UNIVERSITY OF PETROLEUM AND ENERGY STUDIES, DEHRADUN

Recipe Sharing Application

Software Requirement Specification

(Minor Project-II, VI Sem)

Submitted by:

S.No.	Student Name	Roll No.	Sap ID
1	Akshay Mohpal	R2142210078	500088177
2	Diya Khandelwal	R2142210293	500090939
3	Gaurav Bhandari	R2142210311	500090993
4	Kanishka Singh	R2142210393	500091377

BACHELOR OF TECHNOLOGY, COMPUTER SCIENCE ENGINEERING

With specialization in DevOps

Under the guidance of

Mr. Sandeep Pratap Singh

School of Computer Science (SOCS)

Department of CSO, UPES

Bidholi Campus, Energy Acres, Dehradun – 248007

Software Requirements Specification (SRS)

The introduction of the Software Requirements Specification (SRS) provides an overview of the entire SRS with purpose, scope, definitions, acronyms, abbreviations, references and overview of the SRS. The aim of this document is to gather, analyze, and give an in-depth insight of the complete **Project** by defining the problem statement in detail. Nevertheless, it also concentrates on the capabilities required by stakeholders and their needs while defining high-level product features. The detailed requirements of the **Project** are provided in this document.

INDEX

Recipe Sharing Application

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.

Purpose: This document's aim is to give a comprehensive overview of the Recipe Sharing App, outlining its features, needs, and components.

Scope: MERN stack recipe-sharing app, Create, explore, and share recipes while using user authentication and social features.

Definitions, Acronyms, and Abbreviations: In a recipe sharing app, users can search, save, and contribute recipes (often with photos and reviews) while interacting with a community through comments, likes, and personalized recommendations.

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.

4. **Deployment with Amazon EKS:** Containerize your application with Docker to ensure consistency and portability. Configure Kubernetes for deploying and managing containers on

Amazon EKS. Use EKS capabilities such as Auto Scaling to handle varying loads efficiently. Set up a CI/CD pipeline to automate the deployment process[1].

Over all descriptions

A recipe sharing app tackles the dilemma of "what to cook for dinner?" by offering a searchable recipe database with user-contributed content. Users can browse, save, and share recipes (often with visuals and reviews) while fostering a community through comments, likes, and personalized recommendations[2].

