TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PURWANCHAL CAMPUS

A MINOR PROJECT REPORT ON A DECENTRALIZED SOCIAL MEDIA FOR SCIENTIFIC COMMUNICATION

BY

Rijan Karki(PUR078BCT067)

Saurav Khanal(PUR078BCT080)

Spandan Guragain(PUR078BCT086)

Sudesh Subedi(PUR078BCT088)

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

PURWANCHAL CAMPUS

DHARAN, NEPAL

FEBRUARY,2025

Head

Department of Electronics and Computer Engineering

Purwanchal Campus, Institute of Engineering

Dharan, Sunsari

Nepal

DECLARATION

We declare that the work hereby submitted for Bachelors of Engineering in Computer Engineering at Institute of Engineering, Purwanchal Campus entitled **A DECENTRALIZED SOCIAL MEDIA FOR SCIENTIFIC COMMUNICATION** is our own work and has not been previously submitted by us at any university for any academic award.

We authorize Institute of Engineering, Purwanchal Campus to lend this report to other institution or individuals for the purpose of scholarly research.

Rijan Karki PUR078BCT067
Saurav Khanal PUR078BCT080
Spandan Guragain PUR078BCT086
Sudesh Subedi PUR078BCT088

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to our supervisor Mr. Pukar Karki, Head of the Department, for his invaluable guidance and unwavering support throughout the preparation of this report. Their insightful suggestions and constructive feedback have been instrumental in shaping our work, and we are truly appreciative of the time and effort they dedicated to our development. Their leadership has not only provided us with direction but has also inspired us to pursue excellence in our academic endeavors.

We are also deeply thankful to our esteemed teachers and faculty members for their encouragement and thoughtful insights, which have significantly enriched this proposal. Their commitment to academic excellence and their willingness to share knowledge have served as a constant source of inspiration for us. The collective guidance and support we received from them have greatly contributed to the refinement and improvement of our work, and we are sincerely grateful for their contributions.

Rijan Karki PUR078BCT067
Saurav Khanal PUR078BCT080
Spandan Guragain PUR078BCT086
Sudesh Subedi PUR078BCT088

ABSTRACT

Scientific communication on mainstream social media platforms is often hindered by a lack of support for mathematical notation, academic citations, and decentralized data control. This report presents the development of a federated social media platform designed specifically for researchers and academics, built using the ActivityPub protocol. The platform enables seamless sharing of scientific content while maintaining interoperability with the broader Fediverse. Key features include support for scientific typesetting, user-friendly server deployment, and the ability for institutions and individuals to host their own instances without losing connectivity. By leveraging decentralization and federation, this project provides a privacy-conscious, accessible, and academically enriching alternative to existing platforms. This report details the design, implementation, and evaluation of the system, highlighting the challenges faced and the solutions developed.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

API	:	Application Programming Interface
UI	:	User Interface
W3C	:	World Wide Web Consortium

CHAPTER 1

INTRODUCTION

1.1 Background

Scientific communication plays a vital role in advancing research and knowledge sharing across academic communities. Traditional social media platforms while effective for general communication often lack specialized features necessary for scientific discource. The emergence of decentralized technologies particularly the ActivityPub [1] Protocol and the Fediverse presents an oppurtunity to create a more suitable platform for academic communication.

1.2 Gap Identification

Current platforms for scientific communication face several limitations:

- Limited accessibility of scientific communication to the broader population beyond niche communities.
- Insufficient support for mathematical expressions and scientific notations.
- Lack of integration with academic citation systems.
- Reliance on a third party for the protection and moderation of user data.

1.3 Motivation

To create a social media platform that empowers researchers and academics to communicate their scientific work effectively. By bridging the gap between specialized communities and the general public, the platform aims to promote the understanding and appreciation of cutting-edge research across a wider audience. Along with the ability to run individual servers by individual people/institutions without losing the ability to communicate between each other.

1.4 Objectives

- Develop a federated social media platform using the ActivityPub protocol with support for mathematical and scientific typesetting.
- Streamline the server setup process to enable technically literate individuals to host their own servers with minimal effort.

CHAPTER 2

RELATED THEORY

2.1 Decentralization

Decentralization refers to the distribution of authority, control, and decision-making away from a central authority. In the context of digital platforms, decentralization allows users to maintain control over their data and interactions, fostering a more democratic and resilient online environment. This approach contrasts with traditional centralized systems, where a single entity governs all operations and data management.

