Cookbook Project Documentation 🍴

**Project Title:** My-Cookbook--

**Team Members:**

T.Vinothini (leader)

K.Shanmadhi

R.Sowmiya

V.Sathya sri

**Introduction:**

This project, \*\*Cookbook\*\*, is a web application designed to help users explore, save, and manage their favorite recipes in an interactive way. It integrates a backend powered by Node.js and MongoDB with a modern frontend built on React.

# Project Overview

Purpose 🎯:  
The purpose of this project is to provide a digital recipe management platform where users can browse, add, and save their favorite meals.  
  
Features ✨:  
- User authentication & secure login  
- Browse recipes and add to favorites  
- Responsive and user-friendly design  
- Backend API for recipe and favorites management  
- Database integration for storing user preferences

# Architecture

The project follows a \*\*MERN (MongoDB, Express, React, Node.js) architecture\*\* 🚀.  
  
- \*\*Frontend (React)\*\*: Provides the UI components, pages, and routing.  
- \*\*Backend (Node.js + Express)\*\*: Handles API requests and business logic.  
- \*\*Database (MongoDB)\*\*: Stores user details, recipes, and favorites.  
  
Component Structure 🧩:  
- Header, Footer, Navigation bar  
- Recipe List Component  
- Favorites Component  
- Authentication Component (Login/Signup)  
  
State Management:  
- Local state for component interactions  
- Global state using Context API  
  
Routing:  
- React Router is used for navigation between pages such as Home, Recipes, Favorites, and Profile.

# Setup Instructions

Prerequisites ⚙️:  
- Node.js (>=16)  
- MongoDB  
- npm or yarn package manager  
  
Installation Steps:  
1. Clone the repository: `git clone [repo link]`  
2. Navigate to backend folder: `cd backend`  
3. Install dependencies: `npm install`  
4. Setup `.env` file with MongoDB URI & secrets  
5. Start backend server: `npm run dev`  
6. Navigate to frontend folder and run: `npm start`

# Folder Structure

📂 Client (Frontend):  
- components/  
- pages/  
- assets/  
- utils/  
  
📂 Backend:  
- server.js  
- models/  
- routes/  
- controllers/  
- config/

# Running the Application

Frontend: `npm start` (runs on http://localhost:3000)  
Backend: `npm run dev` (runs on http://localhost:5000)  
  
Ensure MongoDB is running before starting the servers.

# Component Documentation

Key Components:  
- \*\*RecipeList\*\*: Displays available recipes  
- \*\*Favourite\*\*: Manages user-saved recipes  
- \*\*Auth\*\*: Handles login/signup  
  
Reusable Components:  
- Buttons  
- Cards  
- Modals

# State Management

Global State 🌍:  
- Managed with Context API for auth and favorites  
  
Local State ⚡:  
- Handled via useState in individual components

# User Interface

UI Highlights 🎨:  
- Clean, minimal design  
- Responsive layout for mobile & desktop  
- Intuitive navigation bar  
- Recipe cards with images and titles  
- Favorites page for saved recipes  
  
(Screenshots to be inserted here)