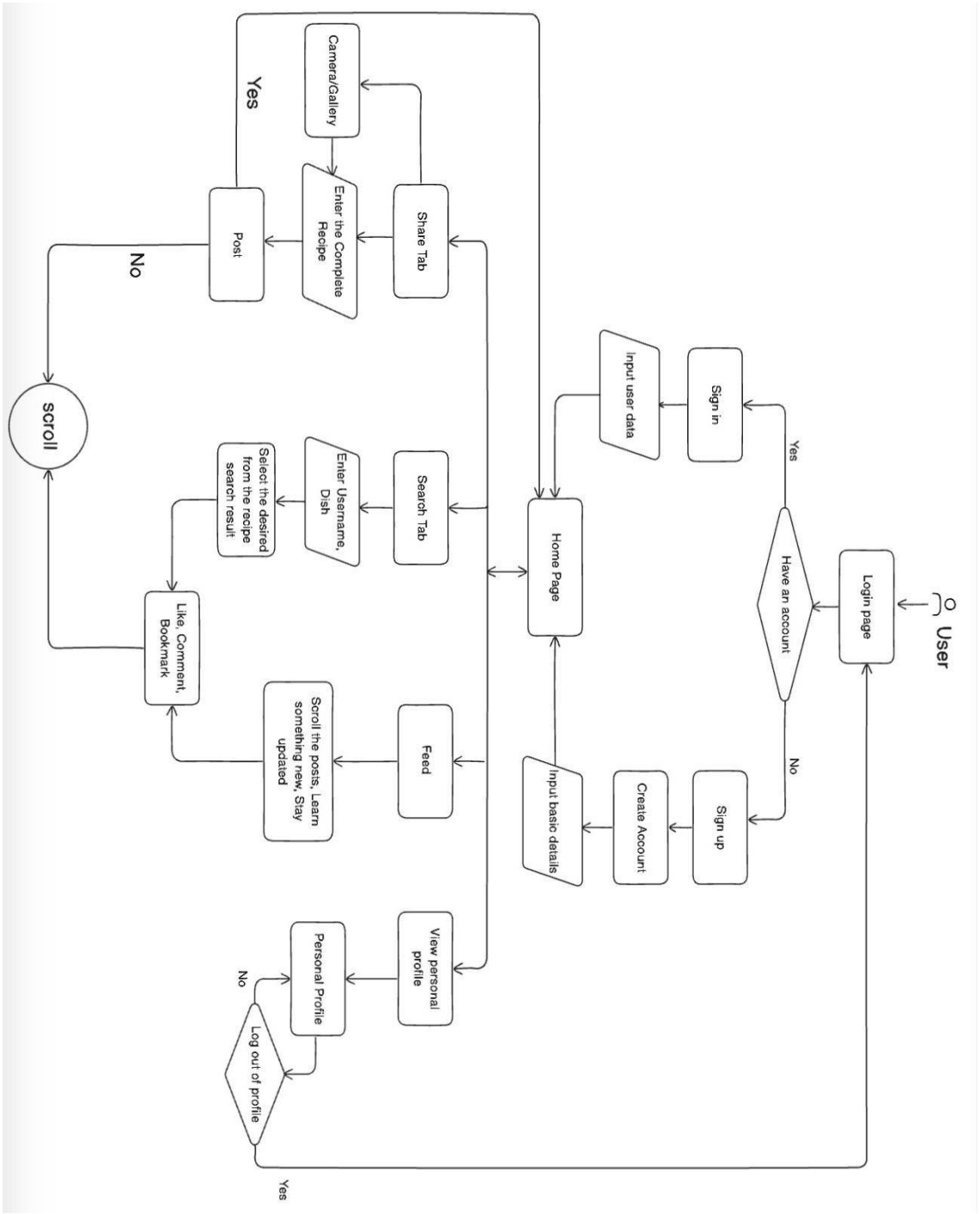


Fig: Proposed Flowchart

How the project works?

Initialization:

Building a Recipe Sharing App involves several key steps, from planning and development to deployment and maintenance. Here is a general outline of how the project might work[3]:

1. **Define Goals and Features**
   - Clearly outline the goals of your recipe sharing app.
   - Identify key features such as user registration, recipe submission, browsing/searching, commenting, and rating.

2. **Market Research:**
   - Analyze existing recipe apps to understand their features, strengths, and weaknesses.
   - Identify your target audience and their preferences.

3. **Wire framing and Design**
   - Create wireframes to visualize the app's structure and user interface.
   - Design the app's user interface and experience (UI/UX).

4. **Database Design**
   - Decide on the database structure to store user data, recipes, comments, and other relevant information.
   - Choose a database management system (DBMS) that fits your needs.

5. **Backend Development**
   - Set up the server-side logic and database connectivity.
   - Develop APIs (Application Programming Interfaces) for the frontend to communicate with the backend.

6. **Frontend Development**
   - Build the user interface based on the design.
   - Implement features like user registration, recipe submission, and browsing.

7. **Integration**
   - Connect the frontend and backend components to ensure seamless communication.
   - Implement user authentication and authorization.

8. **Testing**
   - Conduct thorough testing to identify and fix bugs.
   - Perform usability testing to ensure a smooth user experience.

9. **Deployment**
   - Choose a hosting platform and deploy your app.
   - Configure domain settings if you have a custom domain.

10. **Monitoring and Maintenance**
    - Implement monitoring tools to track app performance.
    - Regularly update and maintain the app to fix bugs and introduce new features.

11. **User Feedback and Iteration**
    - Gather user feedback through reviews, surveys, or analytics.
    - Use feedback to make improvements and release updates.

12. **Marketing and Promotion**
    - Promote your app through various channels to attract users.
    - Consider social media, influencers, and other marketing strategies.

13. **Scaling (if necessary)**
    - If your app gains popularity, be prepared to scale your infrastructure to handle increased traffic.

