Project Document

COOK BOOK

1. Introduction

* Cook book
* Team ID : NM2025TMID42615
* Team Leader : P.Kowsalya & [kowsalyap1202@gmail.com](mailto:kowsalyap1202@gmail.com)
* Team Members :
* K.Mugileswari & [mugileshwarikrishanan@gmail.com](mailto:mugileshwarikrishanan@gmail.com)
* I.Divya & [divyailayaraja@gmail.com](mailto:divyailayaraja@gmail.com)
* M.Durgadevi & [dd0226306@gmail.com](mailto:dd0226306@gmail.com)
* D.Thiruvasagi & [d.sarasu021@gmail.com](mailto:d.sarasu021@gmail.com)

2. Project Overview

* Purpose : Recipe collection , culinary guidance , inspiration and education.
* Features :
* Recipes with ingredient lists and instructions.
* Photos and authentication and authorization.
* Cooking techniques and tips
* Meal planning and organization ideas

3. Architecture

* Frontend : Recipes display and interaction
* Backend : User authentication and authorization
* Database : Data storage and retrieval for recipes and users

4. Setup Instructions

* Prerequisities
* Node.js
* Mongo DB
* Github
* React.js
* Express.js – Mongoose- visual studio code
* Installation steps :

#Clone the repository git clone

#Install client dependence cd

Client npm install

#install server dependencies cd

../server npm install

5. Folder structure

SB-works/

|--client/ # React frontend

|\_\_components/

L\_\_pages/

|\_\_server/ # Node.js backend

|\_\_routes/

|\_\_models/

|\_\_controllers/

6. Running the Application

1. Frontend

cd client

* npm start
* Run npm start or yarn start the frontend development server
* Access the application at <http://localhost:3000(or> specified port)

1. Backend

* Run npm install or yarn install to install dependencies
* Run node server.js or npm start the backend server
* Verify API endpoint using Postman or cURL

Access : Visit http://localhost:3000

7. API Documentation

1. User

* GET/users- List users
* GET/users/{id}-User details
* POST/users-Create user
* PUT/users/{id}-Update user
* DELETE/users/{id}-Delete user

2. Project

* GET/projects-List projects
* GET/projects/{id}-Project details
* POST/projects-Create project
* PUT/projects/{id}-Update project
* DELETE/projects/{id}-Delete project

3. Applications

* GET/application- List apps
* GET/application/{id}-App details
* POST/application-Create app
* PUT/application/{id}-Update app
* DELETE/application- Delete app

4. Chats

* GET/chats-List chats
* GET/chats/{id}-Chats details
* POST/chats-Start chat
* GET/chats/{id}/messages-List messages
* POST/chats/{id}-Delete chat

8. Authentication

1. JWL-Based Authentication Secure Login

User Login

* User provides **username/email + password**
* Server verifies credentials
* If valid- generate **JWL (JSON Web Token)**

Token Structure

* Header (algorithm + type)
* Payload (user info, expiration time)
* Signature(secret key verification)

Response

* Server sends JWL to client
* Client stores it in **localStorage/sessionStorage** (or cookies)

Future Requests

* Client attaches JWL in **Authorization header**-bearer <token>
* Server validates token before processing the request

2. Middleware for Private Routes

* Middleware checks every incoming request

1. Extract JWL from Authorization header
2. Verify with server’s secret key
3. If valid- allow request to proceed
4. If invalid/expired- block with **401 Unauthorized**

9. User Interface

1. Landing Page

1. Heade /Navbar : Logo, Home, Explore Recipes, About, Login/Signup
2. Hero Section : Large banner with tagline(e.g,. “Discover, Share and Manage Your Recipes & Projects”)
3. Features Section : Cards with icons for: Recipe Sharing, Freelancer Collaboration, Admin Oversight
4. Popular Projects / Recipes : Grid of top trending items with images, short description, and “View Details” button
5. Footer : Quick links, social media, contact info

2. Freelancer Dashboard

* Sidebar Menu

a) Dashboard Overview

b) My Projects

c) Application / Proposals

d) Messages / Chats

e) Settings

* Main Area

a) Welcome message with profile picture

b) Statistics

\* Ongoing Projects

\* Completed Projects

\* Earnings (if applicable)

* Projects List

a) Cards / tables view with project title, status, deadlines, and action buttons (*View / Submit Work*)

* Notifications

a) Alerts about project updates, new messages, or deadlines

3. Admin Panal

* Sidebar Menu

1. Dashboard
2. Manage Users (Freelancers / Clients)
3. Manage Projects
4. Manage Applications
5. Reports / Analytics
6. Settings

* Dashboard Widgets

1. Total Users
2. Active Projects
3. Pending Approvals
4. Revenue / Usage stats

* User Management
* Table of users with search, filter, edit, and block options
* Project Oversights

a) Approve / Reject new project

b) Monitor progress and disputes

4. Projects Details Page

* Project Header
  + - * 1. Title of projects / Recipes
        2. Posted by (Client / Freelancer profile link)
        3. Date posted, category, tags
* Main Content
  + - * 1. Description (steps, ingredients, or project details)
        2. Attachments / Files (images, documents)

10. Testing

1. Manual Testing During Milestones
   * + - 1. Perform functional testing at each development milestones
         2. Validate core features like user authentication, project creation, chat functionality, and admin workflows
         3. Use test cases and checklists to ensure feature coverage
         4. Conduct regression testing whenever new features are added
2. Tools: Postman, Chrome Dev Tools
   * 1. Postman- For API testing (checking endpoints for users, projects, applications, and chats )
     2. Chrome Dev Tools- For UI debugging, performance, monitoring, and network inspection

11. Screenshots or Demo



12. Feature Enhancement

* + - * + Authentication & Security

JWL token expiry handling not implemented- users may get logged out abruptly

No refresh token mechanism- frequent re-login required

Weak password rules or no encryption for sensitive data

* + - * + User Interface (UI/UX)

Dashboard not responsive on mobile/tablet screens

Slow loading of recipes/project images due to unoptimized media

* + - * + Functionality

Search/filter in cookbook may return irrelevant or incomplete results

Admin panel updates not reflecting immediately

Notifications/messages not showing real time

* + - * + Data & Backend

**Cloud storage integration** for media files

**Pagination & lazy loading** to handle large recipe database

**API rate limiting** for better scalability

**Analytics dashboard** for admin (top recipes, most active freelancers)

* + - * + User Roles & Community

User reviews & rating for recipes/project

Collaborative recipe/project creation

Gamification features

Community formus or Q&A section

* + - * + Performance & AI Integration

**Image recognition AI:** auto-detect ingredients from uploaded photos

**Chatbot /assistant** to help users discover recipes/projects

* + - * + Testing & Documentation

Expand **automated test coverage** (unit + integration + UI tests)

**API documentation portal** with examples (Swagger / Postman collections)

Regular **usability testing** with end users

13. Known Issues

* + - * Data & Backend

Duplicate entries if the same recipe/project is submitted twice

File upload limits not validated(large recipe images crash server

* + - * User Roles & Permissions

Freelancers accessing admin-only pages due to weak route protection

Role-based permissions not strictly enforced in middleware

* + - * Performance

Slow queries when fetching all recipes/projects

High memory usage on the dashboard with many active projects

* + - * Testing Gaps

Limited coverage for edge cases

Manual testing done, but automation missing for regression