# UNIVERSITY OF PETROLEUM AND ENERGY STUDIES, DEHRADUN


## Recipe Sharing Application

*Final Report of the (Minor Project - II) in Semester VI*

*Prepared by:*


| S. No | Students Name | Roll Number | Sap Id |
|---|---|---|---|
| 1. | Akshay Mohpal | R2142210078 | 500088177 |
| 2 | Diya Khandelwal | R2142210293 | 500090939 |
| 3 | Gaurav Bhandari | R2142210311 | 500090993 |
| 4 | Kanishka Singh | R2142210393 | 500091377 |


BACHELORs OF TECHNOLOGY, COMPUTER SCIENCE ENGINEERING

With specialization in DevOps

*Under the guidance of*

**Mr. Sandeep Pratap Singh**


Division of Computer Science (SOCS), UPES

Bidholi Campus, Energy Acres, Dehradun – 248007

# INDEX

# Recipe Sharing App

## Introduction

In a world driven by digital connectivity and a shared love of culinary delights, recipe-sharing apps have arisen as a game-changing tool for foodies to connect, create, and celebrate the art of cooking. As our kitchens become sanctuaries for experimentation and creativity, these applications offer a virtual platform for users to share their favorite recipes, discover new culinary trends, and develop a community centered on a shared love of fine food.

The user experience is an important part of recipe sharing applications. These platforms' overall success is attributed to its seamless navigation, visually appealing interfaces, and ability to seamlessly engage with other users.

This synopsis will delve into the project's core elements, exploring data representation, algorithmic implementation, real-time adaptability, and user interface design.

## Problem Statement

Developing a robust and innovative recipe-sharing application that focuses on community building, userfriendly interfaces, seamless recipe sharing, enhanced social interactions, and user empowerment. The goal is to build a dynamic digital platform that not only acts as a repository for cooking creativity, but also fosters a vibrant global community of food enthusiasts by providing them with the tools and services they need to connect, share, and empower one another on their cooking adventures.

## Objectives

1. **User Authentication with Passport.js:** Implement user registration and login using Passport.js authentication. Use JWTs (JSON Web Tokens) to secure authentication and manage user sessions. Ensure that Passport tactics are properly integrated (local strategy, social media logins) to create a flexible authentication system.

2. **Favorite List and User Profile**: Create a feature that allows users to add recipes to their favorites list. Provide the ability to view and manage their favorite recipes. Users can establish and customize their profiles, including photographs, bios, and posted recipes.

3. **Recipe Management and Search Functions:** Allow users to create, amend, and remove recipes using a sophisticated text editor. Implement a search function that allows users to look for recipes using keywords, ingredients, or categories. Include filters for selecting and categorizing recipes according to various parameters. Improve search performance by integrating a search engine such as Elastic search or using MongoDB's text search capabilities.

## Problems and Challenges in the model:

- **Community Engagement**- Difficulty in attracting and maintaining an active user base, hindering the creation of a thriving community.
  Challenge: Developing strategies to continuously engage users, encourage participation, and foster a sense of belonging within the community.

- **User Interface Complexity**- Balancing user empowerment with a user-friendly interface might lead to potential complexities, especially for novice cooks.
  Challenge: Designing an interface that is intuitive for both experienced and novice users, avoiding overwhelming features that could hinder usability.

- **Content Quality Control**- Ensuring the shared recipes meet quality standards and contribute positively to the community.
  Challenge: Implementing effective moderation tools and mechanisms to maintain content quality, while still allowing users to express their culinary creativity.

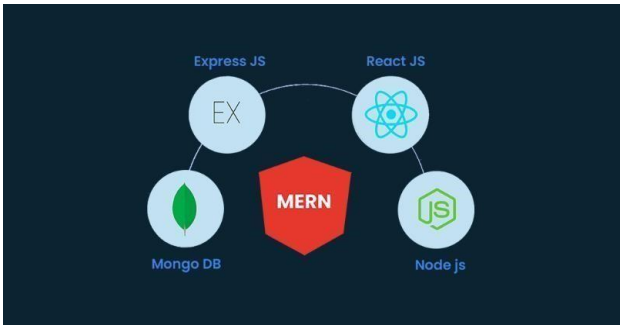
- **Technical Performance**- The application may face performance issues as the user base and content grow.
  Challenge: Optimizing the application's technical infrastructure to handle scalability, ensuring quick response times, and preventing downtime.

- **Privacy Concerns**- Users may be hesitant to share personal recipes or details due to privacy concerns.
  Challenge: Implementing robust privacy measures, transparent data handling policies, and ensuring user data security to build trust among users.

- **Continuous Improvement**- The model might struggle with adapting to evolving user needs and industry trends.
  Challenge: Establishing a system for collecting and analyzing user feedback, and consistently iterating on the application to introduce new features and improvements.[3]

## Tech Stack

1. Frontend: React js
2. Backend: node js and express js
3. Data base: Mongodb
4. Version control: Git



## Proposed Method

Using the MERN (MongoDB, Express, React, and Node.js) stack to develop a recipe sharing app requires a methodical and planned approach. The development process's major steps are outlined in the suggested way below:

- **Describe the App's Needs**- Describe the Recipe Sharing App's features and functionalities in detail. Take into account user roles, essential features such as commenting, sharing, and recipe creation, user profiles, and any other elements that complement the objectives of your app.

- **MongoDB Design Database Schema**- Establish the data structure for users, comments, recipes, and other pertinent entities. Create connections between various entities to guarantee effective data retrieval.

- **Construct Backend Infrastructure Using Express & Node.js**- Installing npm and Node.js on development system. Create a new project in Node.js. Installing Express.js with npm to create a robust backend. Configure the backend's routes, middleware, and error management.

- **Put user authentication (passport, JWT) into practice**- For safe and stateless authentication, integrate user authentication using JWT (JSON Web Tokens). Use Passport.js to expedite user sessions and authentication procedures.

- **Provide an API for CRUD (CREDIT, Delete, and Update) operations on MongoDB recipes**- Create routes on recipes for CRUD activities. Use MongoDB queries to communicate with the database and retrieve recipe data.

- **Create React.js front-end views and components**- Create a brand-new front-end React.js project. Create and design React components for various views, including user profiles, the homepage, and recipe information. React Router can be used to switch between views.

- **Establish a Frontend-Backend API connection (Axios,**) - With Axios or Fetch, link the frontend React application to the back-end API to send HTTP queries. Assure smooth data submission and retrieval between the frontend and backend.

