

Sri Lanka Institute of Information Technology

Programming Applications and Frameworks (IT3030)

Final Assignment Report

GROUP ID: WE-164



Topic

Student ID	Student Name
Subasinghe S M U	IT22186638
Karunarathne B G T N	IT22027788
Pasanjith W A R	IT22898920
Panditharathne R.L	IT22266200

Date: 05.07.2025

Table of Contents

1. Introduction 3

2. Functional Requirements & Non-Functional Requirements 4

3. Overall Architecture Diagram..... 7

4. REST API Architecture diagram 8

5. Frontend Architecture Diagram 8

6. System Functions 9

7. GitHub..... 13

8. References 15

1. Introduction

SkillFlow: The Ultimate Skill-Sharing & Learning Platform

SkillFlow is an innovative platform designed to provide a seamless and engaging learning experience. Built with a React.js frontend for an intuitive interface and a Spring Boot backend for efficient data processing, it enables users to share knowledge and track their learning progress effortlessly.

The platform ensures secure authentication through OAuth 2.0 (Google Login), allowing users to log in safely. A MySQL database powers the system, storing user profiles, skill-sharing posts, learning plans, and community interactions.

To guarantee high performance and scalability, SkillFlow is deployed on Vercel, ensuring smooth accessibility across devices. Whether you're sharing expertise, discovering new skills, or engaging with like-minded learners, SkillFlow provides the perfect ecosystem for collaborative learning.

1. User Management System – This feature handles user authentication, profile management, and social interactions. Users can register, log in via OAuth 2.0, update their profiles, and follow/unfollow other users to build a learning community.

2. Post Management System – This allows users to share skill-related posts, including text, images, and videos. Users can create, edit, and delete posts, making it easy to share knowledge and learning experiences with others.

3. Comment & Feedback Management System – This feature enables users to engage with skill-sharing posts by adding, editing, and deleting comments. It fosters interaction, discussion, and feedback among learners and mentors.

4. Learning Plans Management System – Users can create structured learning plans to track progress on new skills. They can update milestones, add learning resources, and manage their plans to stay organized and motivated.

5. Learning Progress Update- This feature allows users to share updates on their skill development journey. Users can add titles, descriptions, tutorials followed and newly acquired skills to document their learning progress. Each update contributes to tracking personal growth and helps others get inspired by visible progress within structured learning plans.

2. Functional Requirements & Non-Functional Requirements

Functional Requirements for REST API

- User Management:
 - ✓ Register and authenticate users via OAuth 2.0 (Google, Facebook, etc.).
 - ✓ Users can create and manage their profile.
- Skill Sharing:
 - ✓ Users can upload up to 3 photos or a short video (max: 30 sec) per post.
 - ✓ Users can add descriptions to their shared content.
- Learning Plans:
 - ✓ Users can create structured learning plans with topics, resources, and timelines.
 - ✓ Plans can be updated as users progress.
- Learning Progress Update:
 - ✓ Create, update, retrieve, and delete progress updates linked to learning plans.
 - ✓ Ensure only authenticated users can manage their own progress updates.
- Interactivity & Engagement:
 - ✓ Users can like and comment on posts.
 - ✓ Comments can be edited or deleted by the user.
 - ✓ Post owners can delete comments on their own posts.
- User Profiles & Social Features:
 - ✓ Each user has a profile displaying their skill-sharing posts and activities.
 - ✓ Users can follow other users to see their posts.
 - ✓ Profiles are publicly visible.
- Notifications:
 - ✓ Users receive notifications for likes and comments on their posts.

Functional Requirements for Client Web Application

- User Authentication:
 - Allow OAuth 2.0 login (Google, Facebook, etc.).
- Dashboard:
 - Show recent posts, user progress, and trending skills.
- Post Creation:
 - Users can upload text, images, or short videos for skill-sharing.
- Learning Plan Management:
 - Users can create and update structured learning plans.
- User Profiles:
 - Display user posts, learning progress, and followers.
- Social Interactions:
 - Users can like, comment, and follow other users.
- Learning Progress Update:
 - Users can post structured updates on their learning journey.
 - Allow editing and deleting of previously submitted updates.
- Notifications:
 - Users get real-time updates for likes and comments.