14. **Legal Considerations**
    - Ensure that your app complies with relevant laws and regulations.
    - Consider implementing terms of service and privacy policies.

Specific Requirements[4]:

Usability:

- The user interface must be simple to use and navigate.
- All user interactions must be accompanied by clear instructions and prompts.

Reliability & Availability:

- The app should be accessible to users most of the time, minimizing downtime for recipe browsing and sharing.
- User-saved recipes, preferences, and community interactions should remain consistent and accessible across sessions.
- The app's infrastructure should handle surges in users and recipe uploads without compromising performance or stability.

Performance:

- Utilize indexing in MongoDB to ensure fast recipe searches based on ingredients, cuisine, etc.
- Implement lazy loading for images and recipes to minimize initial load time and improve responsiveness.
- Cache frequently accessed data (like static content or popular recipes) to reduce server load and improve browsing speed.

Security:

- Security must be maintained throughout user authentication and authorization.
- Modern encryption techniques must be used to save passwords.
- To prevent common internet vulnerabilities, there must be security measures in place.
- Access to sensitive information is only permitted for authorized individuals.

Supportability:

- MERN stack offers a large developer pool familiar with the technologies (Node.js, React, and MongoDB) for easier maintenance and future support.

Design constraints:

- The solution must work with popular web browsers including Chrome, Firefox, and Safari.
- It must be created with mobile access on multiple devices in mind.

On-line User Documentation and Help System Requirements:

- MERN stack recipe app user doc & help system, provide clear instructions on using the app's features (search, upload, and share) and FAQs for troubleshooting.

Purchased components:

- Pre-built UI components (e.g., buttons, forms) for faster front-end development in React.

Interfaces:

- The system must offer interfaces for administrators and end users.

Licensing requirements:

- All applicable licensing agreements for any third-party components or libraries utilized by the system must be complied with.

Legal copyright and other notices:

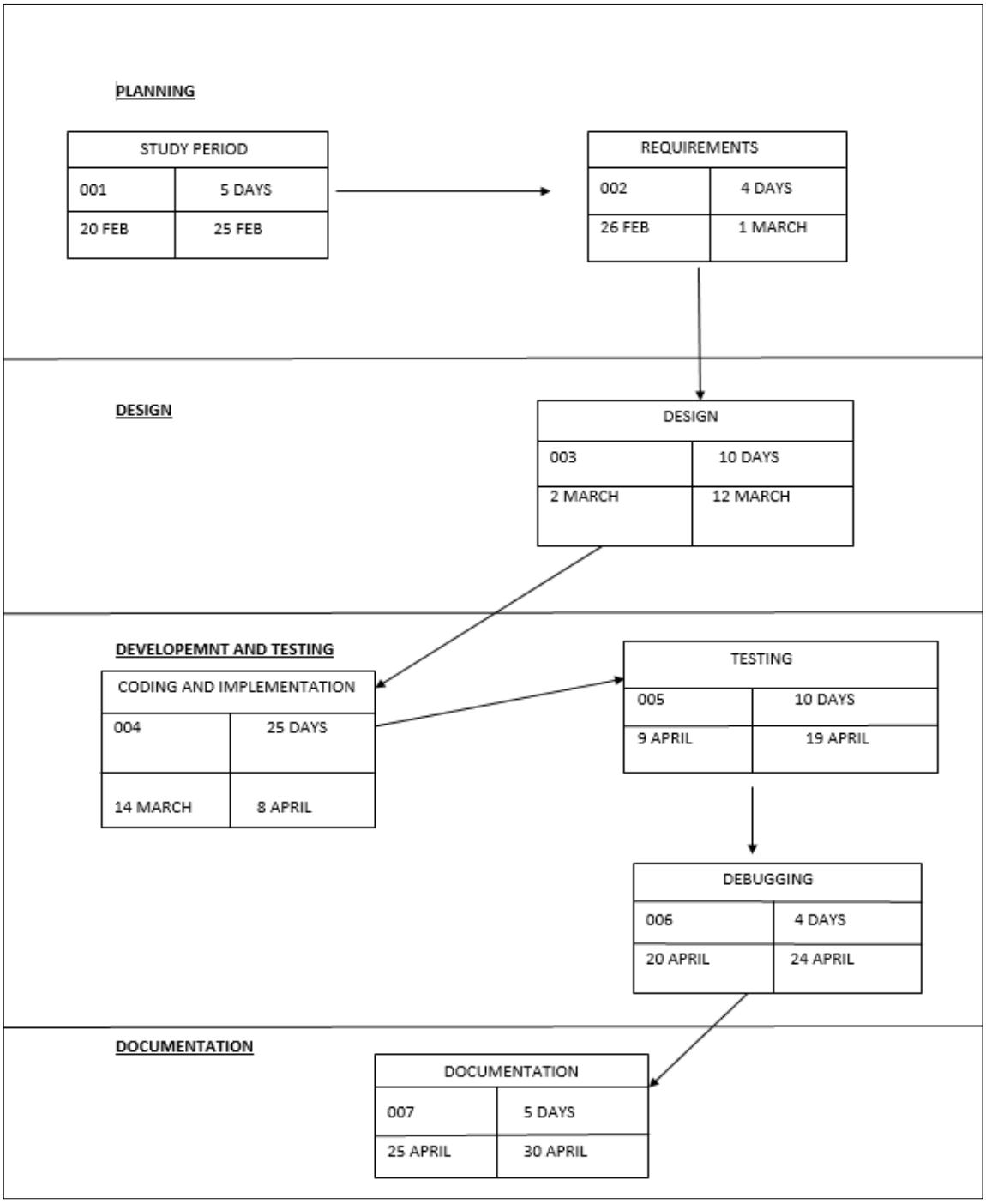
- It is necessary to be aware of and to respect all copyright and intellectual property rights.

