Sri Lanka Institute of Information Technology

Programming Applications and Frameworks (IT3030)

Continuous Assignment – 2025, Semester 1

Initial Document

GROUP ID: Y3.S2.IT.WE.139

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

Contents

Project Description

The platform is designed to provide a seamless **user experience** using a **React.js frontend** for an interactive interface and a **Spring Boot backend** for efficient data processing. It ensures **secure authentication** through **OAuth 2.0 (Google Login)**, allowing users to log in safely. The database, powered by **PostgreSQL**, stores user profiles, posts, learning plans, and interactions. The application is deployed with **Heroku/Render** for the backend and **Vercel/Netlify** for the frontend, ensuring scalability and performance.

1. **Search and Favorite Plans + User Profile –** Implements the plan discovery module, enabling users to search shared learning plans and manage a personalized list of favorite plans for easy access. Develops the profile module, allowing public profiles.

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.

Functional Requirements

REST API

- Search and Favorite Plans + User Profile:

✓Users can search for shared learning plans using filters such as category, popularity, and author.
✓Users can add learning plans to their list of favorite plans for quick access and management.
✓Users can create and manage their public profile, including profile picture, bio, and shared plans.
✓Users can view other public profiles to explore shared learning plans and follow their activity.

- 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 Progress Tracking:

  ✓Users can post updates on their learning journey.
  ✓Provide predefined templates for structured progress updates.

- Learning Plans:

  ✓Users can create structured learning plans with topics, resources, and timelines.
  ✓Plans can be updated as users progress.

- 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.

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.
- Notifications:
  - Users get real-time updates for **likes and comments**.

Non-Functional Requirements

- Security:

- ✓ Implement OAuth 2.0 authentication.
- ✓ Use JWT (JSON Web Token) for session management.
- ✓ Secure API endpoints with Spring Security.

- Scalability:
  - ✓ Design API using RESTful principles to ensure scalability.
  - ✓ Use load balancing and caching mechanisms.
- Performance:
  - ✓ Optimize database queries to improve response time.
  - ✓ Implement pagination to handle large datasets.
- Maintainability:
  - ✓ Follow clean coding principles and modular architecture.

Member's contribution

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 /comments** → Add a comment to a post.
- **GET /comments/{id}** → View a comment.
- **PUT /comments/{id}** → Edit a comment.
- **DELETE /comments/{id}** → Delete a comment.

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. Search and Favorite Plans + User Profile (Karunarathne B G T N IT22032492)

- Implements the plan discovery module, enabling users to search shared learning plans and manage a personalized list of favorite plans for easy access. Develops the profile module, allowing public profiles.

- **POST /users** → Register a new user.
- **GET /users/{id}** → Retrieve user profile.
- **PUT /users/{id}** → Update user profile details.
- **DELETE /users/{id}** → Delete a user account.

- **POST / favorite** → Create a new public learning plan to the list of favorite plans.
- **GET / favorite /{id}** → View favorite plans.
- **PUT / favorite /{id}** → Update a favorite plan.
- **DELETE / favorite /{id}** → Delete a favorite plan.