- **Put User Interface & Interaction into Practice (React Router, Components**) - Features like recipe likes, comments, and sharing can improve user involvement. Create a UI that is responsive and easy to use to guarantee a positive user experience.

- **Increase the number of features and improve user experience**- Add more features, like categories, search capabilities, and tailored user recommendations. Iteratively improve the user interface by considering user feedback.

- **Check the functionality of the app (debugging, unit testing**) - Test every component in the front end and back end thoroughly using unit tests. Fix and debug any problems found during the testing process.

- **Constant Monitoring and Updates (Feature Updates, Bug Fixes) -** Put in place ongoing security and performance monitoring. Release updates often, fixing any faults and adding new features in response to user suggestions.

- **Open the app! -** Make the Recipe Sharing App available to users by launching it formally.

- **Development's End**- Celebrate your recipe-sharing app's successful development and release. For upcoming updates, keep an eye on user input and engagement.

Following this proposed method will help ensure a systematic and successful development process for your Recipe Sharing App using the MERN stack. Adjustments can be made based on specific project requirements and evolving user needs[6].

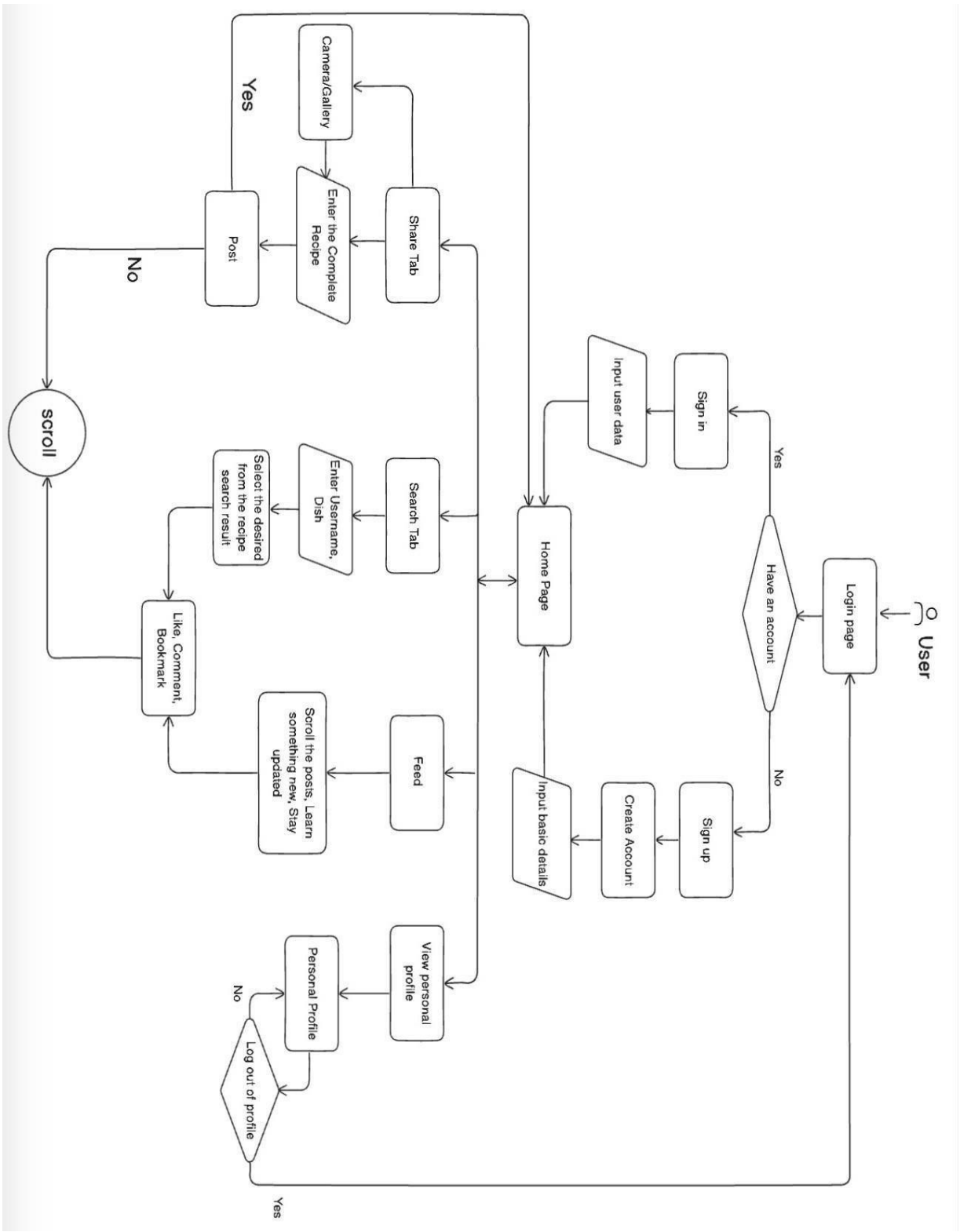
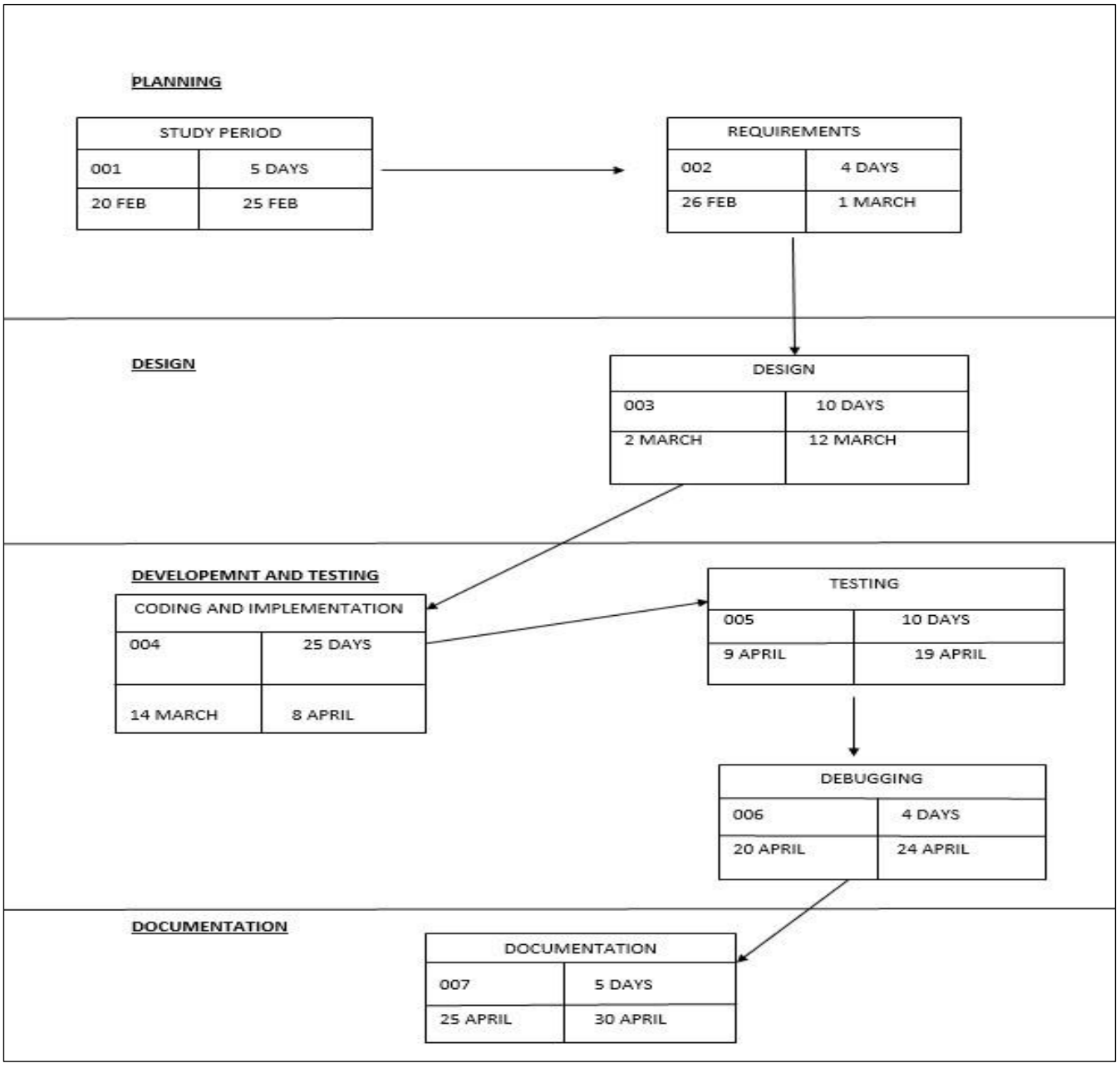