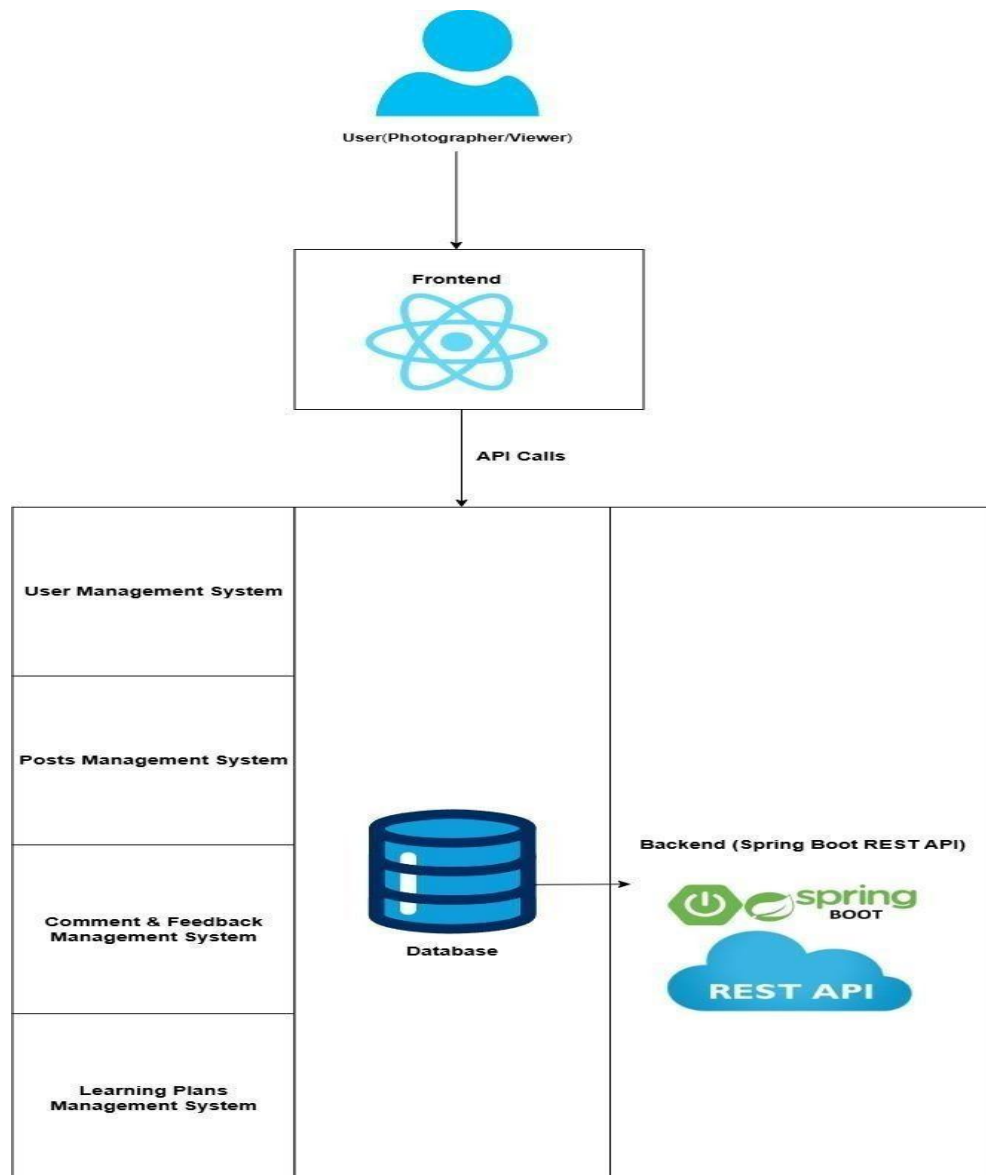
Non-Functional Requirements for REST API

- Performance
 - The API ensures fast response times for all components including user, post, comment, plan, and progress management.
- Scalability
 - Designed to handle increased traffic and a growing number of users without reducing performance.
- Security
 - OAuth 2.0 authentication, role-based access control, and input validation protect sensitive data and prevent unauthorized access.
- Maintainability
 - Each component follows modular design with clear separation of logic, making the API easy to update and extend.

