

**POSTS AND TELECOMMUNICATIONS
INSTITUTE OF TECHNOLOGY**



**FINAL REPORT
OF BASIC INTERNSHIP**

Topic: Social Media Website

Lecturer: Kim Ngọc Bách

Subject: Basic Internship

Student: Nguyễn Đức Anh Tú

Student ID: B21DCVT442

Hà Nội – 2024

Table of Contents

| | | |
|------|--------------------------------------|----|
| I. | Overview | 3 |
| 1. | Reason for Choosing the Topic | 3 |
| 2. | Research Subjects..... | 3 |
| 3. | Research Objectives | 4 |
| II. | Technologies and Languages Used..... | 4 |
| 1. | HTML, CSS, JavaScript..... | 4 |
| 2. | React.js | 4 |
| 3. | Redux..... | 5 |
| 4. | MUI (Material-UI) | 5 |
| 5. | Tailwind CSS..... | 5 |
| 6. | Formik | 5 |
| 7. | Java..... | 6 |
| 8. | Spring Framework..... | 6 |
| 9. | Spring Security..... | 6 |
| 10. | Spring Data JPA | 7 |
| 11. | MySQL..... | 7 |
| III. | Project Management Report | 7 |
| 1. | Requirements Analysis..... | 7 |
| 2. | System Analysis And Design | 10 |
| 2.1. | Overview Design..... | 10 |
| 2.2. | Entity Class Diagram | 11 |
| 2.3. | Details Design | 13 |
| 2.4. | Testing and Quality Assurance..... | 17 |
| 3. | Project Schedule..... | 20 |
| 4. | Results | 20 |
| 4.1. | Completed Features | 20 |
| 4.2. | Incomplete Features | 21 |
| 4.3. | Conclusion..... | 22 |

| | | |
|------|-------------------------------------|----|
| 4.4. | Ongoing Development Challenges..... | 23 |
| 4.5. | Future Development Directions | 23 |
| 4.6. | Summary | 23 |
| IV. | Appendix..... | 24 |
| V. | References..... | 29 |

I. Overview

Website Name: Social Media Website

1. Reason for Choosing the Topic

- In the current era of rapid technological advancement, information technology has emerged as a leading industry, playing a pivotal role in the overall development of societies worldwide. Its applications extend across various domains, including scientific research, healthcare, education, and everyday life, fundamentally transforming how we interact with the world. Among these advancements, web technology has gained significant traction, rapidly evolving to meet the needs of users and organizations alike. The benefits of web technology are immense, offering solutions that enhance accessibility, streamline communication, and foster collaboration within communities. As a result, the growing reliance on digital platforms has reshaped social interactions, enabling individuals to connect, collaborate, and share information in unprecedented ways.
- By choosing to focus on the topic of "Building a Social Media Website," I aim to explore and provide an effective solution for creating a dynamic online community platform that caters to the needs of modern users. A social media website presents numerous advantages, such as facilitating global connectivity and enhancing communication among individuals from diverse backgrounds. It serves as a powerful platform for sharing information, ideas, and experiences, allowing users to engage in meaningful discussions and build relationships. This interconnectedness fosters a sense of community and collaboration, empowering users to contribute to collective knowledge and support one another. Ultimately, my goal is to develop a robust social media that not only meets the functional requirements of its users but also enriches their online experience, promoting a vibrant and inclusive digital community.

2. Research Subjects

- Social media websites, programming languages, and frameworks in web development.

3. Research Objectives

- Develop an online social media platform.
- Create a user-friendly and interactive interface.
- Enhance user engagement and connectivity.

II. Technologies and Languages Used

In this project, I utilized various technologies and frameworks to build the social media website. The technologies include HTML, CSS, JavaScript, React.js, Redux, MUI, Tailwind CSS, Formik for the front-end, and Java, Spring Framework, Spring Boot, Spring Security, Spring Data JPA, and MySQL Database for the back-end. This combination provides a robust and scalable solution for developing a full-featured social media application.

1. HTML, CSS, JavaScript

- HTML is the foundational markup language used to create the structure of web pages. It defines elements such as headings, paragraphs, links, and multimedia items, enabling browsers to render content correctly. HTML5, the latest version, introduces new semantic elements like <article>, <section>, and <nav> to improve accessibility and SEO.
- CSS is a stylesheet language used to describe the presentation of a document written in HTML or XML. It controls the layout, colors, fonts, and overall visual aesthetics of a webpage. CSS3, the latest standard, includes advanced features such as media queries for responsive design, animations, and grid layouts for complex designs.
- JavaScript is a versatile programming language that adds interactivity and dynamic behavior to web pages. It allows developers to create responsive interfaces, handle events, and perform asynchronous operations using AJAX.

2. React.js

- React.js is a JavaScript library developed by Facebook for building user interfaces, particularly single-page applications. It allows

- developers to create reusable UI components, promoting code reusability and efficiency.
- React's virtual DOM improves performance by minimizing direct DOM manipulation, leading to faster rendering. Its component-based architecture encourages a modular approach to web development, making code easier to manage and scale.

3. Redux

- Redux is a predictable state container for JavaScript applications, often used with React. It centralizes application state into a single store, making state management more predictable and easier to debug.
- Redux's unidirectional data flow ensures that changes to the state are traceable and manageable, which is particularly beneficial in large and complex applications.

4. MUI (Material-UI)

- MUI is a popular React UI framework that implements Google's Material Design. It provides a comprehensive set of components that help developers build aesthetically pleasing and consistent user interfaces quickly.
- MUI's components are highly customizable, allowing developers to tailor the design to match specific branding requirements while maintaining a modern look and feel.

5. Tailwind CSS

- Tailwind CSS is a utility-first CSS framework that provides low-level utility classes to build custom designs without leaving the HTML. It promotes a different approach to styling, focusing on composability and reusability.
- Tailwind CSS simplifies the styling process by providing a comprehensive set of classes for common design patterns, reducing the need for custom CSS. It also supports responsive design and theming, making it a versatile tool for modern web development.

6. Formik

- Formik is a popular open-source library for managing forms in React applications. It simplifies the process of handling form state, validation, and submission, making it easier for developers to create user-friendly interfaces. Formik allows for efficient management of form inputs and provides built-in support for validation, which can be customized using various libraries such as Yup.
- By utilizing Formik, developers can reduce boilerplate code and enhance the overall user experience. The library promotes a clear structure for managing form data, enabling features like dynamic form fields and real-time validation feedback. This results in more responsive applications that can effectively handle user input while maintaining data integrity.

7. Java

- Java is a robust, object-oriented programming language widely used for server-side development. Its platform independence, achieved through the Java Virtual Machine (JVM), allows Java applications to run on any device with a JVM.
- Java's strong typing and extensive standard library make it a preferred choice for building secure, scalable, and maintainable applications. It is often used in conjunction with frameworks like Spring for enterprise-level applications.