**Fig: Proposed Flowchart**

## Pert Chart

A PERT chart, short for Program Evaluation Review Technique, is a project management tool used to visually represent and analyze the tasks and timelines involved in completing a project. It helps identify critical activities, dependencies, and the optimal path for project completion, aiding in efficient planning and resource allocation.



## HOW THE PROJECT WORKS

The Recipe Sharing App using the MERN (MongoDB, Express.js, React.js, Node.js) stack works by allowing users to share, discover, and interact with recipes through a web-based platform. Here's how it generally functions:

1. **User Authentication:**

    - Users can register for a new account or log in with existing credentials.
    - Authentication ensures that users have personalized experiences and can access features like saving favorite recipes and interacting with the community.

2. **Recipe Creation and Management:**

    - Registered users can create and publish their own recipes by filling out a form with details such as ingredients, cooking instructions, and images.
    - Once submitted, recipes are stored in the database and made accessible to other users.

3. **Recipe Discovery:**

    - Users can explore a variety of recipes through different browsing options, including categories, tags, and search functionality.
    - The app may provide personalized recommendations based on the user's browsing history, saved recipes, and preferences.

4. **Interaction and Engagement:**

    - Users can interact with recipes by liking, commenting, and sharing them with others.
    - Social features foster community engagement and allow users to connect with fellow cooking enthusiasts.

5. **User Profiles:**

    - Each user has a profile page where they can view and manage their own recipes, saved recipes, and activity history.
    - Profiles may also showcase user details, such as bio, profile picture, and social links.

6. **Real-Time Updates:**

    - The app may implement real-time notifications to alert users about new recipes, comments on their recipes, or interactions from other users.
    - Real-time updates enhance user engagement and provide a seamless browsing experience.

7. **Admin and Moderation:**

    - Admin users may have special privileges to manage the platform, including approving new recipes, moderating user-generated content, and enforcing community guidelines.

8. **Scalability and Performance:**

    - The MERN stack allows for scalability and performance optimization to handle increasing user traffic and data volume.
    - Techniques such as database indexing, caching, and load balancing may be employed to ensure smooth operation of the application.

Overall, the Recipe Sharing App using MERN provides a user-friendly platform for sharing culinary creations, discovering new recipes, and connecting with a community of food enthusiasts. It leverages modern web technologies to deliver an engaging and interactive cooking experience for users[7].

## Code Summary

Below is a summary of the code structure for the Recipe Sharing App using the MERN (MongoDB, Express.js, React.js, and Node.js) stack:

1. **Backend (Node.js with Express.js):**

- server.js: Entry point for the backend application initializes Express server.
- Routes: Directory containing route handlers for different API endpoints.
- authRoutes.js: Handles user authentication and authorization.
- recipeRoutes.js: Manages CRUD operations for recipes.
- userRoutes.js: Handles user-related operations like profile management.
- Controllers: Directory-containing controller functions for each route.
- Models: Directory containing Mongoose schemas for User and Recipe models.
- Middleware's: Directory containing custom middleware functions for authentication, error handling, etc.
- Config: Directory-containing configuration files such as database connection setup and environment variables.
- Helpers: Directory containing utility functions used across the backend.

2. **Frontend (React.js):**

- src: Main directory for React application source code.
- components: Directory containing reusable UI components.
- Navbar.js: Navigation bar component.
- RecipeCard.js: Component to display individual recipe cards.
- RecipeForm.js: Component for creating/editing recipes.
- pages: Directory containing different pages/routes of the application.
- Home.js: Landing page displaying featured recipes.
- Login.js: Login page for user authentication.
- Register.js: Registration page for new users.
- Profile.js: User profile page displaying user details and saved recipes.
- RecipeDetails.js: Page displaying detailed information about a specific recipe.