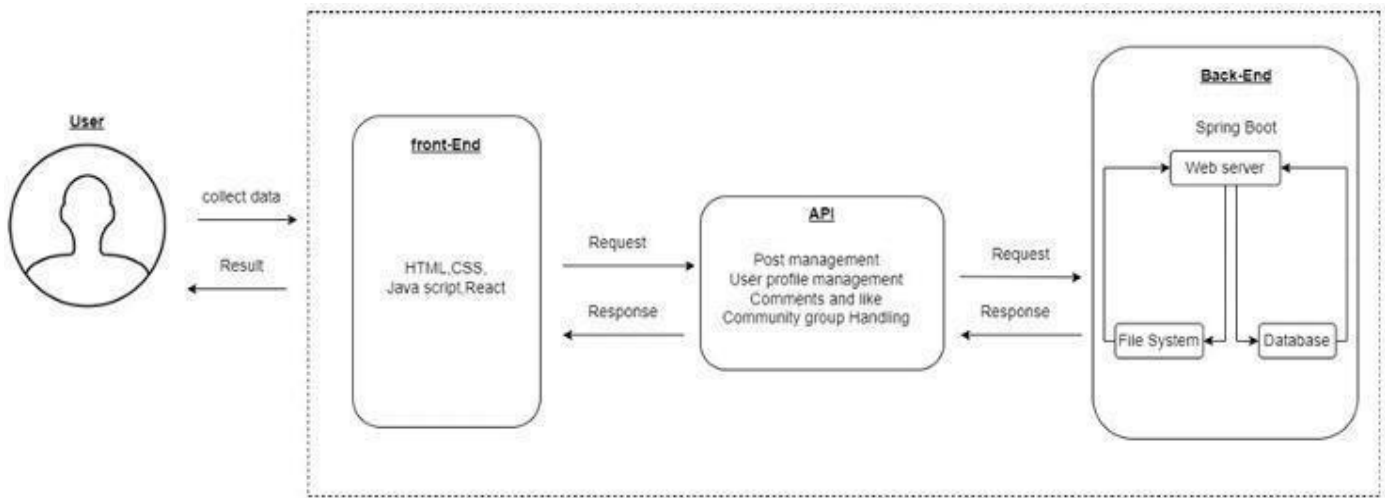
Non-Functional Requirements for Client Web Application

- Usability
 - The platform provides a clean, user-friendly interface that supports easy navigation for all users.
- Privacy
 - User profiles and shared content respect privacy settings, allowing control over visibility.
- Compatibility
 - The system works seamlessly across major browsers and both desktop and mobile devices.
- Performance
 - Client-side features are optimized for fast loading and responsive interactions.
- Reliability
 - Ensures stable performance with minimal downtime to maintain continuous user access.