Applicable Standards:

- The system must follow industry standards and best practices for user experience, security, and web application development.

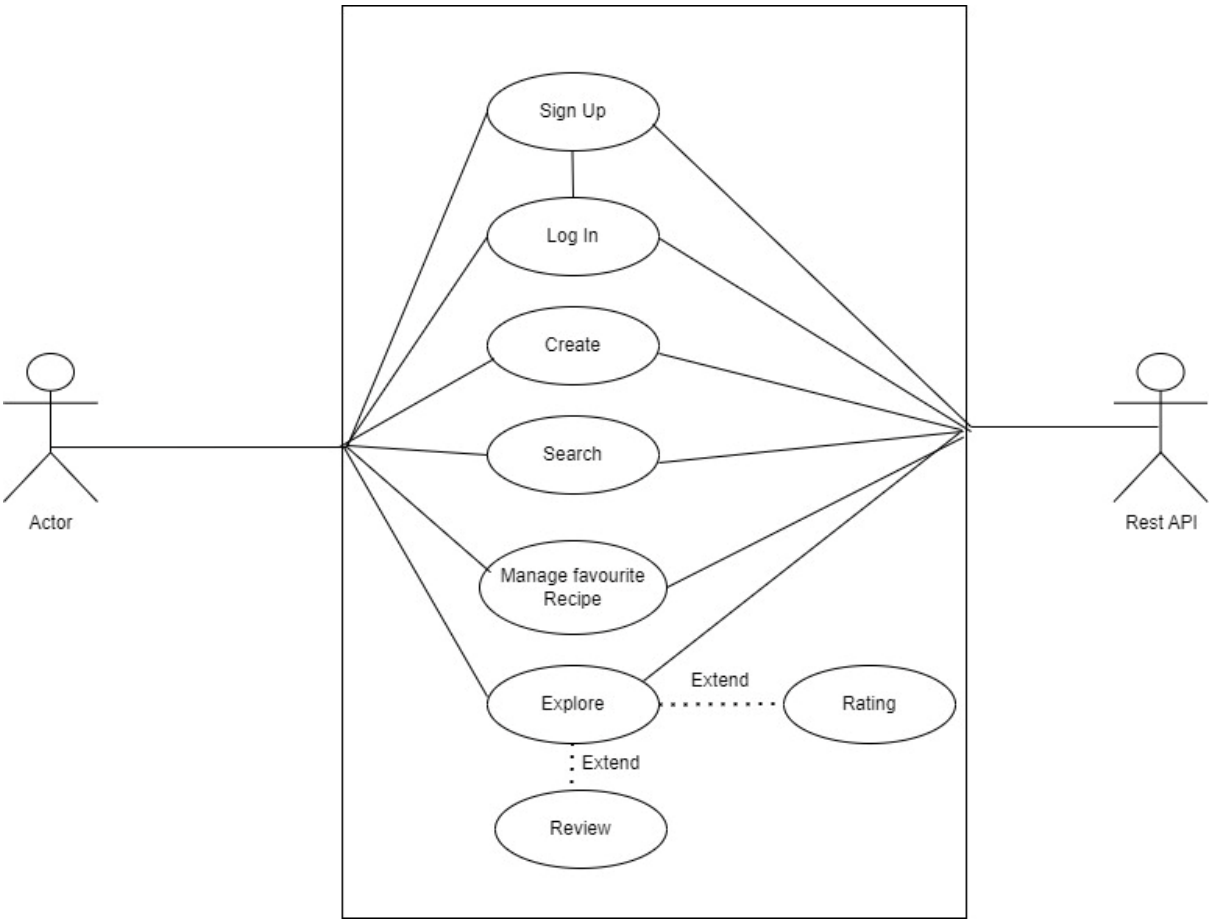
Pert Chart

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.



Use case analysis

Use case analysis is a technique used in software development and requirements engineering to define and document how users or actors interact with a system or software application. It helps capture functional requirements and understand the system's behavior from the user's perspective. Below, some key use cases for our Recipe Sharing Application are outlined below[5]:



Supporting Information

1. **Structural models.**

   It is a streamlined recipe sharing application powered by advanced structural models. Users can easily submit, discover, and interact with recipes from various cuisines and dietary preferences. Key features include personalized recommendations, social interaction, and community engagement. With its mobile compatibility and intuitive interface, the application offers a seamless culinary experience for food enthusiasts worldwide.

2. **Behavioral models.**

   The recipe suggestions based on users' tastes and behaviors. It adapts search results and offers personalized cooking tips. Users can share their culinary experiences and join cooking communities. The app assists in meal planning by suggesting recipes and

generating shopping lists. Continuous improvement is driven by user feedback and data analysis.

3. Nonfunctional requirements model.

1. **Performance**: Ensure fast and responsive recipe search and browsing, with minimal loading times for images and content.
2. **Security:** Implement robust user authentication and data encryption measures to safeguard user information and recipe data.
3. **Scalability:** Design the application architecture to handle increasing user traffic and accommodate future growth in recipe submissions and user interactions.
