Sri Lanka Institute of Information Technology

Programming Applications and Frameworks (IT3030)

Final Assignment Report

GROUP ID: Y3S1-WE-17

Cook-Book

Student ID	Student Name
IT22127778	Chavindee M.A. P
IT22235688	Perera K.T. K
IT22208576	Fernando M.S.V.
IT22272768	Basnayake W.B.M.D. K

Date: 2024.05.07

Table of Contents

1. Introduction

The Cooking Skill-Sharing & Learning Platform is an interactive web-based application that allows users to share, explore, and learn various cooking techniques and recipes. The platform serves as a community-driven space where cooking enthusiasts can document their progress, engage with other users, and enhance their culinary skills. The system is designed to provide users with the ability to create and share recipes, upload images and short videos, follow structured cooking plans, and interact with others through likes, comments, and notifications. Users can also track their learning journey by posting progress updates and creating personalized cooking plans.

1. **Recipe Management**

Users can create and share their own recipes with the community. Each recipe post includes step-by-step instructions, ingredient lists, and media such as images or short videos. Users can categorize their recipes based on cuisine type, cooking difficulty, or preparation time, making it easier for others to discover new dishes. Additionally, users can edit or delete their recipes as needed.

2. **Post sharing and notifications**

The *Post Sharing & Notification Management* module is a key part of our Cookbook Project, designed to enhance user engagement and communication. This feature allows users to share their favorite recipes as posts with the community and receive real-time notifications about interactions such as likes, comments, or when a new recipe is posted. It ensures users stay informed and connected, creating an interactive and lively cooking community within the app

.

3. **Recipe Rating & Review System**

Users can rate recipes on a 1 to 5-star scale and leave reviews to help others determine the quality of a dish. The system calculates an average rating for each recipe based on user feedback. Reviews can be edited or deleted by the author, ensuring users have control over their input. Highly rated recipes may be featured on a "Top Recipes" section, guiding others to the best dishes.

4. **Cooking Challenges & Competitions Users**

can participate in themed cooking challenges and competitions. These events allow users to submit their best dishes, which the community can vote on. Winners may receive recognition in a “Recipe of the Month” section. This feature encourages friendly competition and fosters creativity among users while promoting engagement within the platform.

2. Functional Requirements & Non-Functional Requirements

Functional Requirements

REST API Functional Requirements

1. **Authorization and Authentication**
    - Enable secure user login and access control using standard authentication methods.
    - Support multiple authentication techniques such as:
        - Email/password-based login
        - OAuth 2.0 (Google, Facebook, etc.)

2. **HATEOAS (Hypermedia as the Engine of Application State)**
    - API responses must include navigational links to relevant resources (e.g., self, next, previous, related) to improve client-side navigation and discovery.

3. **Resource Manipulation (CRUD)**
    - Support full CRUD operations with proper HTTP verbs:
        - POST – Create new resources
        - GET – Read resources
        - PUT – Update resources
        - DELETE – Remove resources
4. **Rate Limiting**
    - Implement request throttling mechanisms to limit excessive API usage.
    - Define rate limits per user/IP to ensure fair usage (e.g., 100 requests/minute).
5. **Cacheable**
    - Support caching for GET requests using HTTP cache headers (e.g., ETag, Last-Modified) to reduce load and latency.

Client Web Application Functional Requirements

1. **User Registration and Authentication**
    - Allow users to register and authenticate using:
        - Email
        - Phone number
        - Social accounts (Google, Facebook, etc.)
    - Secure login with session or token-based authentication (e.g., JWT).
2. **Commenting**
    - Users can:
        - Post comments
        - Like/dislike comments

        - Reply to existing comments
3. **Follow and Unfollow**
    - Users can follow or unfollow:
        - Other users
        - Skills
        - Groups
        - Categories
    - Receive updates and personalized suggestions.
4. **Post Reviews**
    - Users can share reviews (text, ratings, and media).
    - Enable cross-platform sharing (e.g., to Instagram, Facebook, etc.).
5. **Share Posts**
    - Enable re-sharing of other users' posts with added comments or insights.
    - Shared posts appear in user feeds.
6. **Notification System**
    - Real-time or batched notifications for:
        - New comments
        - Likes
        - Follows
    - Allow users to customize notification settings.