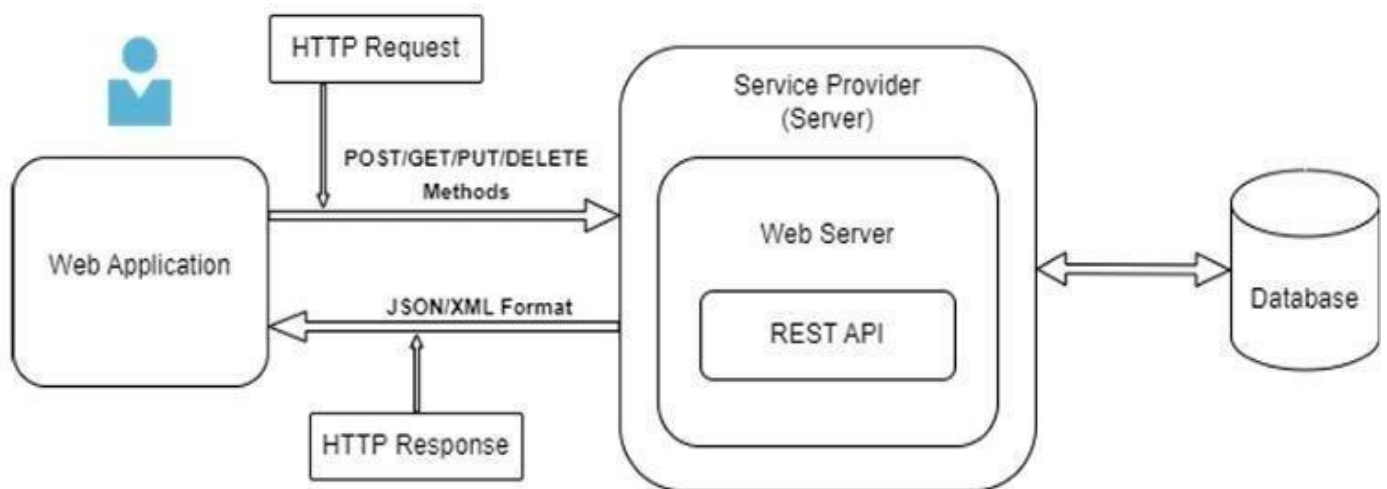
3. Overall Architecture Diagram



4. REST API Architecture diagram



5. Frontend Architecture Diagram



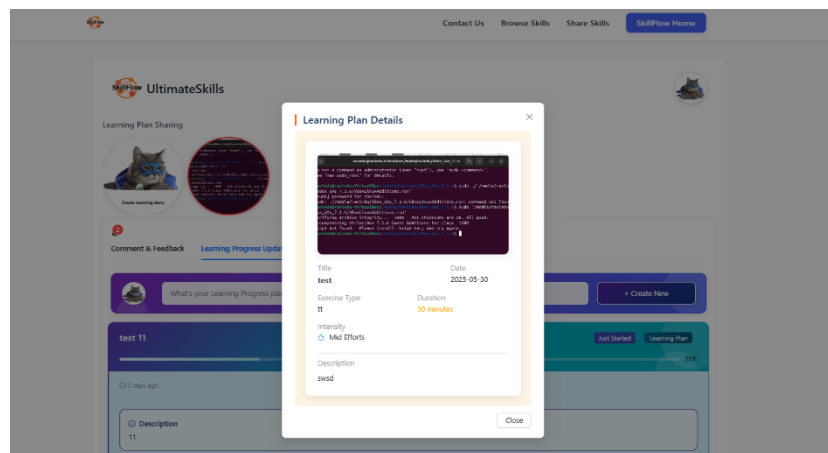
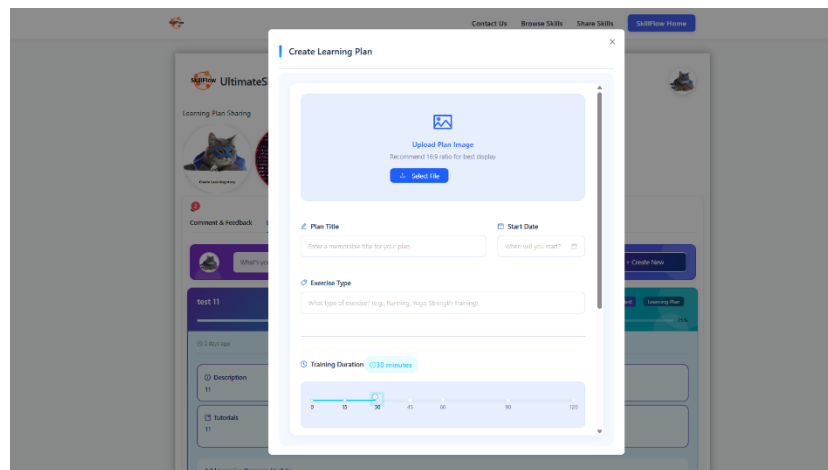
6. System Functions

Work Distribution

1. Learning Plans Component (Subasinghe S M U- IT22186638)

The Learning Plans Component helps users create and manage structured learning plans, allowing them to track their progress in skill development. Users can set topics, add learning resources, and define milestones. Learning plans can be updated as users progress, ensuring an organized approach to acquiring new skills.