8. Spring Framework

- Spring Framework is a comprehensive framework for building Java-based enterprise applications. It provides infrastructure support for developing robust applications, including features like dependency injection, transaction management, and aspect-oriented programming.
- Spring Boot, a part of the Spring ecosystem, simplifies the setup and development of new applications by providing defaults and reducing configuration overhead. It is particularly popular for building microservices.

9. Spring Security

- Spring Security is a powerful and customizable authentication and access control framework for Java applications. It provides comprehensive security services for Java EE-based enterprise software applications.
- Spring Security offers features such as configurable security policies, authentication, authorization, and protection against common vulnerabilities like CSRF and session fixation attacks. It integrates seamlessly with Spring applications, making it a preferred choice for securing web applications.

10. Spring Data JPA

- Spring Data JPA is a part of the larger Spring Data family, aimed at simplifying data access and manipulation. It provides a repository abstraction over JPA (Java Persistence API), facilitating the implementation of data access layers.
- Spring Data JPA reduces boilerplate code by offering a repository model that includes CRUD operations, pagination, and custom query methods. It integrates well with Spring applications, promoting a more streamlined and efficient data access process.

11. MySQL

- MySQL is a widely-used open-source relational database management system (RDBMS). It is known for its reliability, scalability, and ease of use, making it a popular choice for web applications.
- MySQL supports complex queries, transactions, and indexing, which are essential for managing large datasets and ensuring data integrity. Its compatibility with various programming languages and platforms enhances its versatility.

III. Project Management Report

1. Requirements Analysis

1.1. Registration

- The form includes text fields for entering **first name**, **last name**, **email**, **password**, and two radio buttons for selecting **gender**.

- All fields are required, the **email** must be in the correct format, and the **password** must be at least 6 characters long.
- Successful registration will redirect the user to the login page.

1.2. Login

- The form consists of two text fields for entering email and password.
- All fields are required, the email must be in the correct format, and the password must be at least 6 characters long.
- An error message will display if the email or password is incorrect.

1.3. Logout

- Clicking the logout button will log the user out of their account.

1.4. Reset Password

- A form is provided for entering the email to reset the password via email.
- If the entered email does not exist, an error message will be displayed.
- After entering a valid email, an OTP code will be sent to that email.
- The user will be redirected to the OTP entry page, displaying a confirmation form with a field to enter the OTP.
- An error message will appear if the entered OTP is invalid.
- Upon entering a valid OTP, the user will be redirected to the password reset page, which includes fields for new password and password confirmation.
- An error message will display if the password confirmation does not match.
- Successful password reset will redirect the user to the login page.

1.5. View Home Page

- The left sidebar will display navigation buttons to various pages and functions such as Home, Message, Profile, user information, and the logout button.
- The right sidebar will feature a search bar and a list of users.
- The central section will display the post creation area and existing posts.

1.6. Post Creation

- There will be a text field for entering the post content.
- Users can attach images and videos, with the option to preview them before posting.
- Pressing enter will submit the post.

1.7. Delete Post

- Clicking the delete button will remove the post.

1.8. Like or Unlike Posts

- Users can like or unlike posts.
- The number of likes for each post will be displayed.

1.9. Save or Unsave Posts

- Users can save or unsave posts.

1.10. View Profile

- Users can view the cover photo, profile picture, user information, and the option to edit personal information.
- The number of posts, followers, following, and bio will also be displayed.

1.11. Edit Profile Information

- The form includes text fields for entering first name, last name, email, password, and two radio buttons for selecting gender.
- Users can change the cover photo and profile picture.
- Clicking the save button will save the changes.

1.12. Comment on Posts

- Users can view comments on each post.
- Users can submit comments on posts.

1.13. Real-time Messaging with Other Users

- The left sidebar will display ongoing conversations and a search box for finding users.
- Selecting a user will display the conversation content on the right, allowing users to send messages with images.
- Messages can be sent by pressing the enter key.

1.14. Search for Users

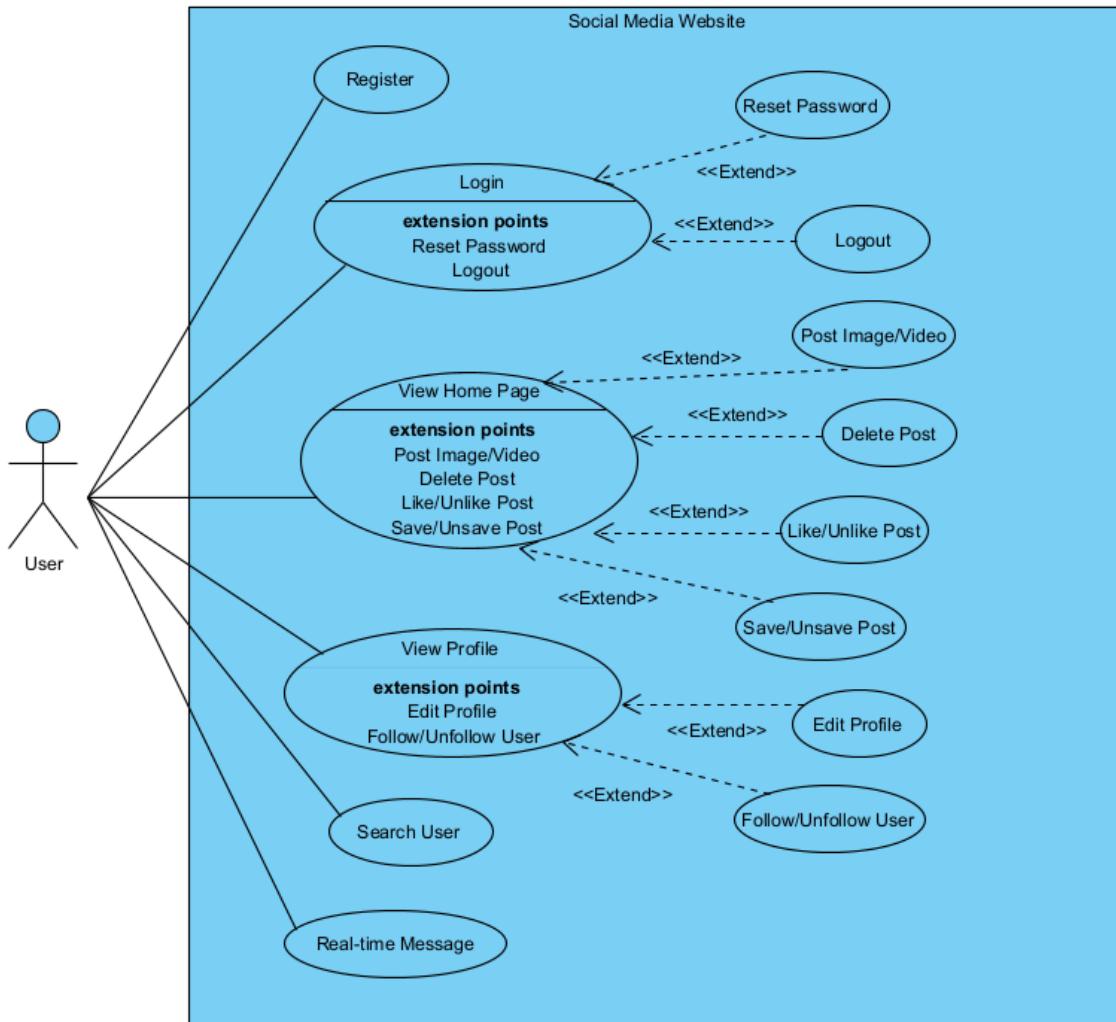
- Users can search for others based on first name, last name, and email.
- A list of users will be displayed, and selecting a user will redirect to that user's profile page.