Non Functional Requirements

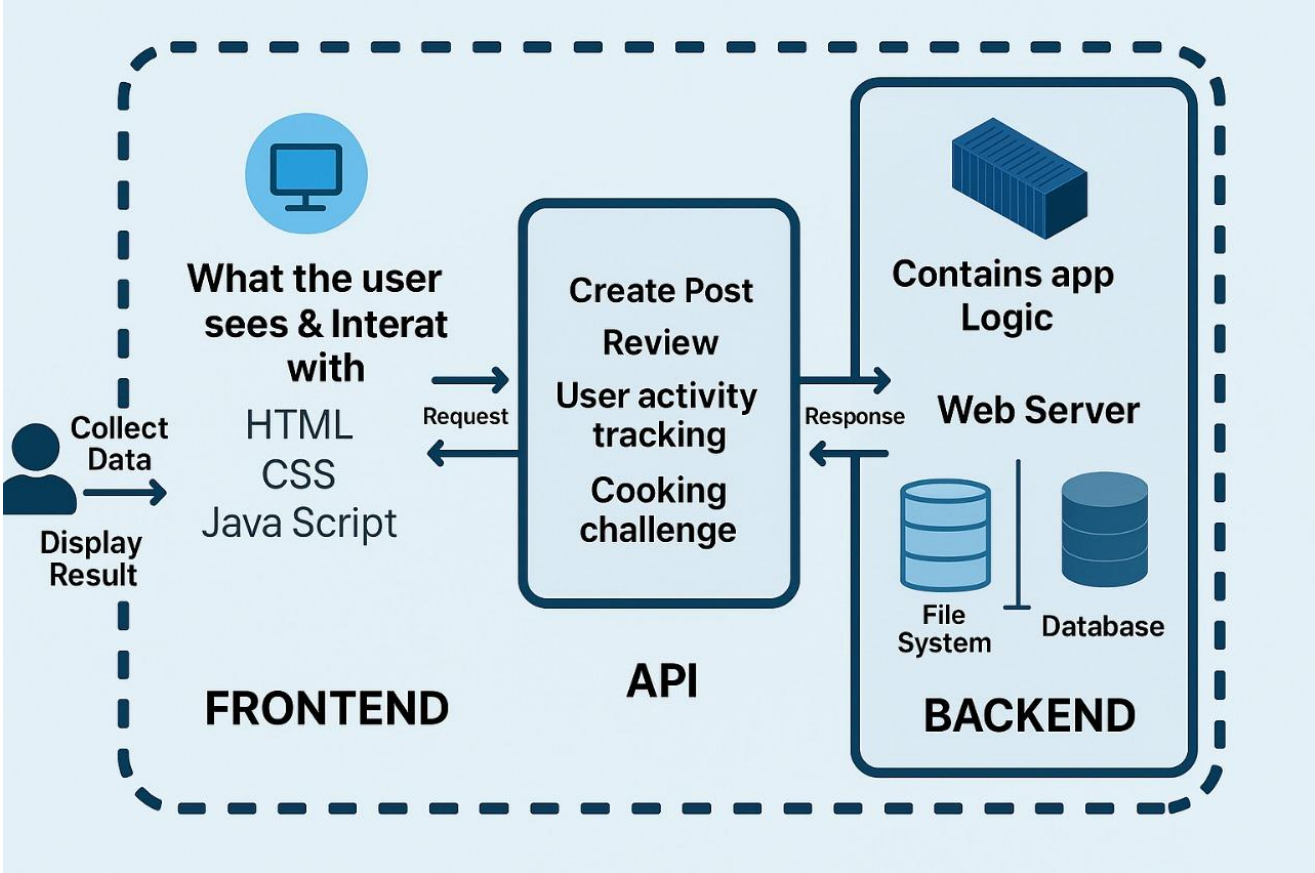
REST API Non-Functional Requirements

1. **Performance**
    - Low-latency response times (<300ms) even under high load.
2. **Scalability**
    - Horizontal and vertical scaling support to accommodate growing traffic and data volume.
3. **Security**
    - Implement industry best practices:
        - HTTPS for all communication
        - Input validation and sanitization
        - Authentication and authorization checks
        - Rate limiting and monitoring
        - Data encryption at rest and in transit
4. **Maintainability**
    - Modular code structure with clean separation of concerns.
    - Easy to extend and debug using modern frameworks and testing tools.
    - Versioned API for backward compatibility.