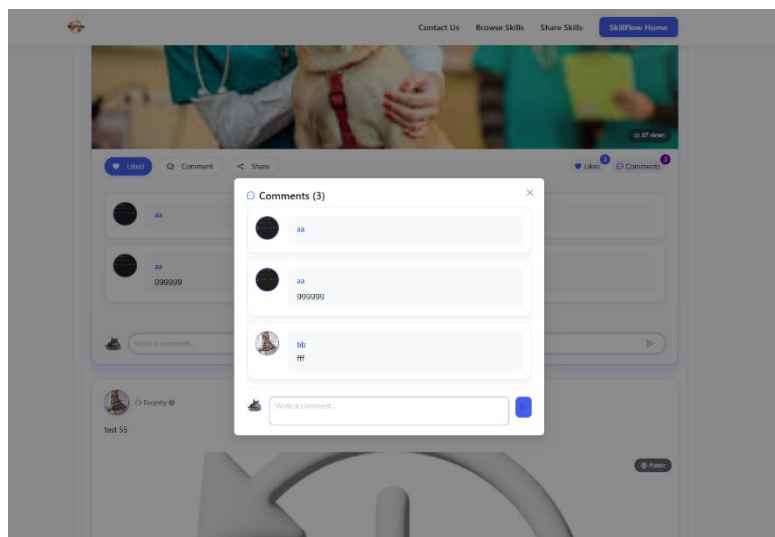
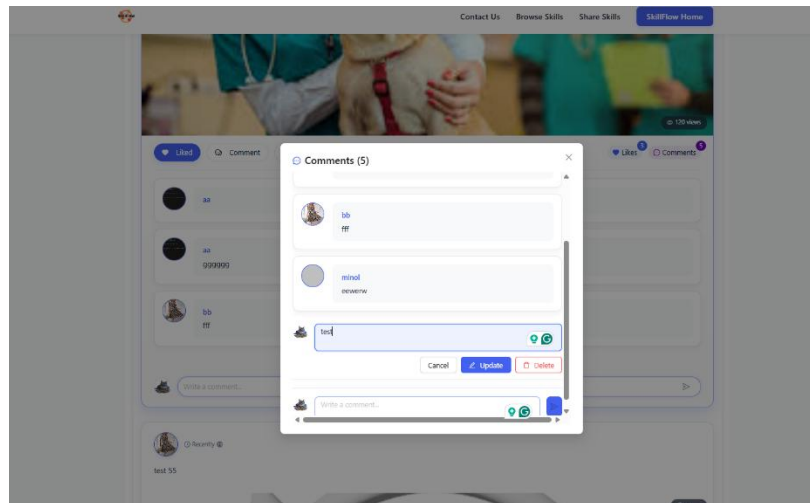
- **POST /learning-plans** → Create a new learning plan.
- **GET /learning-plans/{id}** → View a learning plan.
- **PUT /learning-plans/{id}** → Edit a learning plan.
- **DELETE /learning-plans/{id}** → Delete a learning plan.



2. Comments & Feedback Component (Panditharathne R.L - IT22266200)

The Comments & Feedback Component enables users to interact with skill-sharing posts by leaving comments, providing feedback, and engaging in discussions. Users can edit or delete their comments, and post owners have the ability to remove unwanted or inappropriate feedback. This feature fosters interaction and knowledge exchange within the platform.

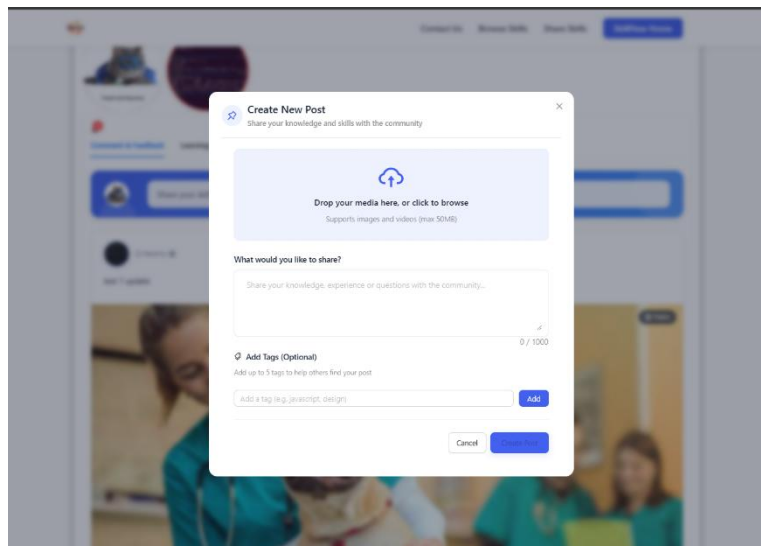
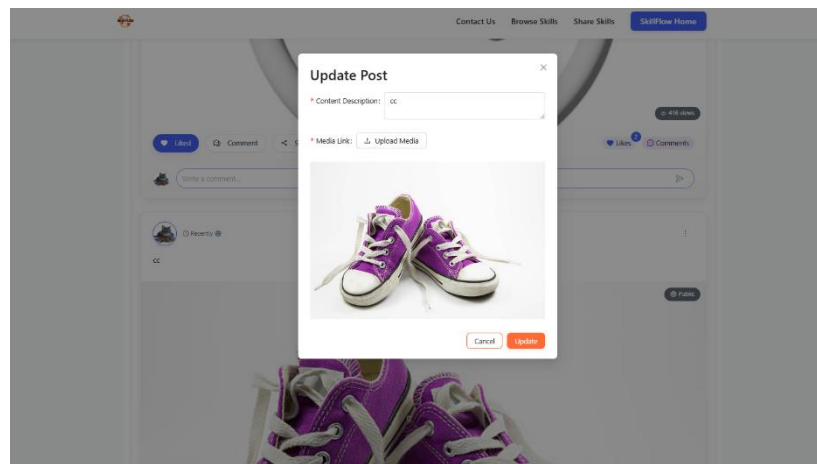
- **POST /posts/{postId}/comments** → Add a comment to a post.
- **GET /comments/{commentId}** → View a specific comment.
- **PUT /comments/{commentId}** → Edit a comment.
- **DELETE /comments/{commentId}** → Delete a comment.
- **POST /posts/{postId}/likes** → Like a post.
- **DELETE /posts/{postId}/likes** → Unlike a post.
- **POST/notifications** → Create a new notification when a user likes or comments on a post.