System Overview Diagram

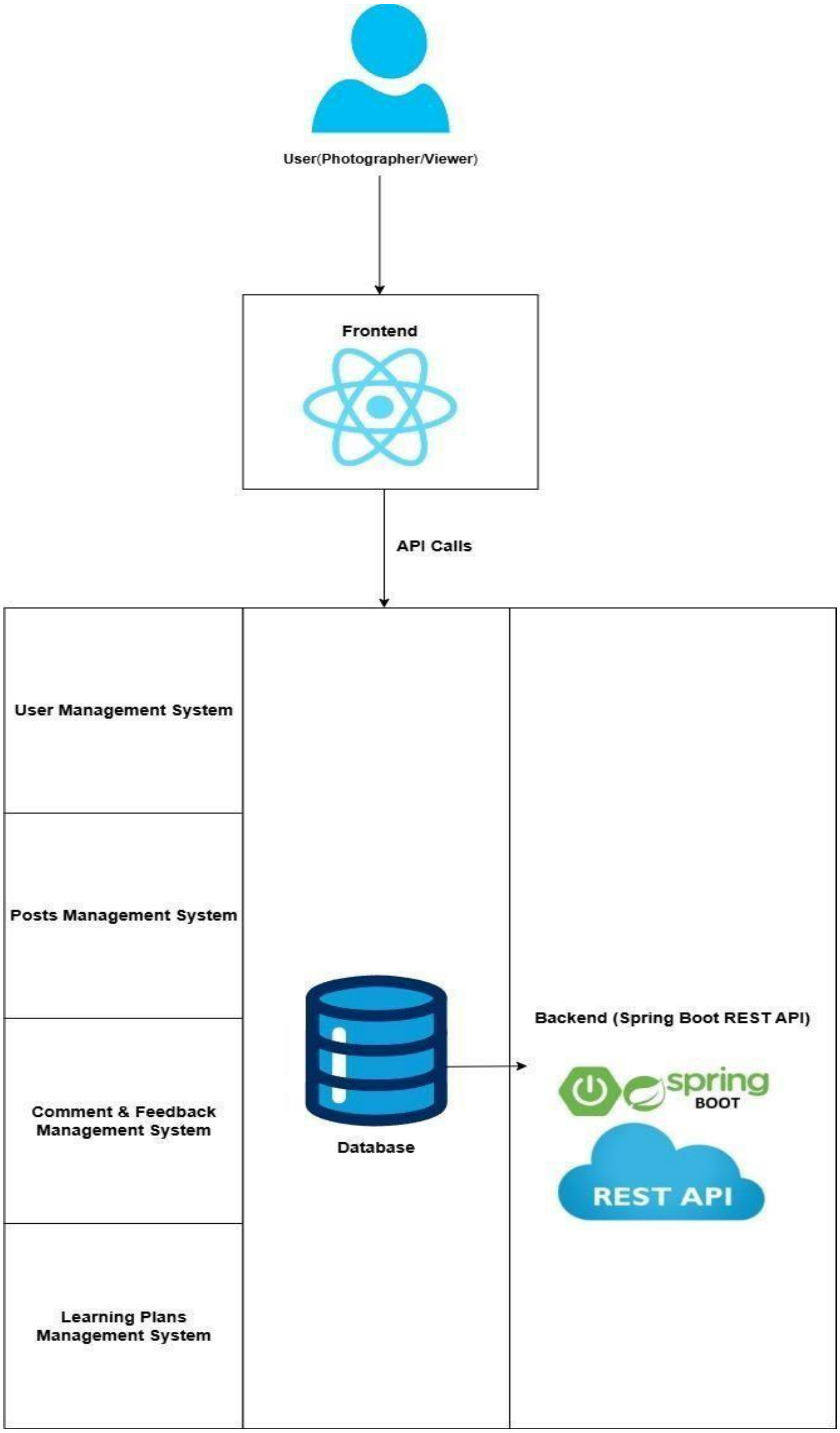
User(Photographer/Viewer)
Frontend
API Calls
User Management System
Posts Management System
Comment & Feedback Management System
Learning Plans Management System
Database
Backend (Spring Boot REST API)
spring BOOT
REST API