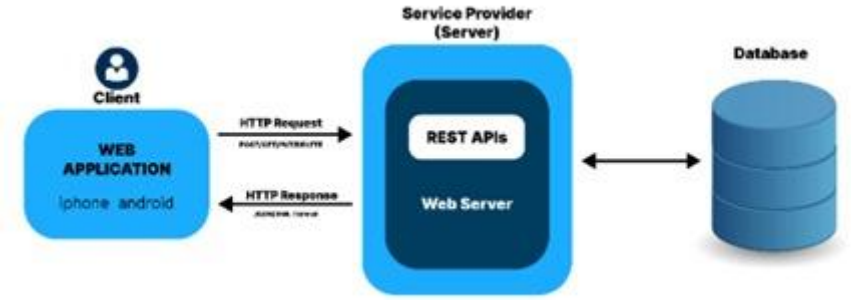
Client Web Application Non-Functional Requirements

1. **Usability**
    - Clean, responsive, and intuitive UI/UX.
    - Accessible to users of all skill levels.
2. **Privacy**
    - Users control visibility of their:
        - Reviews
        - Profile information
        - Activity feed
    - Compliance with privacy regulations (e.g., GDPR).
3. **Compatibility**
    - Cross-browser support (Chrome, Firefox, Safari, Edge).
    - Responsive design for mobile, tablet, and desktop.
4. **Performance**
    - Fast loading (first contentful paint <2 seconds).
    - Efficient asset loading with lazy loading and compression.
5. **Reliability**
    - High availability (99.9% uptime).
    - Robust error handling and user-friendly fallback pages.