2. System Analysis And Design

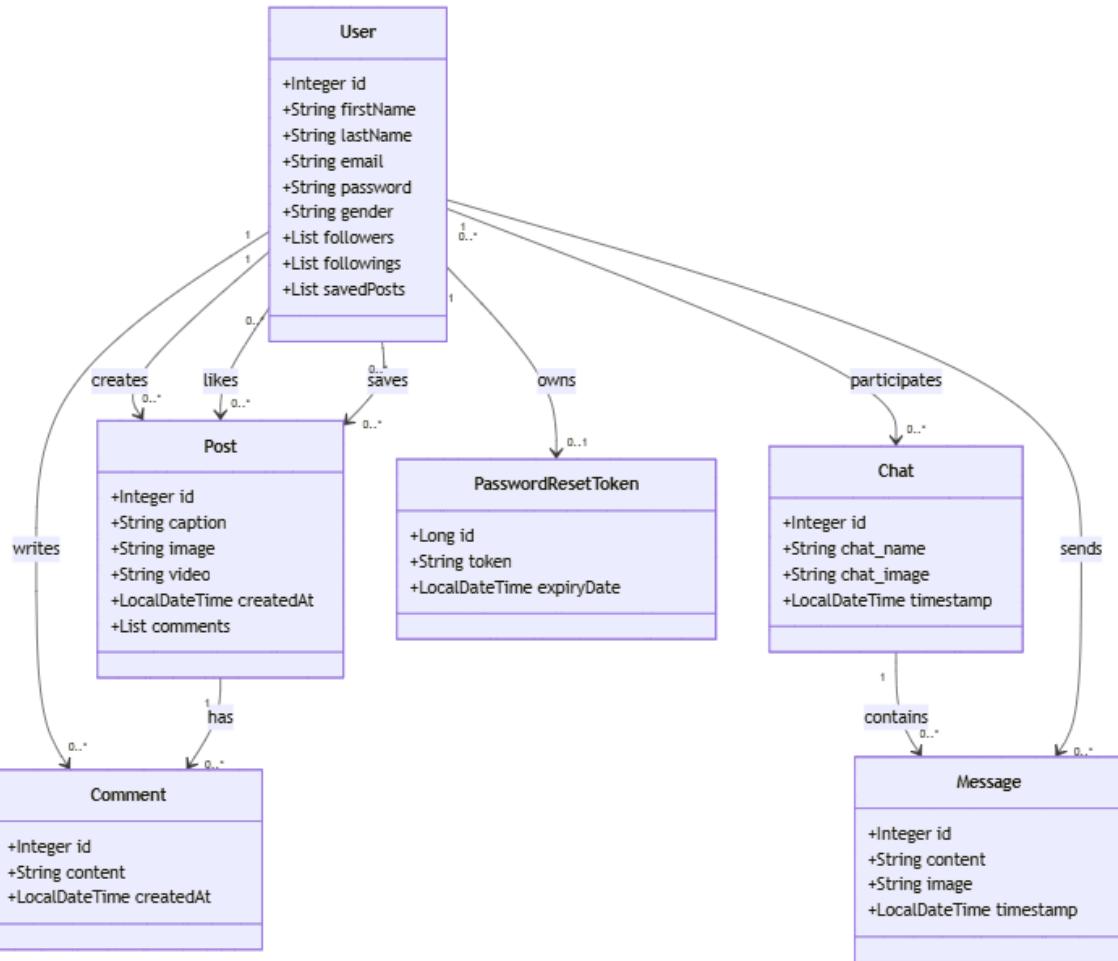
2.1. Overview Design

- **Use Cases:**
 - + **Register:** UC allows users to create an account.
 - + **Login:** UC allows users to log into their account.
 - + **Reset Password:** UC allows users to change their password.
 - + **Logout:** UC allows users to log out of their account.
 - + **View Home Page:** UC allows users to view posts.
 - + **Post Image/Video:** UC allows users to create posts.
 - + **Delete Post:** UC allows users to remove a post.
 - + **Like/Unlike Post:** UC allows users to like or unlike posts.
 - + **Save/Unsave Post:** UC allows users to save or unsave posts.
 - + **Comment on Post:** UC allows users to comment on posts.
 - + **View Profile:** UC allows users to view their own or others' profiles.
 - + **Follow/Unfollow User:** UC allows users to follow or unfollow others.
 - + **Edit Profile:** UC allows users to edit their profile.

- + **Search User:** UC allows users to search for other users.
- + **Real-time Message:** UC allows users to send and receive messages in real-time.



2.2. Entity Class Diagram



2.3. Details Design

- RESTful API Endpoints:

Authentication Endpoints (/auth)

| Method | Path | Description | Request Body | Response | Authentication |
|--------|-----------------------|--|---|-----------------------------------|----------------|
| POST | /auth/signup | Register a new user | User object | AuthResponse (JWT token, message) | ✗ No |
| POST | /auth/signin | Authenticate user and obtain JWT token | LoginRequest (email, password) | AuthResponse (JWT token, message) | ✗ No |
| POST | /auth/forgot-password | Initiate password reset process | PasswordResetRequest (email) | AuthResponse (message, status) | ✗ No |
| POST | /auth/reset-password | Reset user's password using token | ResetPasswordRequest (token, newPassword) | AuthResponse (message, status) | ✗ No |

User Endpoints (/api/users)

| Method | Path | Description | Request Body | Response | Authentication |
|--------|------------------------------|---|--------------|-----------------------|---|
| GET | /api/users | Retrieve all users | None | List of User objects | <input checked="" type="checkbox"/> Yes |
| GET | /api/users/{userId} | Retrieve user profile by user ID | None | UserProfileResponse | <input checked="" type="checkbox"/> Yes |
| PUT | /api/users | Update the authenticated user's profile | User object | Updated User object | <input checked="" type="checkbox"/> Yes |
| POST | /api/users/follow/{userId} | Follow a specific user | None | FollowUserResponse | <input checked="" type="checkbox"/> Yes |
| POST | /api/users/unfollow/{userId} | Unfollow a specific user | None | UnFollowUser Response | <input checked="" type="checkbox"/> Yes |
| GET | /api/users/search?query= | Search for users based on a query | None | List of User objects | <input checked="" type="checkbox"/> Yes |
| GET | /api/users/profile | Retrieve the authenticated user's profile | None | UserProfileResponse | <input checked="" type="checkbox"/> Yes |
| POST | /api/users/save/{postId} | Save or unsave a specific post | None | Post object | <input checked="" type="checkbox"/> Yes |

| | | | | | |
|-----|--------------------------|--|------|----------------------|---|
| GET | /api/users/save/{userId} | Retrieve all saved posts for a specific user | None | List of Post objects | <input checked="" type="checkbox"/> Yes |
|-----|--------------------------|--|------|----------------------|---|