- context: Directory containing React context providers for managing global state.
- AuthContext.js: Context provider for user authentication state.
- hooks: Directory containing custom React hooks for managing state and side effects.
- utils: Directory containing utility functions for API requests, authentication, etc.
- App.js: Main component defining application routing and layout.
- index.js: Entry point for React application, renders the root component.
- public: Directory containing static assets like images and favicon.

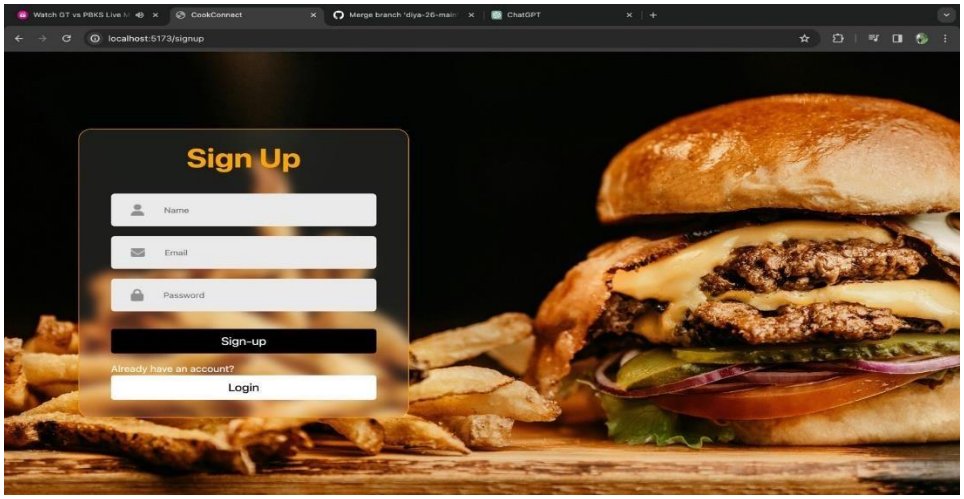
**3. Database (MongoDB):**

- MongoDB database to store user information, recipe data, and other application-related data.

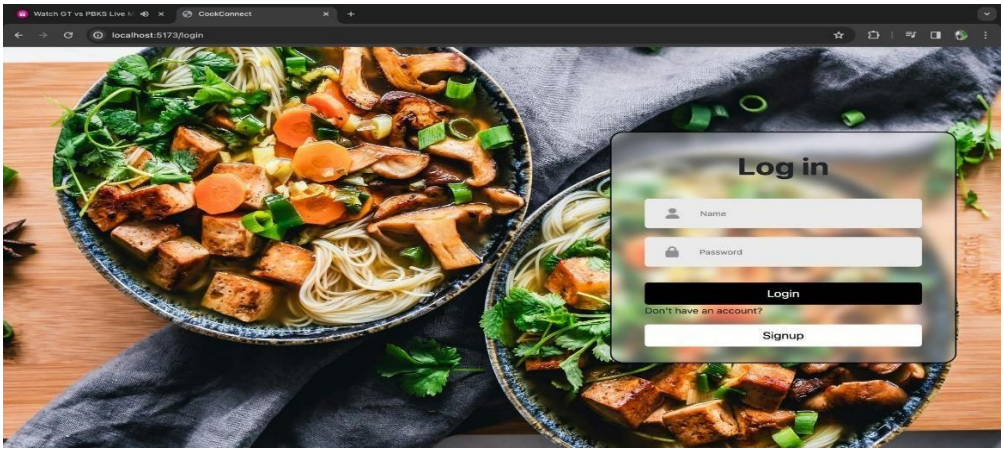
This structure provides a modular and scalable approach for building a Recipe Sharing App using the MERN stack, separating concerns between frontend and backend, and organizing codebase for easier maintenance and collaboration