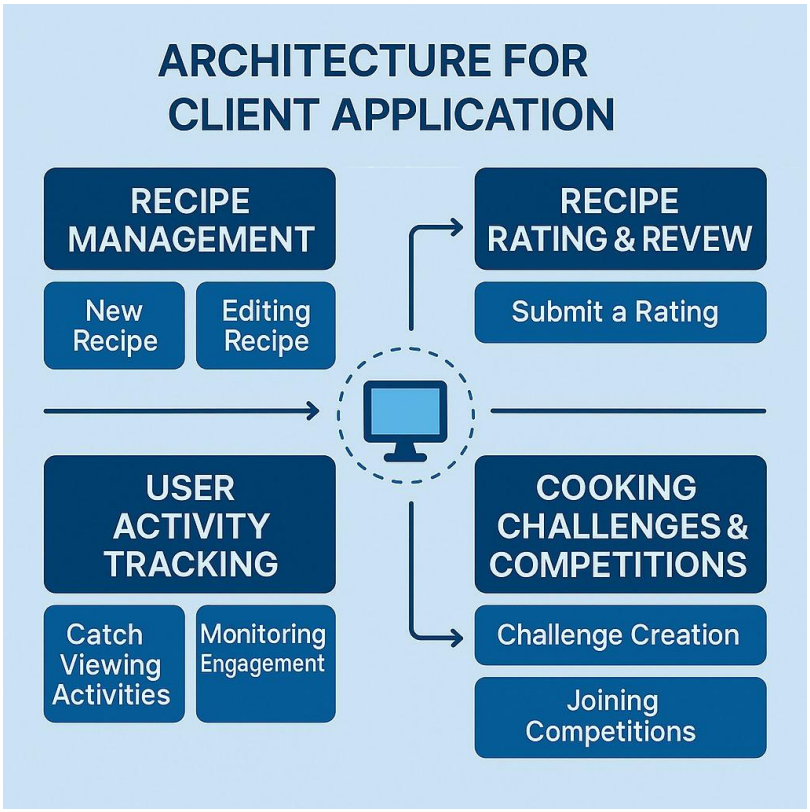
3. Overall Architecture Diagram



4. REST API Architecture diagram

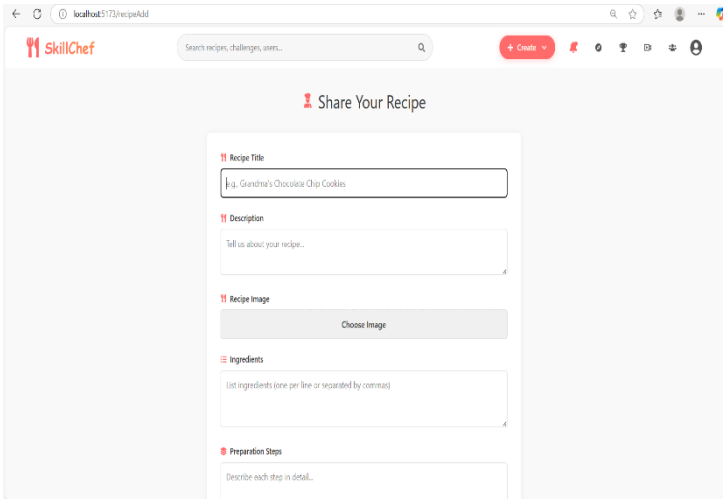


5. Frontend Architecture Diagram

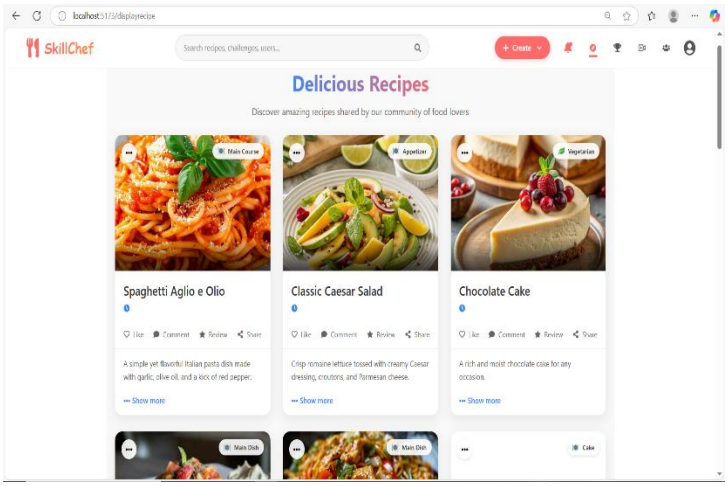


6. System Functions

1.Post Management - IT22127778 – Chavindee M.A.P



Add a Post



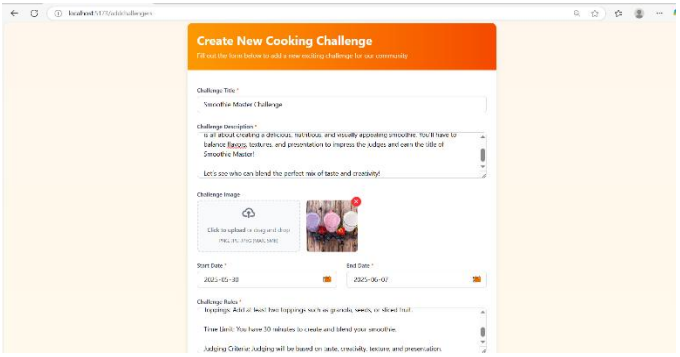
Retrieve Created Post

The *Post Management* feature allows users to share their cooking experiences, recipes, and food photos directly with the community. Users can create posts with titles, descriptions, images, and tags to express their creativity and inspire others. Other users can view, like, and comment on these posts, fostering interaction and a sense of community. This feature helps turn the app into more than just a cookbook—it becomes a space for food lovers to connect and share their passion.

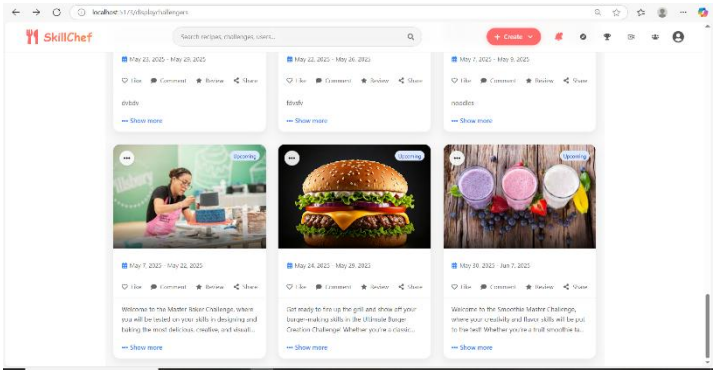
On the backend, Post Management is powered by Spring Boot and MongoDB, with secure access and operations protected by Spring Security and JWT authentication. APIs are provided for creating, reading, updating, and deleting posts, as well as managing likes and comments.

