**Sri Lanka Institute of Information Technology**

**Programming Applications & Frameworks – IT3030**

# **Assignment – 2025, Semester 1**

**Skill Sharing & Learning Platform – Initial Document**



**Group ID: Y3S1-WD-50 Group Members:**

|  |  |  |
| --- | --- | --- |
| **Student ID** | **Student Name** | **Function** |
| IT22088314 | MALSHANI W.A.M.S | User Management |
| IT22549204 | HARISCHANDRA R..D.I.D | Post Management |
| IT22248176 | HERATH H.M.T.N | Comment & Feedback Management |
| IT22643636 | HUSSAINIYA M.H.F | Learning Plan Management |

**Table of Contents**

[**Assignment – 2025, Semester 1** 1](#_Toc193221784)

[**Project Description** 3](#_Toc193221785)

[**Overall Architecture Diagram for Entire System** 4](#_Toc193221786)

[**Detailed Architecture Diagram(s) for the REST API** 5](#_Toc193221787)

[**Detailed Architecture Diagram(s) for the Client Web Application** 6](#_Toc193221788)

[**Functional Requirements** 7](#_Toc193221789)

[**• REST API (Backend – Spring Boot)** 7](#_Toc193221790)

[**Non-Functional Requirements** 9](#_Toc193221791)

[**• Scalability** 9](#_Toc193221792)

[**• Performance** 9](#_Toc193221793)

[**• Maintainability** 9](#_Toc193221794)

[**Work Breakdown Structure (WBS)** 10](#_Toc193221795)

[• **Frontend Development (React UI)** 10](#_Toc193221796)

[**References** 12](#_Toc193221797)

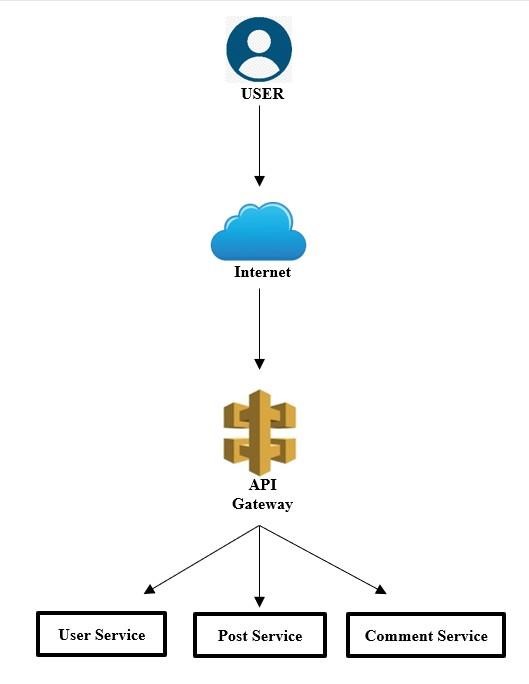
# **Project Description**

The **Skill-Sharing & Learning Platform** is an interactive web application designed to connect individuals who want to share and learn various skills such as coding, cooking, photography, and DIY crafts. Users can track their learning progress, share and manage skillsharing postings, and build structured learning plans using the site. The system aims to provide a community-driven learning environment where people can encourage and assist one another through an intuitive UI and social engagement elements.

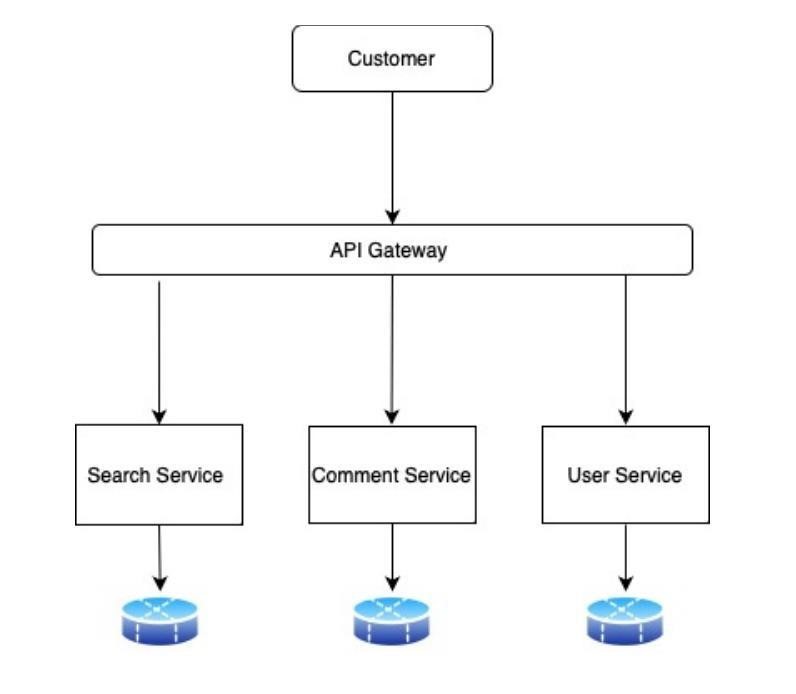
The platform consists of several main functionalities, including skill-sharing posts where users can upload images and short videos, learning progress updates to track personal achievements, and structured learning plans to help users organize their learning journey. Additionally, users can communicate with one another by liking postings, leaving comments, and following other students thanks to user profiles and social features. This ensures a lively and engaging experience that promotes cooperation and ongoing education.