Chat Endpoints (/api/chats)

| Method | Path | Description | Request Body | Response | Authentication |
|--------|------------|---|----------------------------|----------------------|---|
| POST | /api/chats | Create a new chat between users | CreateChatRequest (userId) | Chat object | <input checked="" type="checkbox"/> Yes |
| GET | /api/chats | Retrieve all chats for the authenticated user | None | List of Chat objects | <input checked="" type="checkbox"/> Yes |

Message Endpoints (/api/messages)

| Method | Path | Description | Request Body | Response | Authentication |
|--------|-----------------------------|--|----------------|-------------------------|---|
| POST | /api/messages/chat/{chatId} | Send a message within a specific chat | Message object | Message object | <input checked="" type="checkbox"/> Yes |
| GET | /api/messages/chat/{chatId} | Retrieve all messages from a specific chat | None | List of Message objects | <input checked="" type="checkbox"/> Yes |

Post Endpoints (/api/posts)

| Method | Path | Description | Request Body | Response | Authentication |
|--------|--------------------------|---|--------------|-------------------------------|---|
| POST | /api/posts | Create a new post | Post object | Post object | <input checked="" type="checkbox"/> Yes |
| DELETE | /api/posts/{postId} | Delete a specific post by ID | None | ApiResponse (message, status) | <input checked="" type="checkbox"/> Yes |
| GET | /api/posts/{postId} | Retrieve a specific post by ID | None | Post object | <input checked="" type="checkbox"/> Yes |
| GET | /api/posts/user/{userId} | Retrieve all posts from a specific user | None | List of Post objects | <input checked="" type="checkbox"/> Yes |
| GET | /api/posts | Retrieve all posts | None | List of Post objects | <input checked="" type="checkbox"/> Yes |
| PUT | /api/posts/like/{postId} | Like or unlike a specific post | None | Post object | <input checked="" type="checkbox"/> Yes |

Comment Endpoints (/api/comments)

| Method | Path | Description | Request Body | Response | Authentication |
|--------|-----------------------------|-------------------------------------|----------------|----------------|---|
| POST | /api/comments/post/{postId} | Create a comment on a specific post | Comment object | Comment object | <input checked="" type="checkbox"/> Yes |

- **WebSocket API Endpoints**

| Protocol | URL Path | Description | Message Type | Response Type | Authentication |
|-----------|----------|---|---------------------|----------------|---|
| WebSocket | /ws | Establish a WebSocket connection for real-time chat | /app/chat/{groupId} | Message object | <input checked="" type="checkbox"/> Yes |

2.4. Testing and Quality Assurance

- **Usability Testing:**

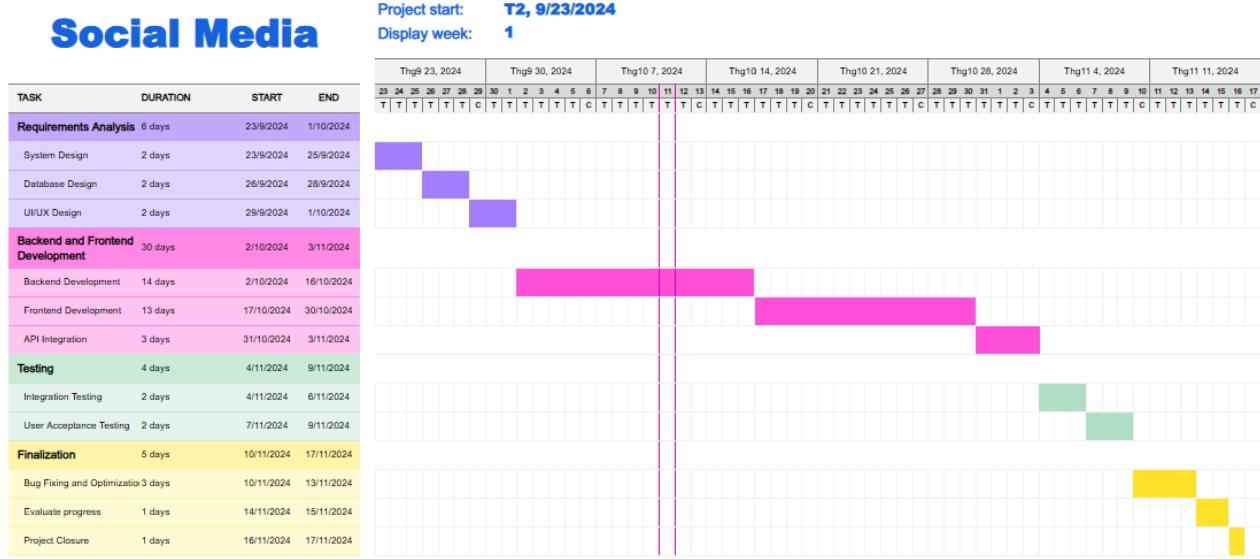
- + All website content is displayed correctly without any spelling or grammatical errors.
- + All fonts adhere strictly to the design specifications, ensuring uniformity across the platform.
- + All textual elements, including headings, paragraphs, and labels, are precisely aligned as per the design guidelines.
- + All input fields, such as registration forms, login forms, and post creation forms, are correctly aligned to maintain a clean and organized layout.
- + All buttons (e.g., Submit, Like, Follow) have the correct form, style, and size, ensuring they are easily identifiable and accessible to users.
- + All pages are properly linked to the home page and other relevant sections, ensuring seamless navigation throughout the application.

- + All hyperlinks and images load correctly without broken links or missing assets.
- + All tabs within the application (e.g., navigation tabs, content tabs) operate correctly, allowing users to switch between different sections without issues.
- + Scroll bars appear and function as intended, providing smooth navigation through content that exceeds the viewport.
- **Functional Testing:**
 - + All mandatory (required) fields in forms (e.g., registration, login, post creation) trigger appropriate error messages if left empty during submissions.
 - + Fields that are not mandatory do not display error messages when left empty, allowing users flexibility in form submissions.
 - + The password reset process functions correctly, including token generation, validation, and expiration.
 - + Emails are sent reliably for password reset requests without failures.
 - + Users can successfully follow and unfollow other users, with updates reflected accurately in follower and following lists.
 - + Users can like and unlike posts, with the like counts updating correctly.
 - + Real-time chat features operate smoothly, allowing users to send and receive messages instantly.
 - + Message broadcasting to chat participants works without delays or losses.