The frontend displays all posts in a clean and responsive layout, allowing users to browse by popularity or recency. Together, the frontend and backend work to provide a smooth and engaging posting experience for all users.

2.Challenge Management- IT22208576 – Fernando M.S.V



Add a Challengers



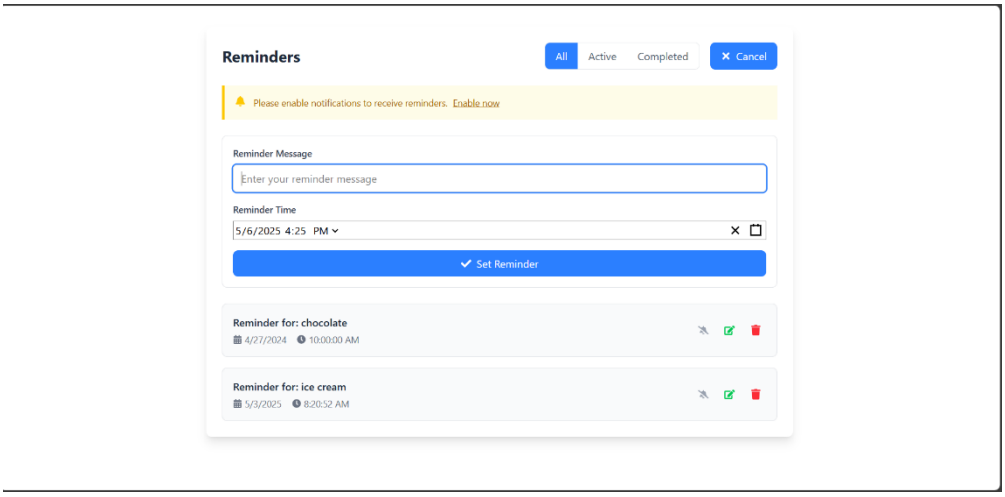
Retrieve Created Post

The *Cooking Challenges & Competition* feature is an exciting addition to our cookbook project that encourages user engagement and creativity. It allows users to view ongoing cooking challenges, read the rules and details, and participate by submitting their own recipes or photos. Each challenge is designed to inspire fun and friendly competition among cooking enthusiasts. Users can also browse through other participants' entries, react, and leave comments, making the app more interactive and community driven.

On the backend, the system is built using Spring Boot and MongoDB, with secure authentication and role-based access handled by Spring Security and JWT. Admins can create and manage challenges, while users can retrieve available challenges and submit entries through the frontend.

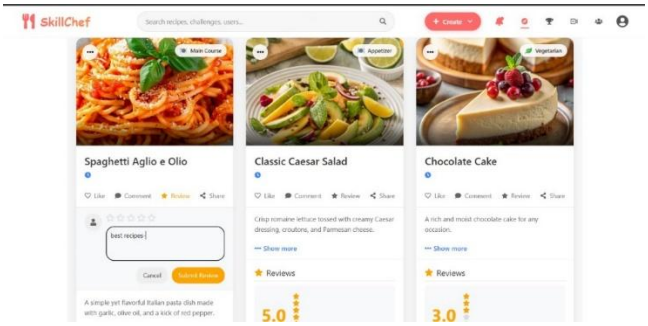
The frontend is built to be user-friendly and responsive, displaying challenges and submissions clearly across devices. Real-time notifications are also integrated to alert users when new challenges are added or when their entries receive feedback, keeping them connected and engaged.

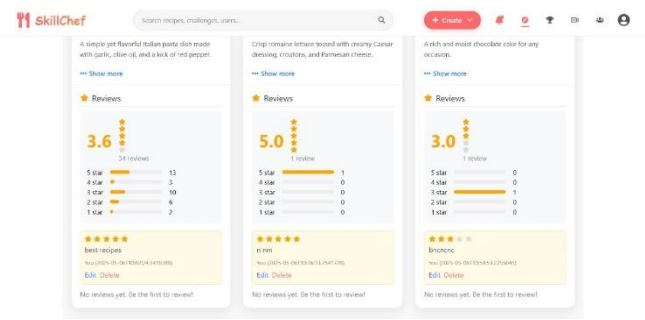
3.Post Sharing & Notification Management – IT22235688 (PERERA K.T.K)



The backend supports reading all existing posts, sharing posts using a specific ID, updating all shared information via a unique ID, and deleting shared posts with the same ID, along with viewing and removing notifications. On the frontend, users can share, edit, delete, and view previously created posts, while the notification system allows them to see likes, comments, and shared posts as alerts, with the option to delete individual notifications.

