**Project Documentation**

**Cook book : your virtual kitchen assistant**

# 1. Introduction :

* **Project Title :** Cook Book your virtual assistant
* **Team ID** : NM2025TMID29894
* **Team Leader :** SARANYA S
* **LEADER MAIL ID :**  [202400954@sigc.edu](mailto:202400954@sigc.edu)
* **Team Members:**

SARU LATHA A – [202400689@sigc.edu](mailto:202400689@sigc.edu)

RINOSHLINE A – [202400108@sigc.edu](mailto:202400108@sigc.edu)

PREETHI N – [202400892@sigc.edu](mailto:202400892@sigc.edu)

# 2. Project Overview :

# A cookbook project involves conceptualizing, writing, and presenting a collection of recipes, with potential goals including skill improvement, community history, and sharing culinary traditions.

**1.Concept & Planning:**

* + **Choose Your Concept:** Decide on the focus of your cookbook, such as a specific cuisine, dietary need, or skill level.
  + **Plan the Structure:** Outline how the recipes will be organized, including potential sections and a table of contents.
  + **Create a Proposal:** A written document detailing the cookbook's concept, target audience, and unique selling points.

**2. Content Creation:**

* + **Write Recipes:**Develop clear, step-by-step instructions for each dish.
  + **Test Recipes:**Ensure recipes are accurate and reliable by testing them thoroughly**.**

**3.Design & Presentation:**

* + **Design the Layout:**Decide on the visual aesthetic for the cookbook's cover, dividers, and individual pages.
  + **Edit Text:**Refine the language, ensuring consistency and clarity**.**
  + **Proofread & Index**: Carefully check for any errors and create an index to help readers find specific recipes.

**3. Architecture**:

* **Client layer (Frontend):**

This is the user-facing part of the application, responsible for the user interface and user experience.

* **Web application:** A single-page application built with a framework like React, Vue.js, or Angular.
* **Mobile application:** Native (Swift for iOS, Kotlin for Android) or cross-platform (React Native, Flutter) for iOS and Android devices.
* **API layer (Backend):**

The backend provides the business logic and serves as a communication hub between the client and the database. It can be built using a microservices or monolithic approach.

* **User service:** Manages user registration, authentication, and profiles.
* **Recipe service:** Handles creating, editing, and retrieving recipes.
* **Search service:** Provides advanced search and filtering capabilities.
* **Community service:** Manages user ratings, reviews, and social interactions.

# 4. Setup Instructions :

* **Prerequisites:**

Before you begin, ensure that the following tools and services are installed and set up:

* + Node.js : It is required to run the backend server and manage dependencies.

\* **Installation:**

* + Download and install Node.js from here.

Verify installation:

* + node -v

npm -v

 **MongoDB :**

It is used to store recipe data, user profiles, and other application data.

 **Installation:**

* Install MongoDB by following the official MongoDB setup guide.
* Alternatively, you can use a cloud service like [MongoDB Atlas](https://www.mongodb.com/cloud/atlas).
* After installation, start the MongoDB service:

sudo systemctl start mongodb

sudo systemctl enable mongodb

 Git is necessary for version control and managing your source code.

 **Installation:**

* Download Git from [here](https://git-scm.com/).
* Verify installation:
  + git --version
  + React.js

 React.js is used to build the frontend of your cookbook application.

 **Installation:**

* If you don't have create-react-app installed, use the following:
  + npm install -g create-react-app
  + Express.js **–** Mongoose **–** Visual Studio Code

npm install -g create-react-app

-Create a new React project (or clone your existing repo):

create-react-app cookbook-app

Install necessary dependencies by navigating to the project directory:

cd cookbook-app

npm install

# 6. Running the Application :

**Frontend:**

1.Navigate to the client directory:

cd client

2. Start the React frontend:

npm start

**Backend:**

1.Navigate to the server directory:

cd server

2.Start the Node.js backend:

npm start

**Access the Application:**

1.Once both the frontend and backend are running, open your browser and go to:

http://localhost:3000

# 7. API Documentation

**User**

1. **POST** /api/user/register

**Projects**

1. **POST** /api/projects/create
2. **GET** /api/projects/:id

**Applications**

1. **POST** /api/apply

**Chats**

1. **POST** /api/chat/send
2. **GET** /api/chat/:userId

# 8.AUTHENTICATION:

# JWT-based Authentication

* **Description:** The application uses JSON Web Tokens (JWT) to secure login and protect private routes.
* **User Login:** When a user logs in, the server validates their credentials.
* **JWT Generation:** Upon successful login, a JWT token is generated and sent to the client.

# 9. User Interface :

* **Key Components:**
  1. **Navigation Bar**:
     + Logo
     + Links to Sign Up, Login, About, and Contact
  2. **Hero Section**:
     + Eye-catching banner with a call-to-action (CTA) like "Start your freelance journey" or "Find your next project."
  3. **Features Section**:
     + Short descriptions of what the platform offers (e.g., "Post Projects", "Apply for Jobs", "Secure Payments").
  4. **How It Works**:
     + A brief explanation of the platform's process (posting a job → applying for a job → working with clients).
  5. **Testimonials/Reviews** (Optional):
     + Quotes or ratings from happy freelancers or clients, building trust and credibility.
  6. **Footer**:
     + Contact info, links to privacy policy, terms of service, and social media.
* **Freelancer Dashboard:**
  1. **Profile Overview**:
     + Display user info like name, profile picture, rating, and short bio.
  2. **My Projects**:
     + List of projects the freelancer has posted or is currently working on, with status updates (e.g., "In Progress", "Completed", etc.).
     + Option to create a new project or edit existing ones.
  3. **Applications**:
     + View applied projects and track the status of applications (Accepted, Pending, Rejected).
     + A button to apply for new projects.
  4. **Messages/Chat**:
     + Real-time chat functionality with clients and other freelancers.
     + Display recent conversations with clients and quick access to ongoing chats.
  5. **Notifications**:
     + Display alerts or updates regarding new project invitations, application status changes, or new messages.
  6. **Earnings and Stats**:
     + Overview of earnings, including paid projects, current balance, and payment history.
  7. **Settings**:
     + Profile settings, notification preferences, account management, etc.