## Objective Achieved Frontend:

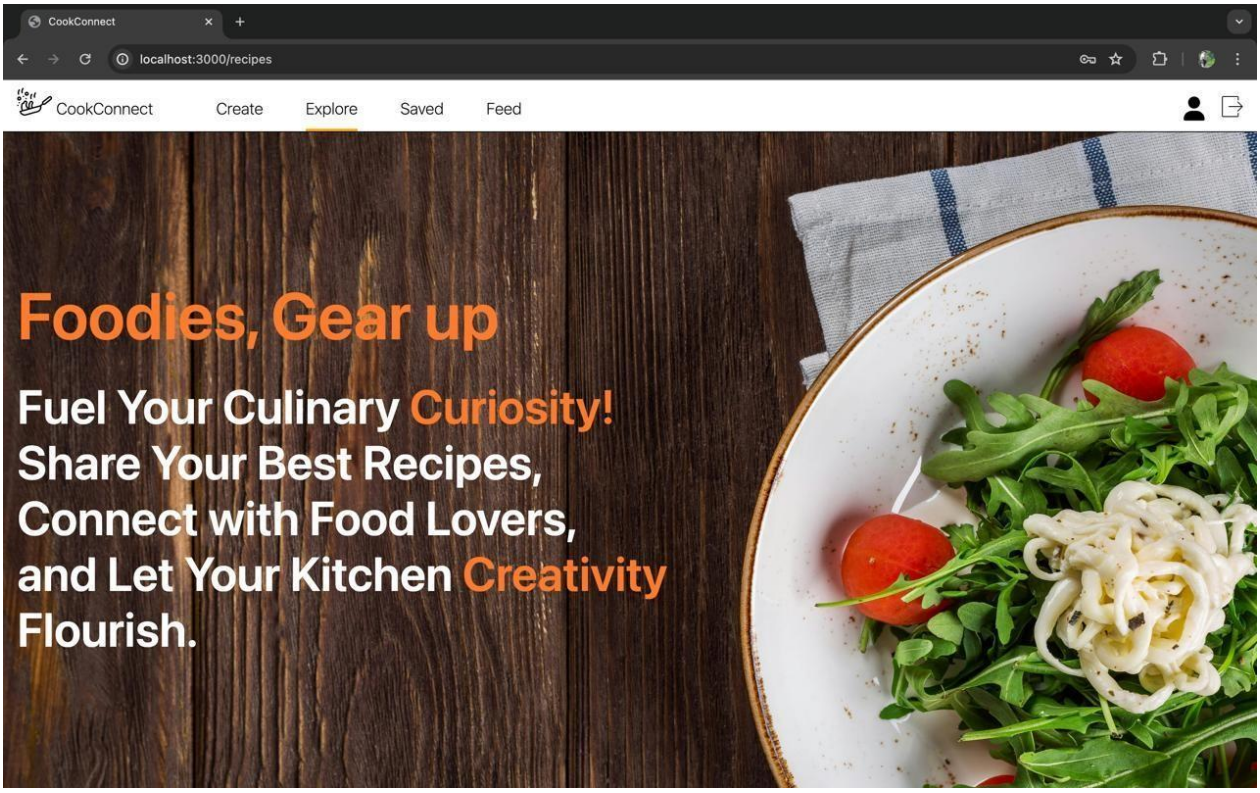
1. **Sign up Page**



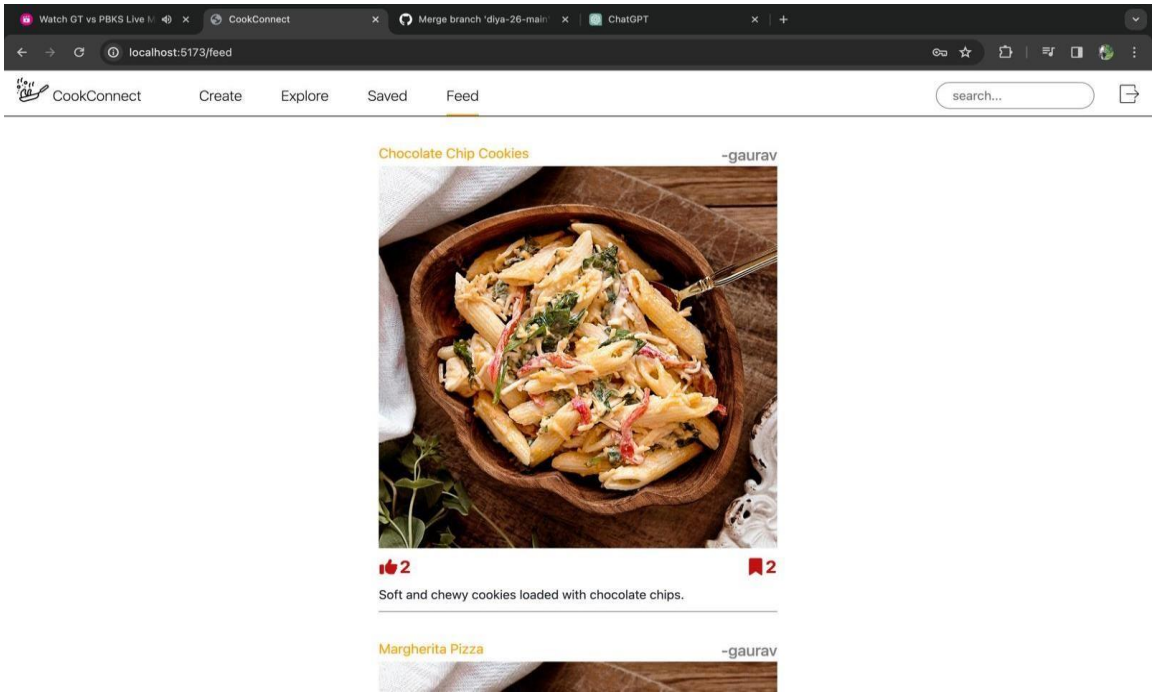

2. **Login Page**



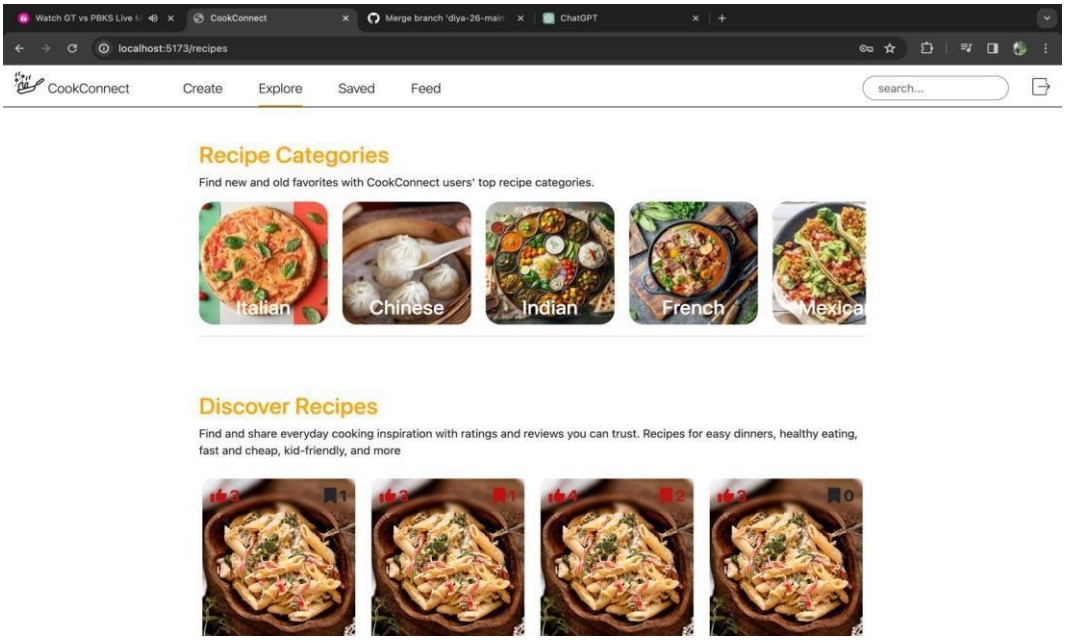

### 3. Landing Page



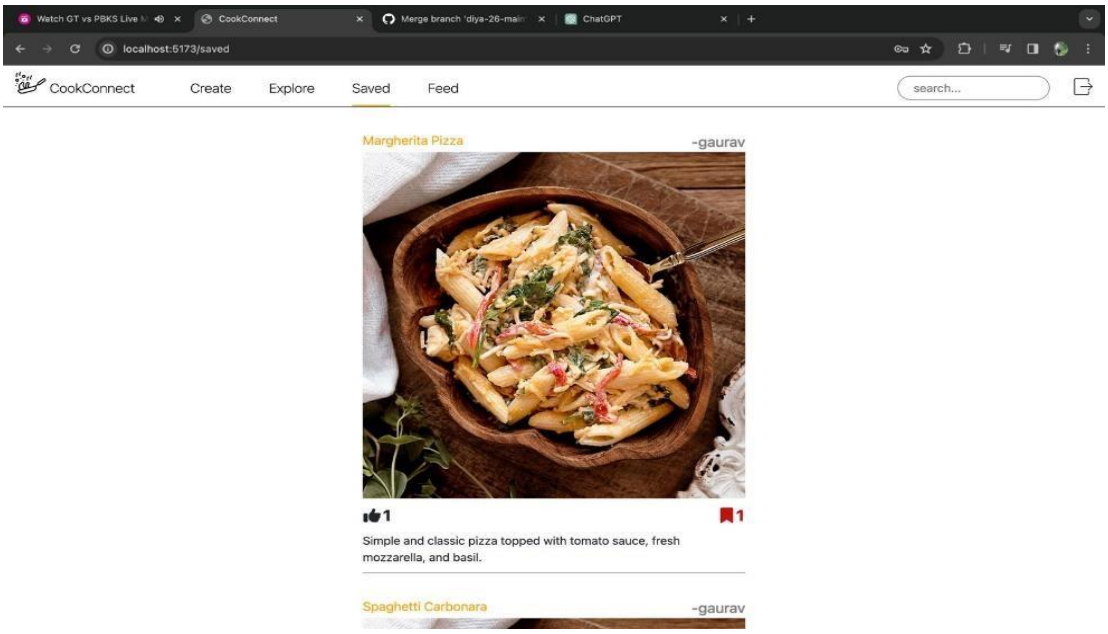
## 4. General Feed



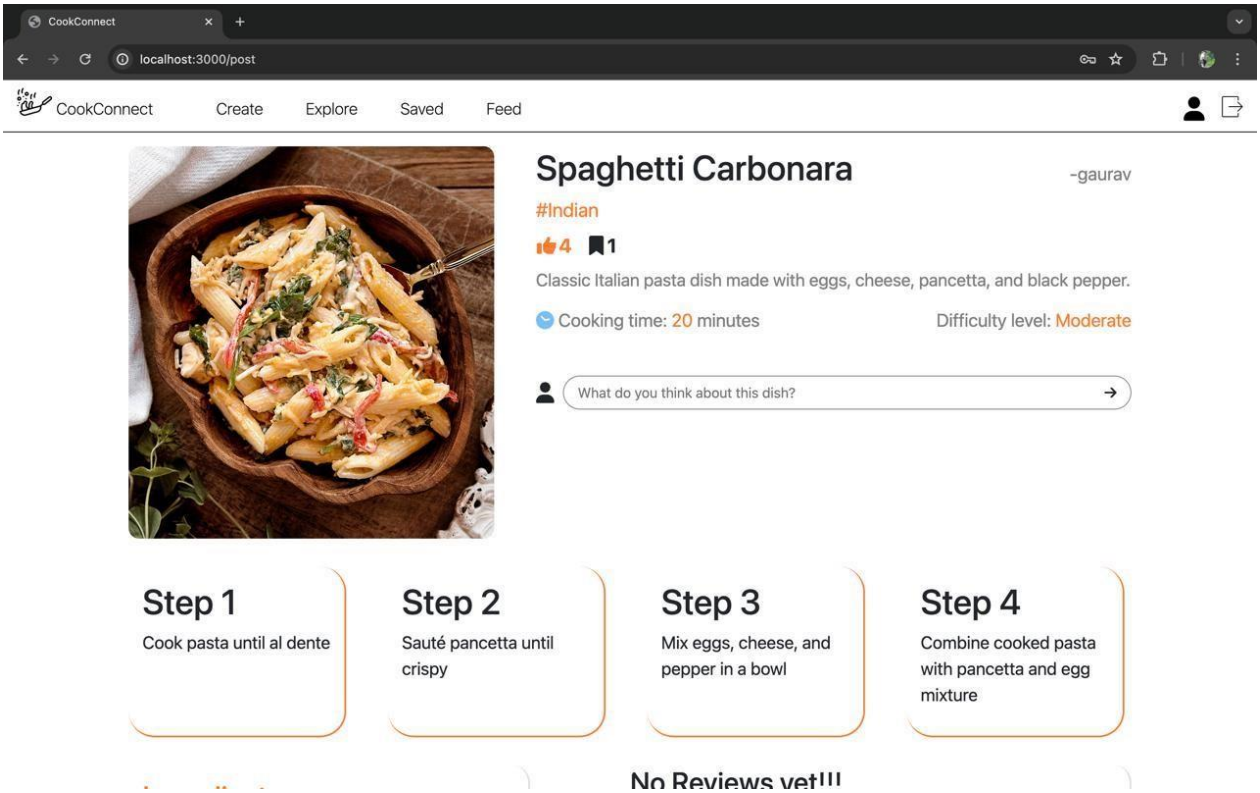

## 5. Explore Page



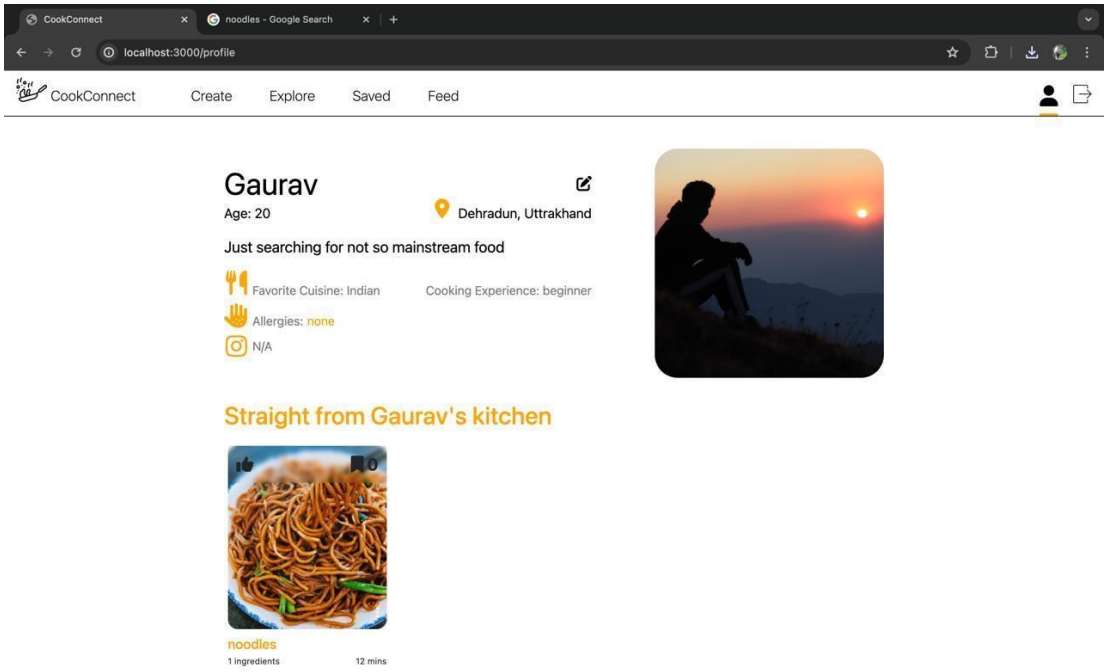
## 6. Saved post view



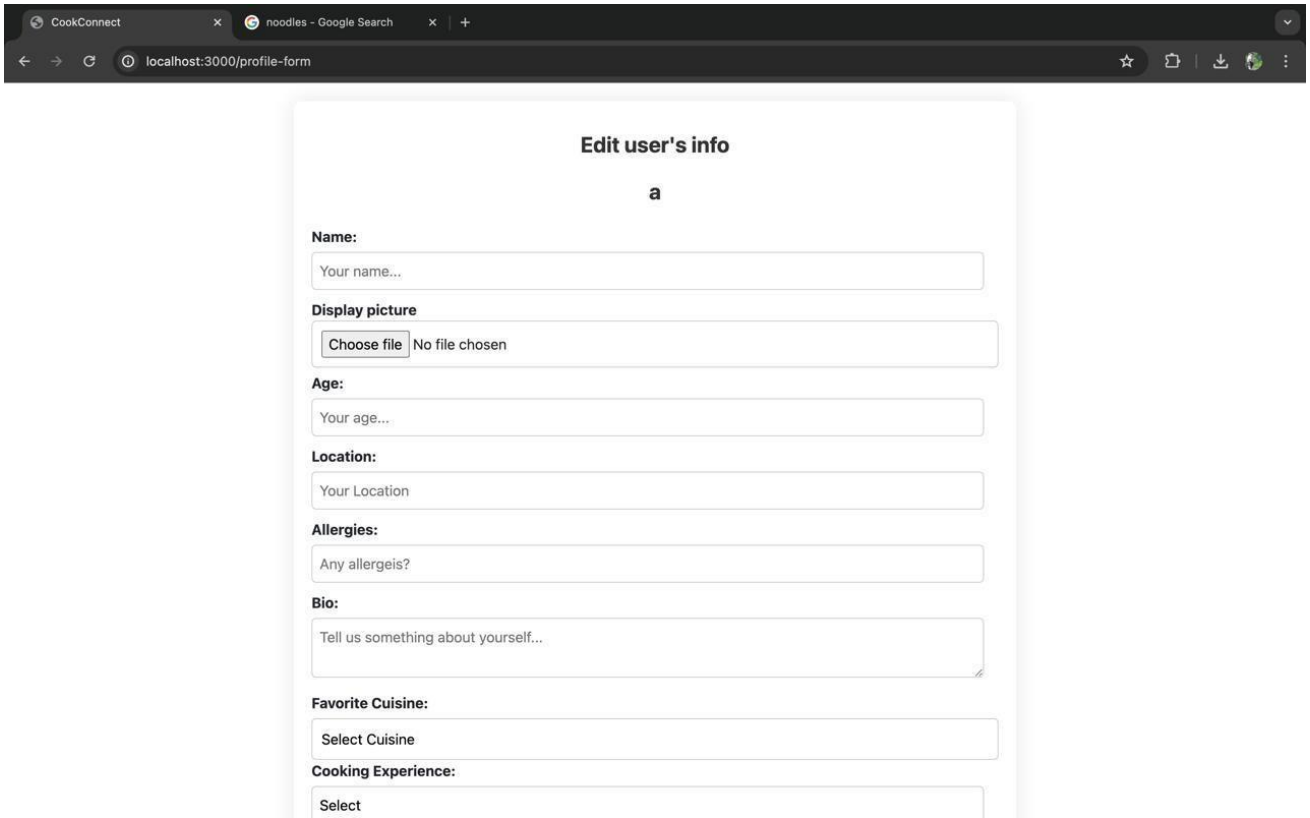

## 7. Detailed Recipe Page



## 8. User Profile




## 9. Edit User Profile



## Other Objectives achieved:

10. Logout
11. Like post
12. Saved post

## Objectives Achieved Backend:

**1. Backend Development:**

- **Authentication and Authorization:**
  - Implemented user authentication using Passport.js with local strategy.
  - Users can sign up, log in, and log out securely.
  - Session management is integrated using Express Session and MongoDB for session storage.

- **User Management:**
  - Created endpoints for user registration, login, logout, and session checking.
  - Implemented functionality to retrieve registered user data.