3. Skill Sharing Posts Component (Pasanjith W A R - IT22898920)

The Skill Sharing Posts Component allows users to create, edit, delete, and view skill-related posts. Users can upload text, images, or short videos (up to 30 seconds) to share knowledge in areas such as coding, photography, or DIY crafts. This component enables easy content sharing, helping users engage and contribute to the learning community.

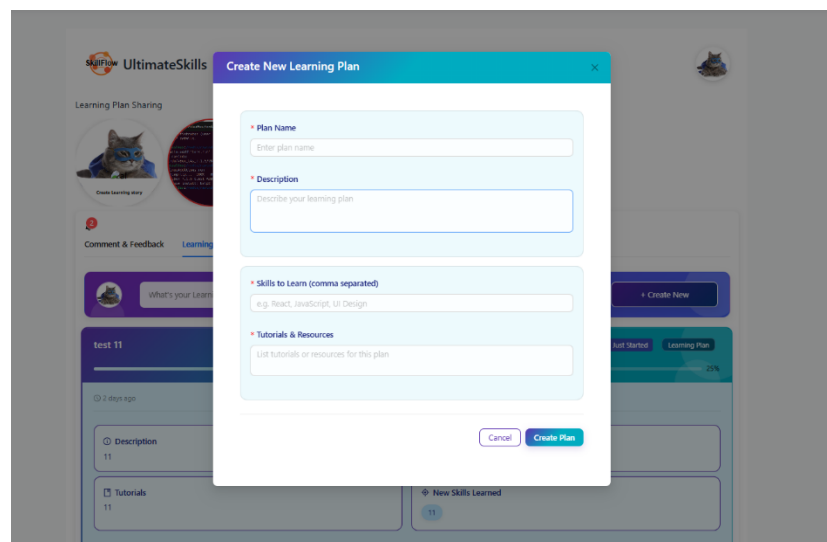
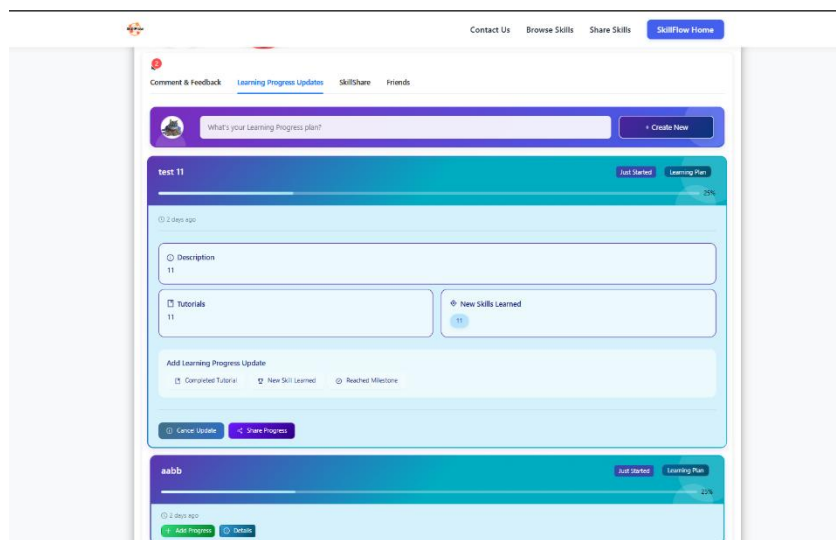
- **POST /posts** → Create a new post.
- **GET /posts/{id}** → Retrieve a specific post.
- **PUT /posts/{id}** → Update post details (title, description, media).
- **DELETE /posts/{id}** → Remove a post.