Built using **Spring Boot for the backend and React for the frontend**, the platform ensures high performance, scalability, and maintainability. It integrates **OAuth 2.0 authentication** for secure login and follows **RESTful API principles** for seamless communication between the backend and frontend. The Skill-Sharing & Learning Platform, which makes use of modern web technologies, offers a dependable and effective way for those who want to learn more and share their knowledge with others.

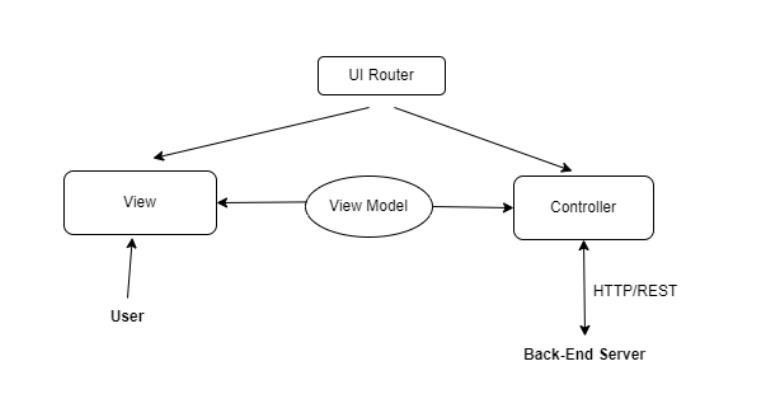
# **Overall Architecture Diagram for Entire System**



# **Detailed Architecture Diagram(s) for the REST API**



# **Detailed Architecture Diagram(s) for the Client Web Application**



# **Functional Requirements**

## **• REST API (Backend – Spring Boot)**

1. **User Management:**

* Register and authenticate users via OAuth 2.0
* Users can create and manage their profile

1. **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

1. **Learning Progress Tracking:**

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

1. **Learning Plans:**

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

1. **Interactivity & Engagement:**

* Users can like and comment posts
* Comments can be edited or deleted by user
* Post owners can delete comments on their own posts

1. **User Profile & 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

1. **Notifications:**

* Users receive notifications for likes and comments on their posts

**Client Web Application (Frontend - React)**

1. **User Authentication:**

* Allow **OAuth 2.0 login**

1. **Dashboard:**

* Show **recent posts, user progress, and trending skills**

1. **Post Creation:**

* Users can upload **text, images, or short videos** for skill-sharing

1. **Learning Plan Management:**

* Users can create and update **structured learning plans**

1. **User Profiles:**

* Display user posts, learning progress, and followers

1. **Social Interactions:**

* Users can like, comment, and follow other users

1. **Notifications:**

* Users get real-time updates for **likes and comments**

# **Non-Functional Requirements**

• **Security**

* OAuth 2.0 authentication for user login4
* Encrypted password storage using Spring Security
* API rate limiting to prevent misuse

## **• Scalability**

* Efficient database schema to support high user activity
* Load balancing strategies for handling multiple requests

## **• Performance**

* + - * API response time should be under 500ms
      * Lazy loading for images/videos to optimize frontend performance

## **• Maintainability**

* React and Spring boot both use a modular programming framework
* Swagger-based proper API documentation