- **Recipe Management:**
  - Developed endpoints for creating, retrieving, liking, and saving recipes.
  - Users can like and save recipes, and the application tracks the number of likes and saves for each recipe.
  - Implemented endpoints to check the like and save status of a recipe.
  - Added functionality to fetch saved recipes for a user.

- **Middleware and Error Handling:**
  - Implemented middleware for checking user authentication status.
  - Set up error handling for various scenarios to ensure robustness and reliability of the application.

- **Frontend Development:**
  - Start frontend development to create an intuitive and engaging user interface.
  - Design wireframes and prototypes for user interaction.

- **Enhancements:**
  - Implement additional features such as recipe categorization, search functionality, and user profile customization.
  - Improve the user experience through responsive design and performance optimization.

- **Testing:**
  - Conduct thorough testing to ensure functionality across different devices and browsers.

- **Challenges Faced:**

1. Session Management:
    - Configuring session management with MongoDB presented some challenges, but they were successfully resolved through research and experimentation.

2. Data Modeling:
    - Designing the data model for recipes and user interactions required careful consideration to ensure efficiency and scalability.

**2. Backend Report: Recipe Sharing MERN Application**

**a** **Middleware Configuration:** We have successfully set up middleware in our Express application to handle JSON requests efficiently. Additionally, session management has been established using the express-session middleware, which is configured with a secret key for enhanced security. To store session data, we've integrated MongoStore with MongoDB.

**b** **Authentication Implementation:** Integration of Passport.js for user authentication has been completed. We've utilized the local strategy provided by passport-local for authentication.
Serialization and deserialization methods have been implemented to manage user authentication state effectively.