- + Users can create, edit, and delete posts effectively, with changes reflected immediately across the platform.
- + The application handles errors gracefully, providing meaningful feedback to users without exposing sensitive information.
- + Protected endpoints are inaccessible without proper authentication, ensuring data security and privacy.
- **Compatibility Testing:**
 - + The application maintains full functionality and accurate display across various browsers, including Internet Explorer (IE), Firefox, Chrome, Safari, and Opera.
 - + Images render correctly and consistently across different browsers and devices, maintaining quality and alignment.
 - + All fonts used in the application are supported and display uniformly across different browsers and devices.
 - + JavaScript code executes correctly across all supported browsers, ensuring interactive features function seamlessly.
 - + Real-time chat features via WebSockets operate consistently across different browsers, ensuring reliable message delivery and reception.

3. Project Schedule

Social Media



4. Results

4.1. Completed Features

1. User Registration (Signup)

- New users can create accounts by providing necessary information such as name, email, and password.

2. User Authentication (Login & Logout)

- Secure login and logout mechanisms for users to access their accounts and terminate sessions safely.

3. View and Edit Account Information

- Users can view their profile details and update information like name, email, and password.

4. Password Recovery

- Functionality for users to reset their passwords through email verification.

5. Home Page Display

- An intuitive and engaging home page showcasing recent posts.

6. Search Functionality

- Users can search for other users or specific posts using keywords, enhancing discoverability within the platform.

7. View profile

- Users can view their and others profile details, including personal information, their posts, their saved posts.

8. Like/Unlike Functionality

- Users can express appreciation for posts by liking or unliking them.

9. Follow/Unfollow Users

- Ability for users to follow other users to stay updated with their activities and content.

10. Real-Time Chat

- Engage in real-time conversations with other users through the integrated chat feature.

11. Save/Unsave Posts

- Option for users to save posts for future reference or interaction.

4.2. Incomplete Features

1. Notifications

- Users do not receive real-time notifications for activities such as likes, comments, or new followers.

2. Story

- The feature allowing users to create and view temporary stories is pending development.

3. Reels

- Short-form video content creation and viewing (Reels) functionality has not been developed.

4. Responsive Design Enhancements

- Further optimizations to ensure flawless performance and display across all device types and screen sizes.

⇒ **Project Completion Level:** 90%

4.3. Conclusion

- **Advantages:**
 - **Comprehensive Feature Set:**
 - The website successfully implements a wide range of features as per the project requirements, providing a robust and engaging platform for users.
 - **User-Friendly Interface:**
 - The application boasts an intuitive and visually appealing interface, ensuring ease of use and enhancing the overall user experience.
 - **Intuitive Functionality:**
 - Features such as real-time chat, post interactions (likes, comments), and content management are designed to be user-friendly and align closely with user expectations.
 - **Secure Authentication:**
 - Implementation of secure authentication mechanisms, including JWT-based login and password encryption, ensures user data protection and platform security.
 - **Scalable Architecture:**
 - The modular codebase and use of Spring Boot facilitate scalability, allowing the application to handle increasing user loads and feature expansions seamlessly.
 - **Real-Time Communication:**
 - The implementation of real-time communication technologies like WebSockets enables instant updates and interactions between users and the platform.

- Integration of WebSockets for real-time chat functionality enhances user engagement and interaction within the platform.
- **Disadvantages:**
 - **Incomplete Features:**
 - Several key features, including notifications, stories, and reels, remain undeveloped. These functionalities are vital for enhancing user engagement and interactivity within the platform.
 - **Basic Error Handling:**
 - While basic error handling is in place, more sophisticated mechanisms could be implemented to provide detailed feedback and enhance user experience during failures.

4.4. Ongoing Development Challenges

- As the project is still under active development, some existing features may require further refinement to ensure optimal performance and user satisfaction.

4.5. Future Development Directions

- **Implementation of Pending Features:**
 - + **Notifications:** Introduce real-time notifications to alert users about interactions such as likes, comments, and new followers.
 - + **Story:** Implement the stories feature to allow users to post temporary content, enhancing real-time interaction.
 - + **Reels:** Develop short-form video content creation and viewing to align with current social media trends.
- **User Experience Improvements:**
 - + Optimize the application's performance to ensure fast load times and smooth interactions across all devices.

