visit http://localhost:3000 7. API Documentation

• **User:** A standard cookbook project would use RESTful API endpoints for user authentication. The /api/user/register endpoint would handle new user sign-ups, while /api/user/login would authenticate existing users.

• **Projects:** This API documentation describes the resources for managing projects within the Cookbook application. The API uses a RESTful architecture, standard HTTP methods, and JSON for request and response payloads.

• **Applications:**

* The user's query is about the API documentation for a "cookbook project", specifically the /api/apply endpoint. However, "cookbook project" is a generic term used in many different contexts and does not refer to a single, specific project. Based on the search results, "cookbook" could refer to any of the following, and the specific /api/apply endpoint details would be unique to each:

*A recipe management system or application where /api/apply could be a placeholder for a resource like a recipe.

* The OpenAI Cookbook, which provides examples and guides for using the OpenAI API.An Android Enterprise dedicated devices project.

*The Chef automation platform, which uses "cookbooks" to configure servers.A specific GitHub project using the "cookbook" name.

*A proprietary application like the Alemba API or Trend Micro's Workload Security, which have their own "cookbooks".

* To provide accurate API documentation for the /api/apply endpoint, more specific information is needed about which "cookbook project" the user is referring to.

• **Chats:**

The requested information is an API documentation outline for a "cookbook project" with chat features, based on the provided endpoints /api/chat/send and /api/chat/:userId. The endpoints suggest a RESTful API structure, where /api/chat/send is likely for sending a new message, and /api/chat/:userId is for retrieving chat messages for a specific user.

8. Authentication:

Implementing authentication for your cookbook project requires a JWT-based system that uses middleware to protect private routes. The following steps outline the process for a typical server-side implementation using a framework like Express.js.

9. User Interface:

- **Landing Page:-** The landing page should grab attention quickly, communicate the app's value, and guide users to the next step, whether they are a new user, a potential freelancer, or a returning admin.

- **Freelancer Dashboard:-** This personalized workspace should allow freelance recipe developers and chefs to manage their work efficiently.

- **Admin Panel:-** The admin panel is for managing the platform itself and requires a clean, efficient interface for monitoring users, freelancers, and project activity.

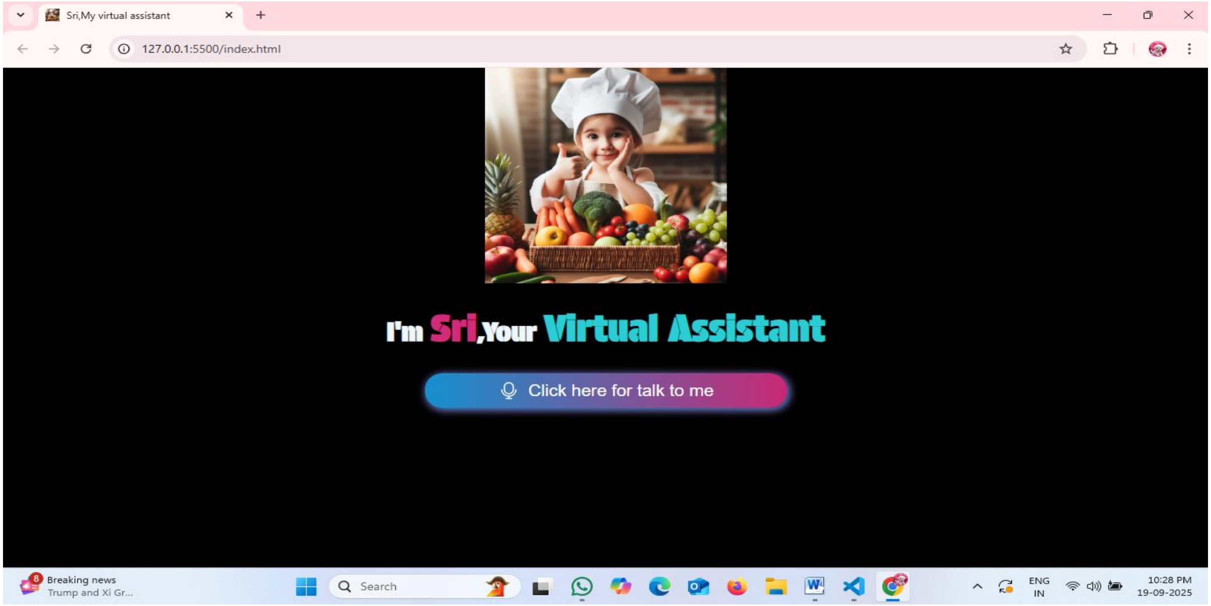
- **Project Details Page:-** This page serves as a hub for all information related to a single project, providing a clear overview and space for communication.

10. Testing:

- **Manual testing during milestones:-** A cookbook project typically progresses through several key stages. Manual testing should be integrated at each of these milestones to ensure quality at every step. °Milestone 1: Recipe API development and integration. °Milestone 2: Frontend user interface (UI) development. °Milestone 3: User account and contribution features. °Milestone 4: Search, filter, and other advanced features. °Milestone 5: Final pre-release testing.

- **Tools:** Setting up Postman °Create a collection: Organize all your cookbook API requests into a single Postman collection. °Use environments: Create environments for your different servers to manage variables like base URLs and API keys. This prevents you from hardcoding values in your requests.

11. Screenshots:-


Sri,My virtual assistant
127.0.0.1:5500/index.html
I'm Sri,Your Virtual Assistant
Click here for talk to me
Breaking news
Trump and Xi Gr...
Search
ENG
IN
10:28 PM
19-09-2025