4. Recipe Rating & Review System - IT22272768 – Basnayaka W.B.M.D.K

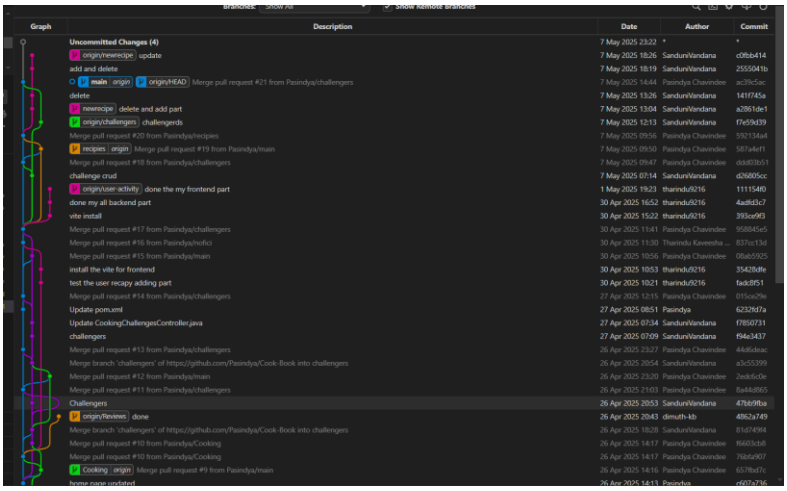


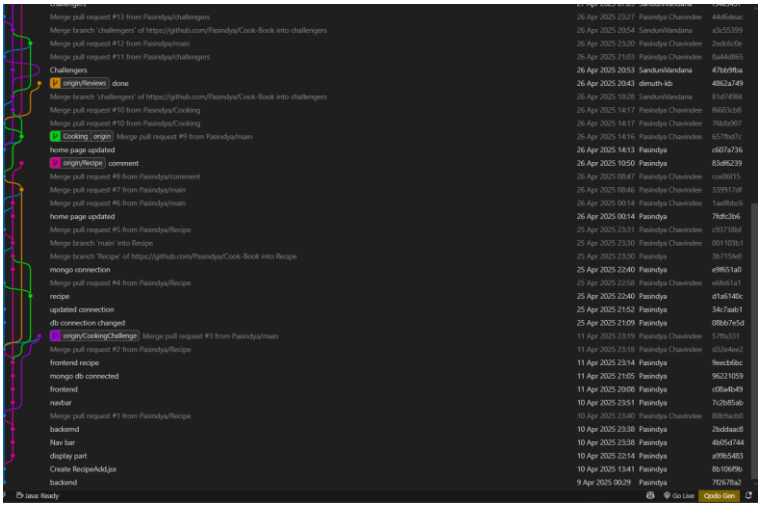


Ratings and Reviews System:
The platform allows users to rate recipes on a scale from 1 to 5 stars, enabling quick feedback and quality assessment. Each recipe displays its average rating alongside the total number of reviews it has received. Ratings are visually represented using horizontal star bars, making it easy to interpret the distribution of user feedback at a glance. Furthermore, users can sort recipes based on their average ratings, helping them quickly identify the most popular or highest-rated dishes. For example, as shown in the interface, Recipe 1 has an average rating of 3.6 from 34 reviews, with the highest counts in 5-star and 3-star ratings. Recipe 2 holds a perfect 5.0 rating from one review, while Recipe 3 has an average of 3.0 from one review. This feature not only promotes user engagement but also provides valuable insight into community preferences and recipe quality.

7. GitHub

- GitHub Repo link -https://github.com/Pasindya/Cook-Book
- Image of the GitHub Commit Tree





References

- SpringBoot You Tube

1.https://www.youtube.com/playlist?list=PLyprkB9NXq2JczKvQe2fG_7Yhy0L32reu

- MongoDB Connection

2.https://www.youtube.com/watch?v=1-M-On89KHY