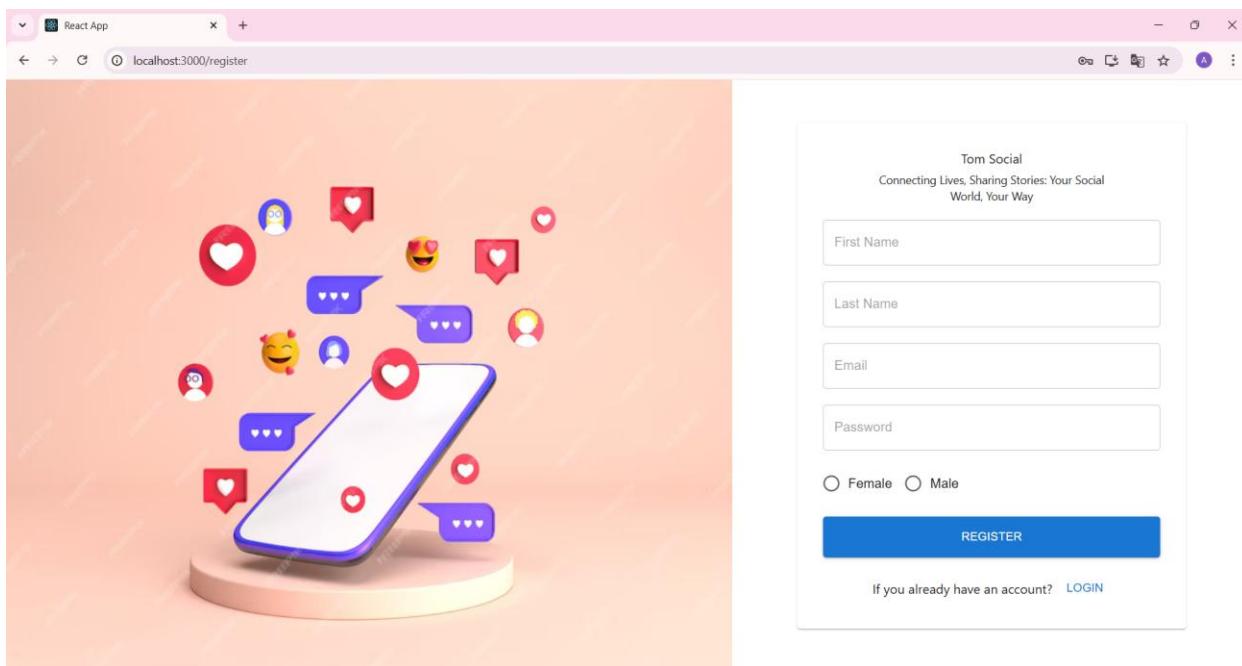
4.6. Summary

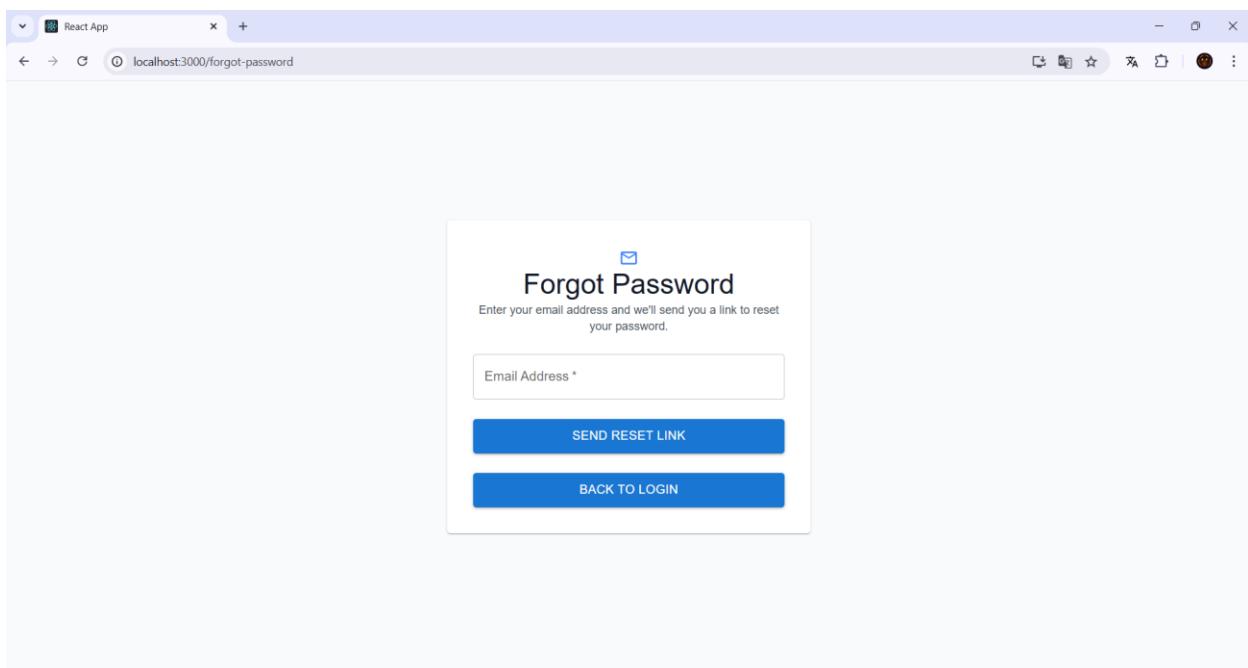
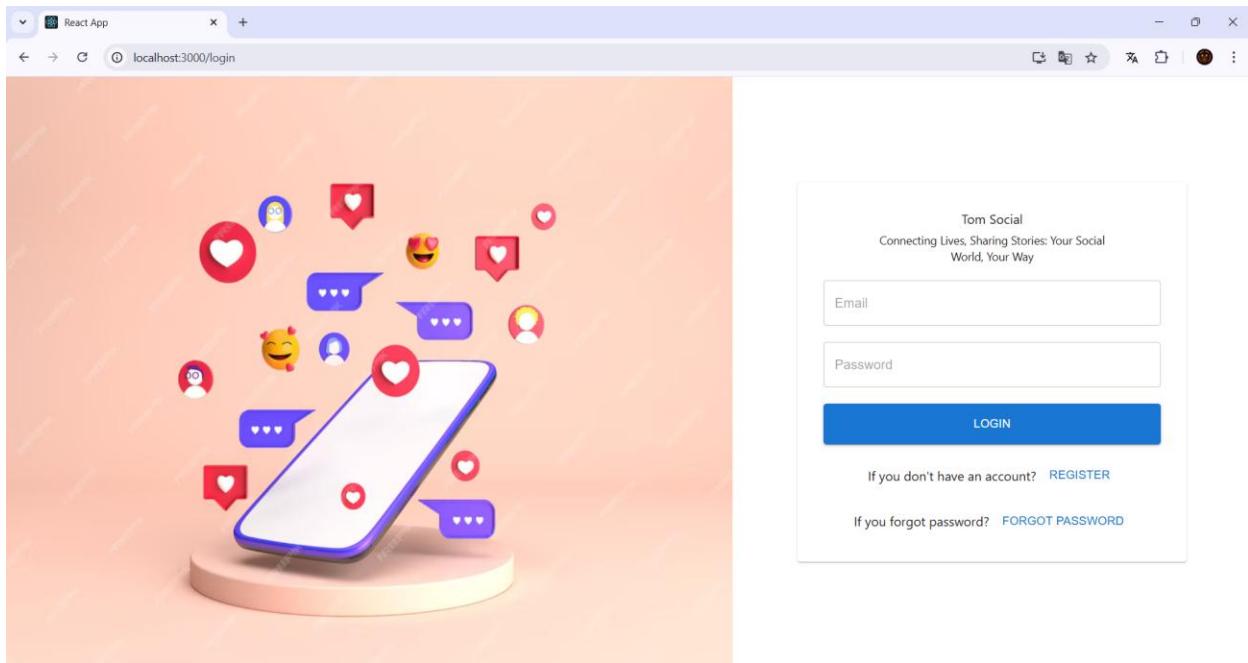
- Throughout the development of the **Social Media** application, significant efforts have been made to implement a robust set of

features that form the foundation of a functional social media platform. While the majority of essential functionalities are in place, certain interactive features such as notifications, stories, and reels are yet to be developed. These pending features are crucial for fostering user engagement and providing a comprehensive social media experience.

- Moving forward, the focus will be on completing the remaining features, enhancing existing functionalities, and optimizing the overall user experience. Continuous research and development efforts will ensure that the application evolves to meet user expectations and industry standards. Special thanks to lecturer who have provided guidance and support throughout the project's development.