# Styling

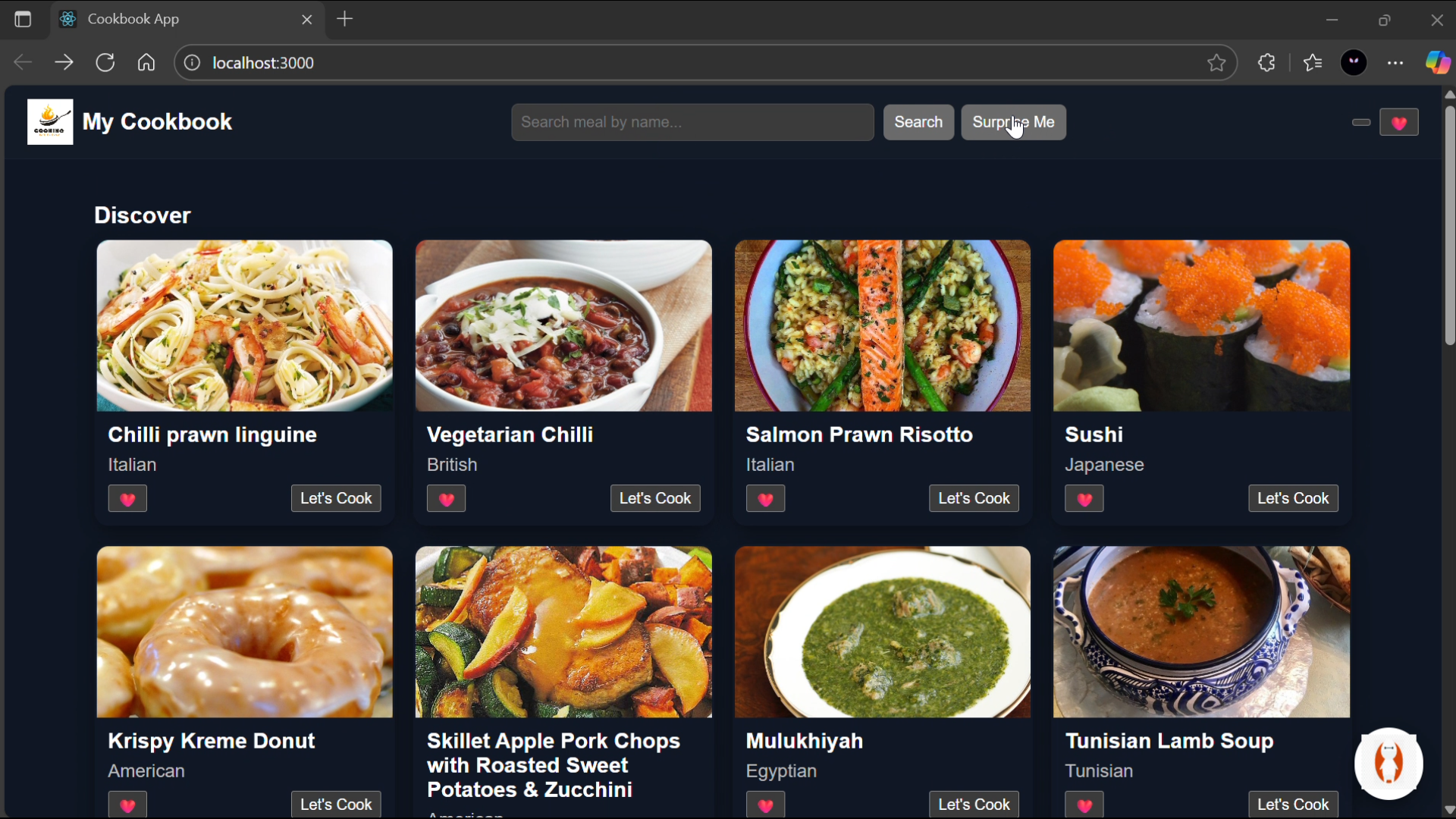
Styling with CSS and Tailwind 💅  
- Responsive grid system  
- Reusable styles for buttons and forms  
- Theming support for light/dark mode (optional)

# Testing

Testing Strategy ✅:  
- Unit testing with Jest  
- Component testing with React Testing Library  
- API testing with Postman  
  
Code Coverage:  
- Ensured through automated test scripts

# Screenshots or Demo

This is the home page view:



# Known Issues

- Limited dataset of recipes  
- No role-based access (admin features missing)  
- Some UI components need optimization for large screens

# Future Enhancements

🚀 Planned Improvements:  
- Add search & filter functionality  
- Implement recipe categories  
- Enable recipe image upload  
- Add admin dashboard for managing recipes  
- Integrate external recipe APIs for more data