Figure 1- System Overview Diagram

Architectural Diagram for REST API

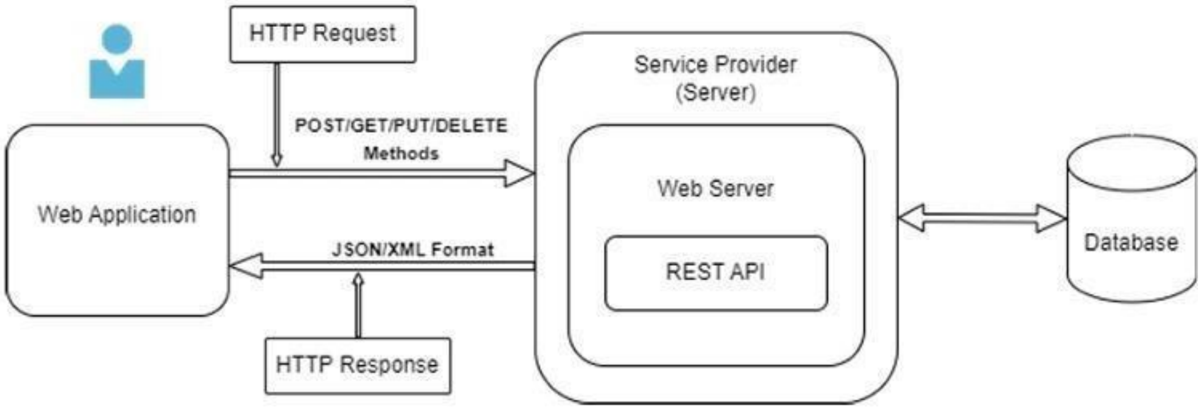

Figure 2 - Front End Diagram

Architectural Diagram for Front End

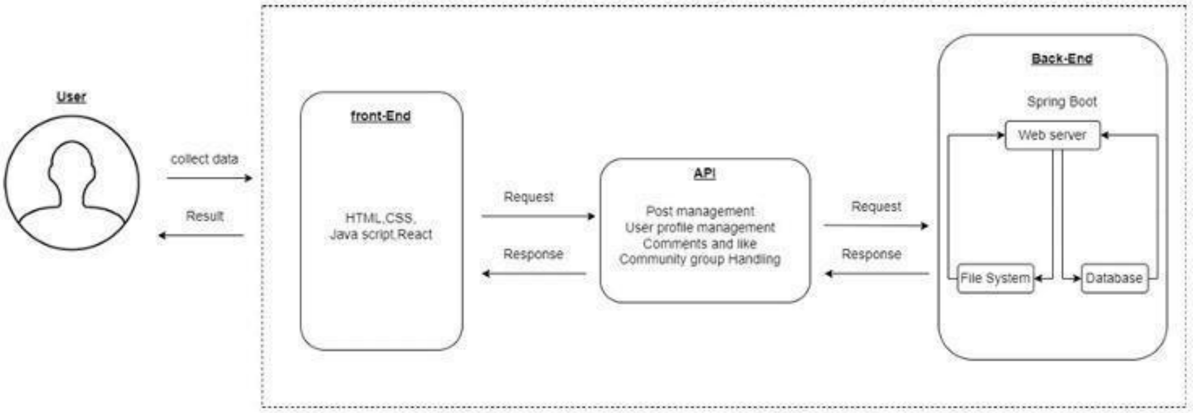

Figure 3 - REST API Diagram

Gantt Chart

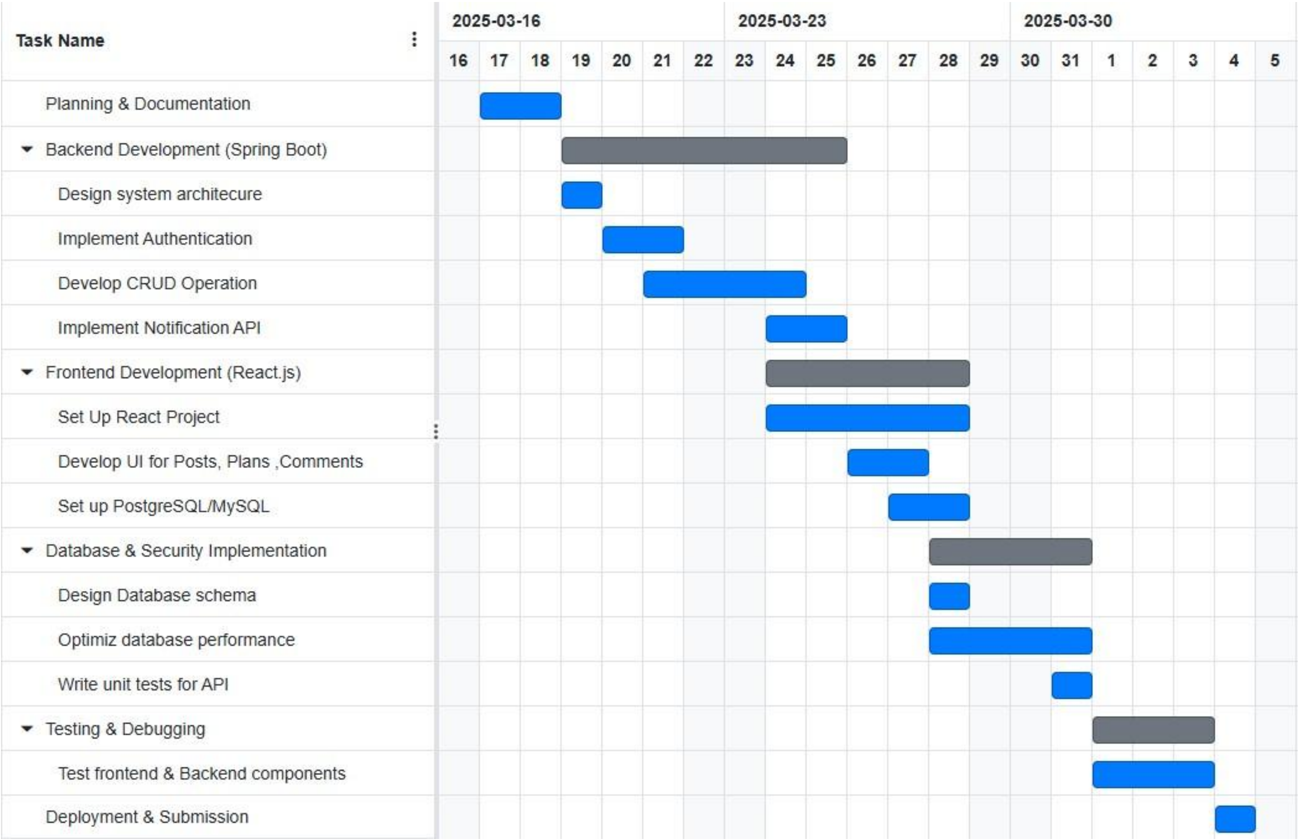

Figure 4- Gantt Chart

References

- **How to design REST APIs** – https://www.redhat.com/en/topics/api/what-is-a-rest-api
- **Spring Security & OAuth 2.0** – https://spring.io/projects/spring-security
- **React & REST API integration** – https://reactjs.org/docs/faq-ajax.html