IV. Appendix



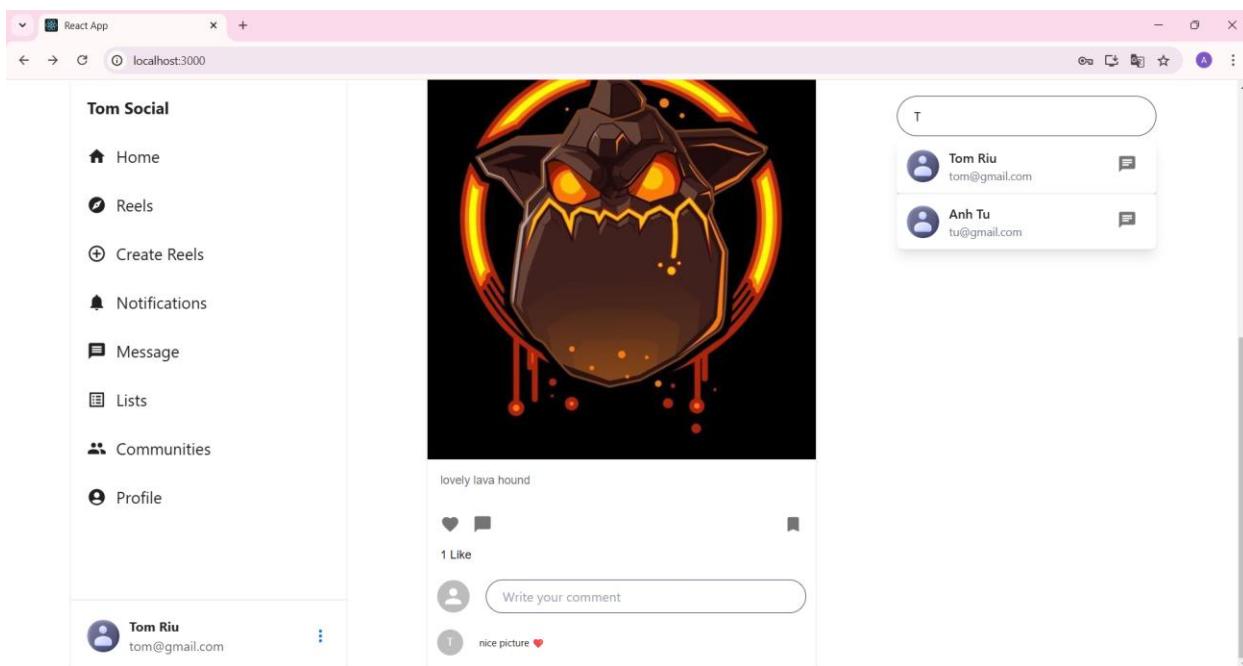
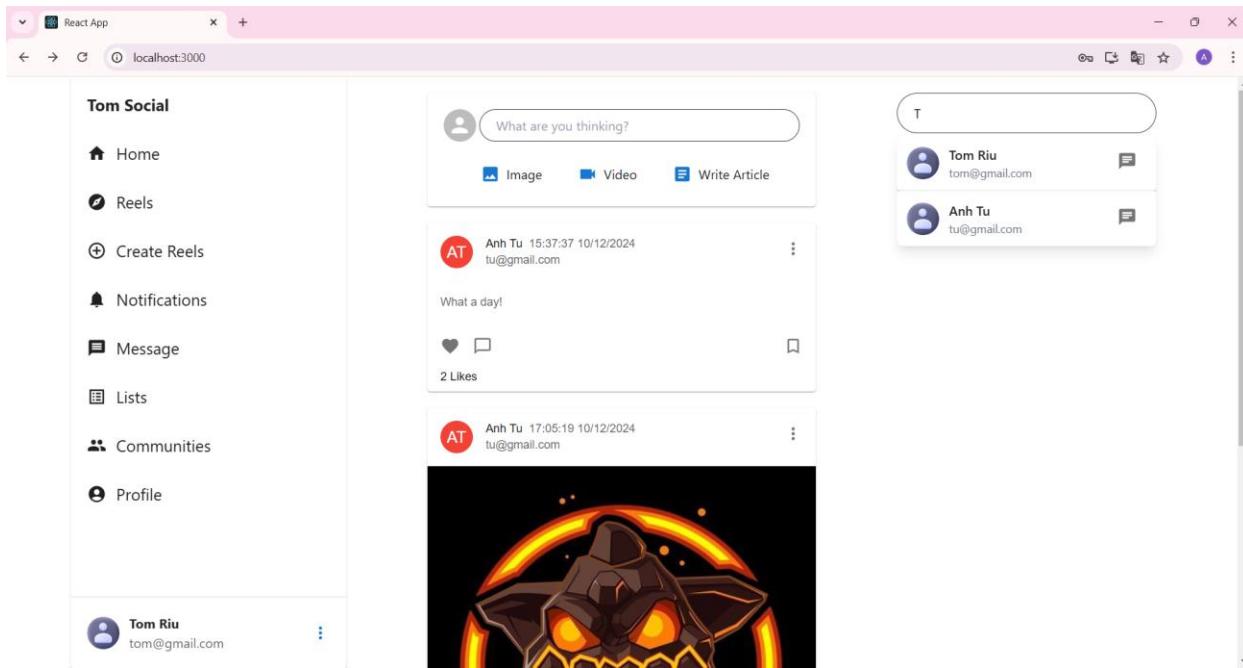


A screenshot of a web browser window showing a password reset form. The title bar says "React App" and the address bar shows "localhost:3000/reset-password?token=0c08be11-3d92-441c-b5d9-312869fad6a3". The main content is a "Reset Password" form with a blue padlock icon at the top. It includes a note "Please enter your new password below" and a "New Password *" input field with a visibility toggle icon. Below the input field is a list of validation rules:

- At least 8 characters
- One uppercase letter
- One lowercase letter
- One number
- One special character

There are two buttons at the bottom: a grey "RESET PASSWORD" button and a blue "BACK TO LOGIN" button.

A screenshot of a web browser window showing a social media application. The title bar says "React App" and the address bar shows "localhost:3000". On the left is a sidebar with user navigation options: Home, Reels, Create Reels, Notifications, Message, Lists, Communities, and Profile. The main area shows a feed with a post from a user named "Code With Tom" (@codewithtom). The post interface includes a text input field with placeholder "What are you thinking?", file upload buttons for "Image" and "Video", and a "POST" button. A decorative graphic of a stylized dragon or creature head is visible at the bottom of the feed area. In the bottom left corner of the main area, there is a profile card for "Tom Liu" with the email "tom@gmail.com".



React App

localhost:3000/profile/1

Tom Social

- Home
- Reels
- Create Reels
- Notifications
- Message
- Lists
- Communities
- Profile

Tom Riu
tom@gmail.com

0 posts 0 followers 2 followings
No bio available.

POST REELS SAVED REPOST

No posts.



EDIT PROFILE

React App

localhost:3000/profile/1

Tom Social

- Home
- Reels
- Create Reels
- Notifications
- Message
- Lists
- Communities
- Profile

Tom Riu
tom@gmail.com

0 posts 0 followers 2 followings
No bio available.