4. Learning Progress Update (Karunarathne B G T N- IT22027788)

The Learning Progress Update Component allows users to document and share updates related to their learning journey. These updates help users reflect on milestones, stay motivated, and maintain consistency in following their learning plans. Structured templates support organized input, and all updates are visible in a timeline format under each learning plan.

- **POST** /learning-progress → Create a new progress update.
- **GET** /learning-progress/{id} → View a specific progress update.
- **GET** /learning-plans/{id}/progress → View all progress updates for a specific learning plan.
- **PUT** /learning-progress/{id} → Edit an existing progress update.
- **DELETE** /learning-progress/{id} → Delete a progress update.

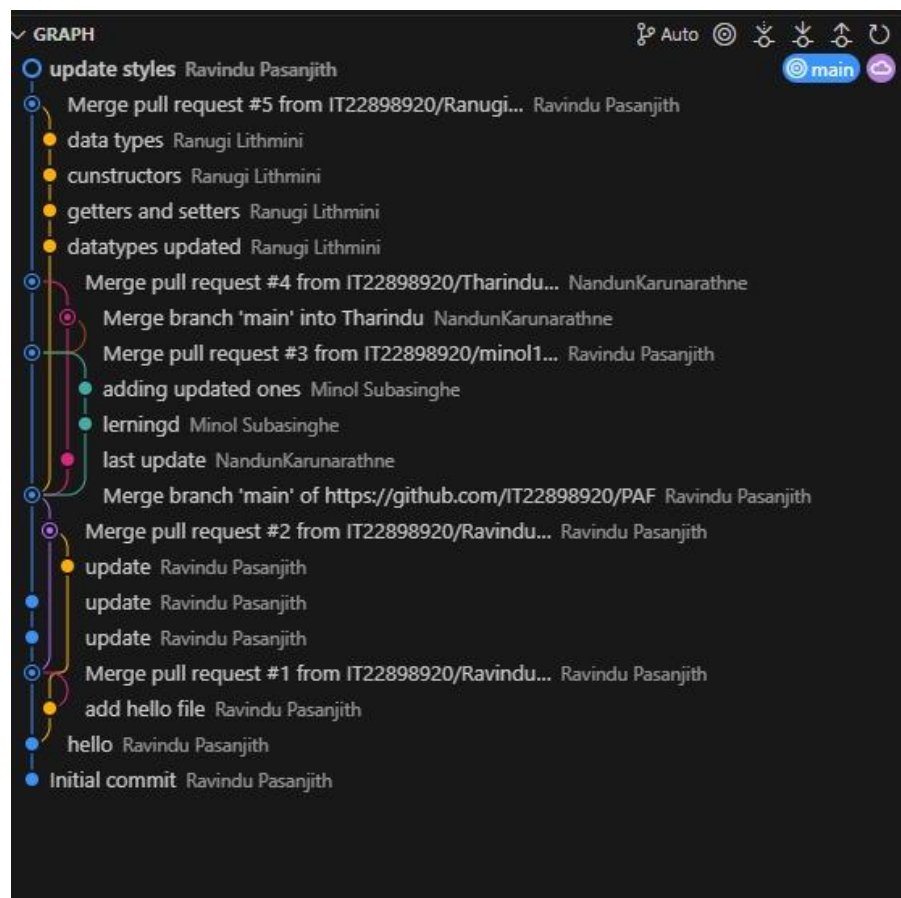
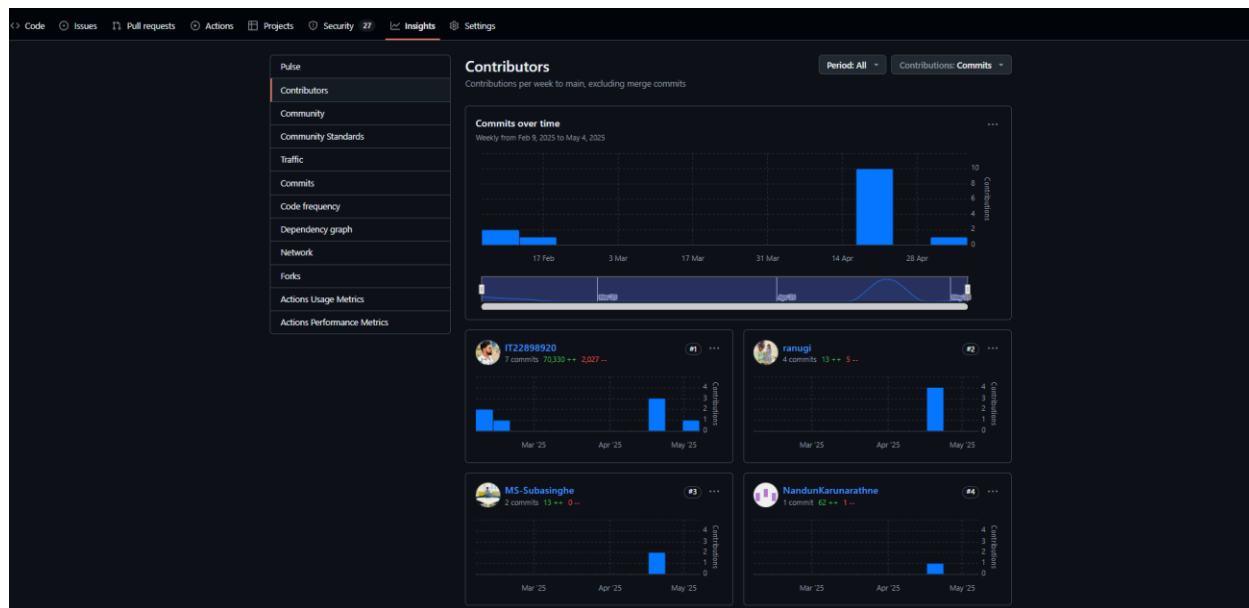