4. **Accessibility:** Ensure the application is accessible to users with disabilities, complying with WCAG guidelines for usability and inclusivity.
5. **Reliability:** Maintain high availability and uptime, minimizing downtime and ensuring uninterrupted access to recipe sharing functionalities.

4. Project Plan

The Recipe Sharing App project aims to revolutionize the way users share and discover recipes online. This strategic project focuses on enhancing user engagement, simplifying recipe management, and fostering a vibrant community of food enthusiasts[6].

a. Planning and preparation:

- Conduct a comprehensive analysis of existing recipe sharing platforms.
- Collect data on user preferences, popular recipes, and browsing behaviors.
- Define categories, tags, and user roles for effective organization and accessibility.

b. Design of a Traceability Matrix:

- Develop a customized Traceability Matrix based on gathered insights.
- Identify variables such as recipe types, ingredients, and user interactions.
- Establish criteria for recipe discovery, including search relevance and personalized recommendations.

c. Algorithm Development for Allocation:

- Design a sophisticated recommendation algorithm tailored to recipe sharing.
- Utilize the Traceability Matrix to optimize recipe suggestions based on user preferences.
- Conduct rigorous testing to ensure accurate and relevant recipe recommendations.

d. Real-time Updates:

- Implement a real-time notification system to keep users informed about new recipes, comments, and activity.

- Integrate monitoring tools to track user engagement, recipe popularity, and community interactions.
- Continuously refine and update the platform based on user feedback and usage patterns.

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[7].

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, CANADA.

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.

Approved by: Mr. Sandeep Pratap Singh