POST REELS SAVED REPOST

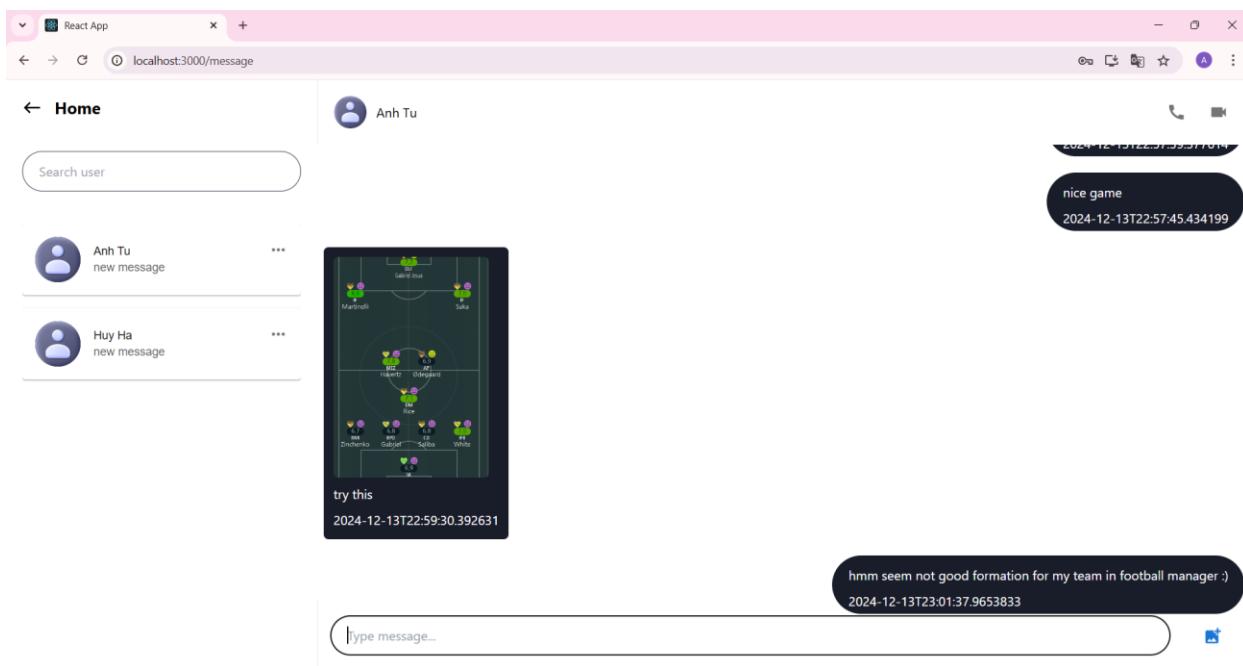
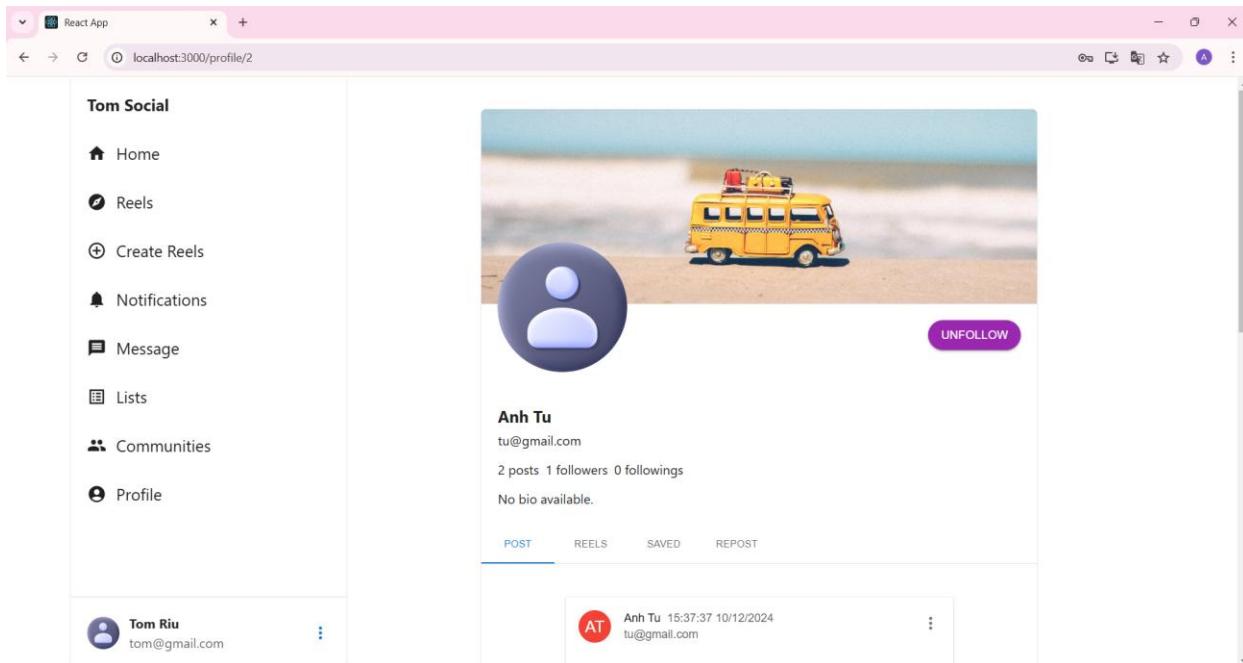
AT Anh Tu 15:37:37 10/12/2024 tu@gmail.com

What a day!

2 Likes



EDIT PROFILE



V. References

1. W3Schools. (2023) HTML Introduction. Available at: https://www.w3schools.com/html/html_intro.asp (Accessed: 1 October 2024).
2. CSS-Tricks. (2023) A Complete Guide to Flexbox. Available at: <https://css-tricks.com/snippets/css/a-guide-to-flexbox/> (Accessed: 1 October 2024).

3. React. (2023) React – A JavaScript library for building user interfaces. Available at: <https://reactjs.org/> (Accessed: 1 October 2024).
4. Redux. (2023) Introduction to Redux. Available at: <https://redux.js.org/introduction/getting-started> (Accessed: 1 October 2024).
5. Material-UI. (2023) Material-UI: React components for faster and easier web development. Available at: <https://mui.com/> (Accessed: 1 October 2024).
6. Tailwind CSS. (2023) Tailwind CSS: Rapidly build modern websites without ever leaving your HTML. Available at: <https://tailwindcss.com/> (Accessed: 1 October 2024).
7. Formik. (2023) Formik: Build Forms in React. Available at: <https://formik.org/> (Accessed: 5 October 2024).
8. Oracle. (2023) Java Platform, Standard Edition Documentation. Available at: <https://docs.oracle.com/javase/8/docs/> (Accessed: 1 October 2024).
9. Spring. (2023) Spring Framework Documentation. Available at: <https://spring.io/projects/spring-framework> (Accessed: 1 October 2024).
10. Spring Security. (2023) Spring Security Reference. Available at: <https://spring.io/projects/spring-security> (Accessed: 1 October 2024).
11. Spring Data. (2023) Spring Data JPA Reference. Available at: <https://spring.io/projects/spring-data-jpa> (Accessed: 1 October 2024).
12. MySQL. (2023) MySQL :: MySQL Documentation. Available at: <https://dev.mysql.com/doc> (Accessed: 1 October 2024).