Bottom of Form

# Testing :

**1. Unit Testing**

**Purpose**: Test individual components and backend functions.

**Tools**: Jest, React Testing Library (Frontend), Mocha/Chai (Backend)

**Example**: Test that recipe details render correctly on the UI.

**2. Integration Testing**

**Purpose**: Test how frontend and backend work together.

**Tools**: Cypress (Frontend), Super test (Backend)

**Example**: Test that submitting a recipe form stores it in the database and displays on the UI.

**3. API Testing**

**Purpose**: Test API endpoints for CRUD operations.

**Tools**: Super test, Postman

**Example**: Ensure POST /api/recipes/create creates a new recipe.

# Run Tests:

**Frontend**: npm test

**Backend**: npm test

**E2E (Cypress)**: npx cypress open

1. **Known Issues**
2. **Search Functionality**:
   * **Issue**: Search results may not always be accurate due to partial matches in the recipe database.
   * **Solution**: We are improving the search algorithm for better accuracy.
3. **Responsive Layout**:
   * **Issue**: Some mobile devices might not display the recipe card grid properly on smaller screens.
   * **Solution**: UI improvements are in progress to ensure a consistent layout across all devices.

# 12.Future Enhancements

* **Advanced Search Filters**:

Add filters for cuisine, diet, and cooking time.

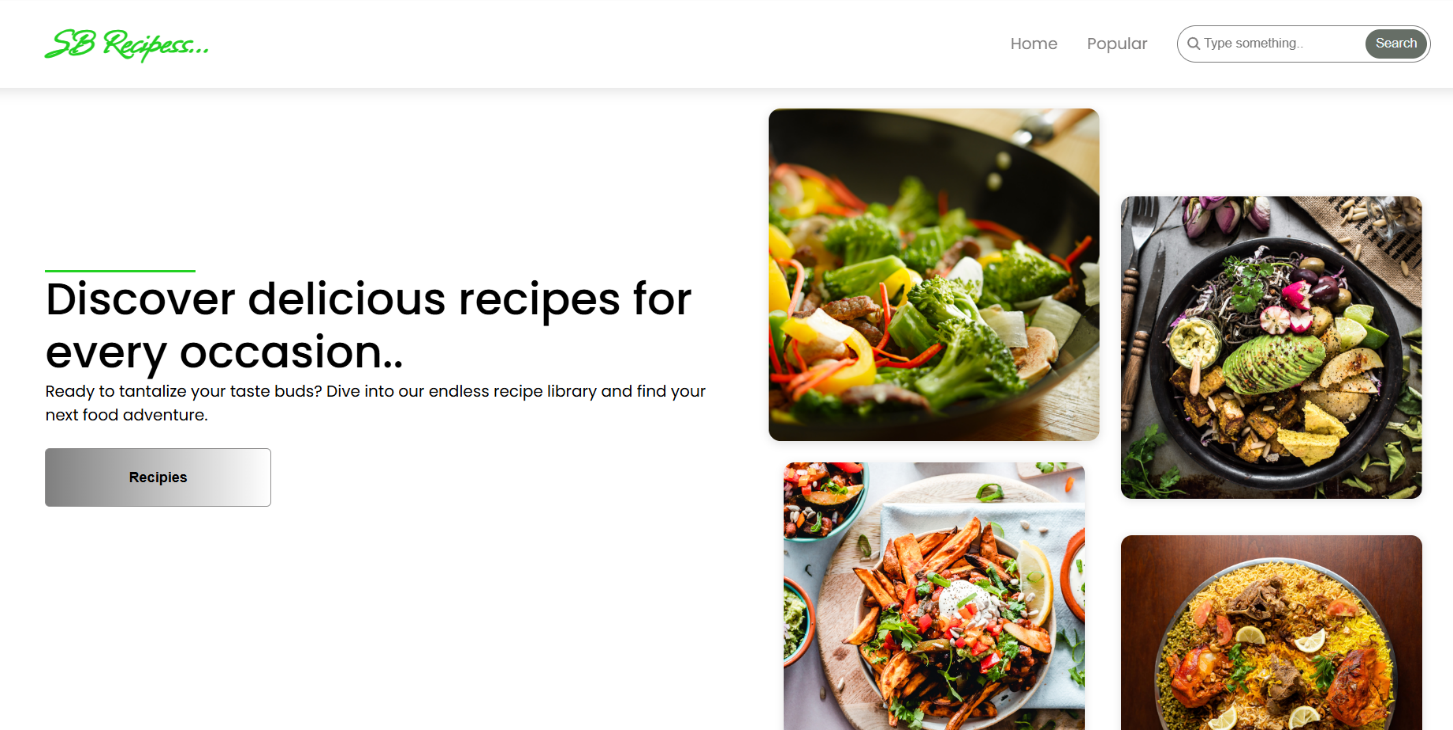
* **User Ratings & Reviews**:

Allow users to rate and review recipes.

* **Recipe Categories**:

Organize recipes by categories (e.g., Desserts, Quick Meals).

# 13.ScreenShot



# 14.Demo video link

<https://drive.google.com/file/d/1Kwfxot5BWq7VB30rzea14qHAXxEUzv34/view?usp=drive_link>