• **User Experience**

* Responsive UI design

# **Work Breakdown Structure (WBS)**

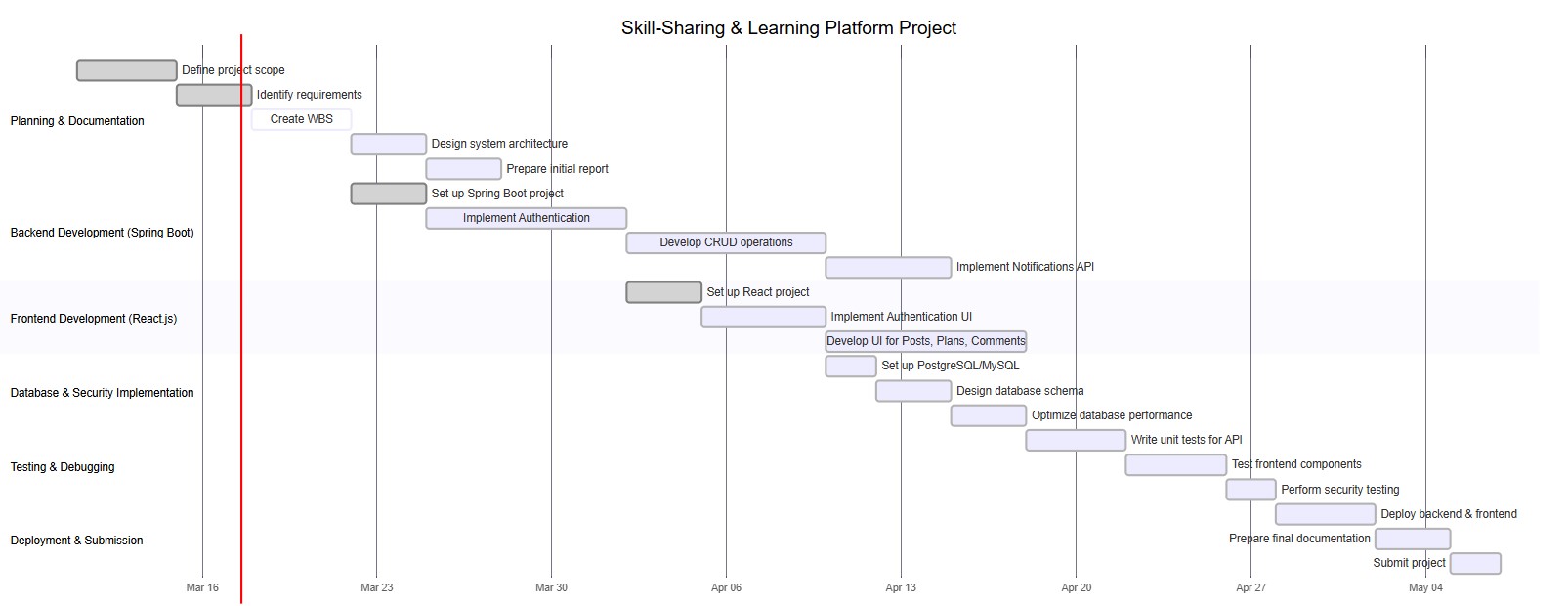
• **Backend Development (Spring Boot API)**

* Set up Spring Boot project structure
* Implement CRUD APIs for every feature
* Integrate authorization and authentication (OAuth 2.0)
* Establish a connection to the database (MongoDB)
* Unit testing and Postman are used for API testing

## • **Frontend Development (React UI)**

* + - Set up React project with proper component structure
    - Design UI components for each feature
    - Integrate API calls with the frontend
    - Ensure responsive UI and smooth navigation

**Gantt Chart**



# **References**

[https://stackoverflow.com/questions/34367031/functional-and-non-](https://stackoverflow.com/questions/34367031/functional-and-non-functional-requirements-relating-to-web-api)

[functionalrequirements-relating-to-web-api](https://stackoverflow.com/questions/34367031/functional-and-non-functional-requirements-relating-to-web-api)

[https://ahmetozlu.medium.com/mastering-rest-architecture-restarchitecturedetails-e47ec659f6bc](https://ahmetozlu.medium.com/mastering-rest-architecture-rest-architecture-details-e47ec659f6bc)

<https://www.wallarm.com/what/the-concept-of-an-api-gateway>

[https://stackoverflow.com/questions/715063/what-is-the-difference-between-](https://stackoverflow.com/questions/715063/what-is-the-difference-between-a-web-application-and-a-client-server-application)

[aweb-application-and-a-client-server-application](https://stackoverflow.com/questions/715063/what-is-the-difference-between-a-web-application-and-a-client-server-application)