7. GitHub

- GitHub Repo link

<https://github.com/IT22898920/PAF>

- Image of the GitHub Commit Tree



8. References

- 9.
10. [1] R. Johnson, "RESTful Web Services: The basics," IBM Developer, 2019. [Online]. Available: <https://developer.ibm.com/articles/ws-restful/>
- 11.
12. [2] M. Fowler, "Microservices: A definition of this new architectural term," martinfowler.com, 2014. [Online]. Available: <https://martinfowler.com/articles/microservices.html>
- 13.
14. [3] A. Gupta and S. Arora, "Secure authentication with OAuth 2.0," International Journal of Computer Applications, vol. 179, no. 3, pp. 1–5, 2017. doi: 10.5120/ijca2017915484.
- 15.
16. [4] Facebook Inc., "React – A JavaScript library for building user interfaces," 2023. [Online]. Available: <https://reactjs.org/>
- 17.
18. [5] P. Deore and A. Varade, "Design and Development of Learning Management System Using Spring Boot and React," International Journal of Computer Sciences and Engineering, vol. 10, no. 5, pp. 135–140, 2022.
- 19.
20. [6] OWASP Foundation, "Top 10 Web Application Security Risks," 2023. [Online]. Available: <https://owasp.org/www-project-top-ten/>
- 21.
22. [7] J. Smith and L. Brown, "Responsive Web Design Principles," IEEE Internet Computing, vol. 22, no. 4, pp. 88–92, Jul.–Aug. 2020. doi: 10.1109/MIC.2020.2996584.
- 23.
24. [8] B. Kitchenham, "Procedures for Performing Systematic Reviews," Keele University Technical Report, TR/SE-0401, 2004.
- 25.
26. [9] GitHub, "PAF_LMS_Group_34," GitHub repository, 2025. [Online]. Available: https://github.com/SandaruwanChandrasena/PAF_LMS_Group_34.git