**c** **User Authentication Endpoints**

a. Signup Endpoint (/signup): Users can register with a unique username, email, and password. We've utilized passport-local-mongoose for seamless user registration.
b. Login Endpoint (/login):Handles user login by authenticating users based on their username and password.
c. Logout Endpoint (/logout): Enables users to log out, effectively terminating their session.

**d** **Authentication Middleware:** A custom middleware function named `isLoggedIn` has been implemented to ensure user authentication before granting access to protected routes.

**e** **Recipe Management Endpoints**:

a. Create Recipe Endpoint (/recipes/create): Users can create new recipes with detailed information and an uploaded image. Multer middleware is employed for image processing.
b. Fetch Recipes Endpoint (/recipes/:page/:limit):Retrieves paginated lists of recipes sorted by creation date, providing pagination and sorting functionality.
c. Fetch Recipe Details Endpoint (/recipes/info/:recipeId): Fetches detailed information about a specific recipe, including author details and the number of likes.

**f** **Like and Save Functionality:** Endpoints have been developed to enable users to like and save recipes, with corresponding functionality to unlike and unsave recipes as well.

**g** **User Data Management Endpoints:** A dedicated endpoint has been created to fetch usernames and emails of all registered users for administrative purposes.

h **Image Processing and Storage:** Multer middleware has been utilized for processing uploaded images during recipe creation. Cloudinary integration facilitates seamless storage and retrieval of recipe images from the cloud.

i **Error Handling and Logging:** We have implemented error handling mechanisms to log errors for debugging purposes. Responses to client requests include appropriate HTTP status codes and error messages.

## 3. Database Overview: MongoDB

**Database Purpose:** MongoDB is utilized as the primary NoSQL database for the Recipe Sharing MERN application. It offers flexibility and scalability, making it suitable for storing and managing diverse data structures and relationships between users and recipes.

**User Schema:**

- Schema Definition: The User schema, defined using Mongoose, represents user data with essential fields such as email and saved recipes.

**Purpose:**

- Email: Stores the user's email address for identification and communication purposes.
- Saved: Represents an array of recipe IDs, indicating recipes that the user has saved for future reference.

**Passport Integration:** Passport-local-mongoose plugin is integrated for authentication, automating the addition of username, hash, salt, and authentication functions to the schema.

**Recipe Schema:**

**Schema Definition:** The Recipe schema captures comprehensive details about recipes, including title, cuisine, description, ingredients, author, and more.

**Purpose:**

- **Title:** Holds the title of the recipe for easy identification.
- **Cuisine:** Represents the cuisine type of the recipe, facilitating categorization and filtering
- **Description:** Provides a brief overview or summary of the recipe.
- **Steps:** Stores an array of strings outlining the cooking steps required to prepare the recipe.
- **Ingredients:** Contains an array of strings listing the ingredients needed for the recipe.
- **Instructions:** Offers additional instructions or notes for preparing the recipe.
- **Cooking Time:** Represents the estimated time required to cook the recipe.
- **Difficulty Level:** Indicates the complexity level of preparing the recipe, aiding users in choosing suitable recipes.
- **Image:** Stores the URL or path to the image associated with the recipe for visual appeal.

- **Author:** References the user who created the recipe, establishing a relationship between users and recipes.
- **Rating:** Stores the average rating of the recipe, allowing users to assess recipe quality.
- **Likes represents** an array of user IDs who have liked the recipe, enabling interaction and engagement.
- **Saved By:** Tracks the count of users who have saved the recipe, providing insights into recipe popularity.

**Pre-Save Hook:** A pre-save hook is implemented to automatically set the creation date (created at) when a new recipe document is created, enhancing data consistency and integrity.

## Conclusion:

In conclusion, this MERN stack recipe sharing application offers a dynamic platform for both novice and experienced cooks. By leveraging the strengths of MongoDB, Express.js, React.js, and Node.js, the application fosters:

**Rich recipe exploration:** Users can discover a vast collection of recipes, filtered by various criteria to suit their preferences and dietary needs.

**Vibrant community:** User profiles, recipe reviews, comments, and social features encourage interaction and knowledge sharing among food enthusiasts.
Personalized experience: Saved recipes, meal plan creation, and potential recommendation features can personalize the user journey and inspire culinary exploration.

## References:


1. Developing a Social Platform using MERN Stack Developing a Social Platform using MERN Stack Desai, Krutika; Fiaidhi, Jinan (2022): Developing a Social Platform using MERN Stack. TechRxiv.

2. Recipe Sharing Jeenu. M. S and Prof. Miriam Thomasm IV Semester MCA, Sree Narayana Institute of Technology, Kollam, Kerala1 Asst. Professor, Dept. of Computer Applications, Sree Narayana Institute of Technology, Kollam, Kerala

3. Recipe recommendations for individual users and groups in a cooking assistance app Toon De Pessemier Kris Vanhecke, Anissa All , Stephanie Van Hove , Lieven De Marez , Luc Martens ·Wout Joseph , David Plets

4. Foodorials- A Cooking Recipe Android AppSourabh D. Mane1, Razia Z. Ratlamwala2, Vinit S. Jain, Prof. Rahul Patil Final Year Computer Engineering, BVCOE, Navi Mumbai, India4Professor, Dept. of Computer Engineering, Bharati Vidyapeeth College of Engineering,Maharashtra, Navi Mumbai, India.

5. Developing a Social Platform using MERN Stack1stKrutika DesaiMSc Computer

ScienceLakehead UniversityOntario, Canadadesaik@lakeheadu.ca2nd Dr. Jinan FiaidhiDepartment of Computer ScienceLakehead UniversityOntario,

6. Food Recipe Finder Mobile Applications Based On Similarity Of Materials Gusti Pangestu Faculty of Computer Science,Brawijaya University Malang, Indonesia, Supianto Faculty of Computer Science,Brawijaya University Malang, Indonesia,Fitri Utaminingrum Faculty of Computer Science,Brawijaya University Malang, Indonesia.

7. QUICK RECIPES ANDROID APPLICATION,A Project Presented to the faculty of the Department of Computer Science California State University, Sacramento,MASTER OF SCIENCE in Computer Science by Ruchi Gupta.

8. Recipe Planning Application Angela Anak Abong a, Zehan Afizah binti Afip, Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, Johor, Malaysia