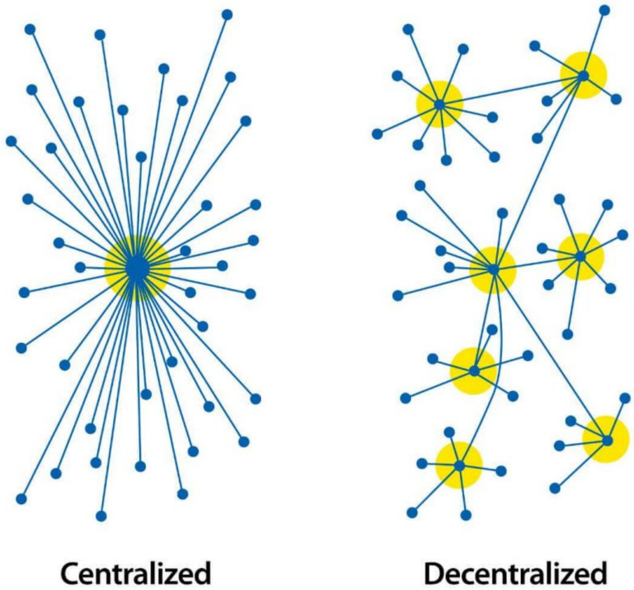

Figure 2.1: The structure of a centralized network compared to a decentralized network.

2.2 Federation

Federation is a model that enables different systems or organizations to interoperate while maintaining their independence. In social media, a federated approach allows users from different platforms to communicate and share content seamlessly, creating a more interconnected online community. This is achieved through protocols that facilitate data exchange and interaction across diverse platforms.

2.3 Scientific Typesetting

Scientific typesetting involves the formatting of mathematical and scientific content for clarity and precision. It is essential for effectively communicating complex ideas in academic and research contexts. Proper typesetting ensures that equations, symbols, and notations are presented in a way that is easily understandable and visually appealing. The prime example of a Scientific Typesetting is LaTeX but there are alternatives like AsciiMath and Typst.

2.4 ActivityPub Protocol

ActivityPub is a decentralized social networking protocol standardized by the World Wide Web Consortium (W3C) [1]. It provides a client-to-server API for creating, updating, and deleting content, as well as a server-to-server API for delivering notifications and content between different servers.

The federation model offers several key advantages over centralized systems, such as resilience (no single point of failure as the network operates across multiple independent servers), data sovereignty (each instance maintains control over its users' data and policies), interoperability (users can communicate across instances using standardized protocols), and scalability (the network can grow organically as new instances join the federation).

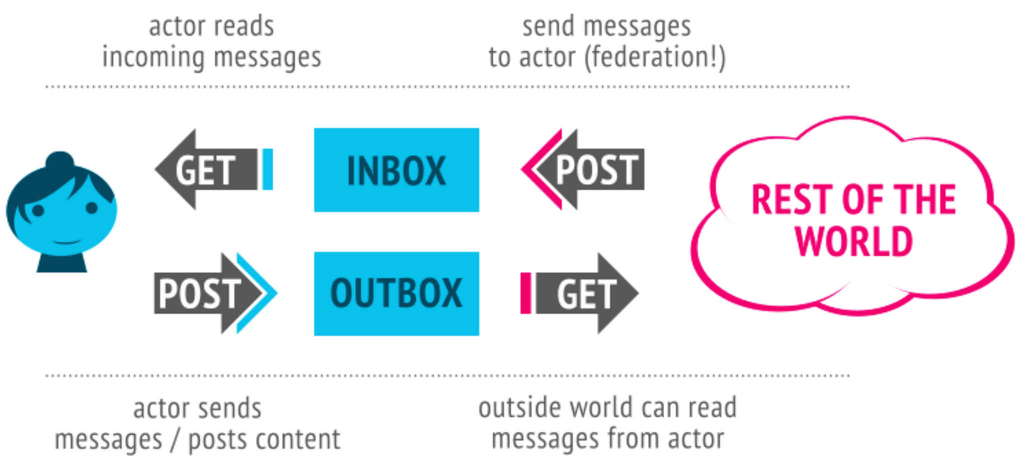

Figure 2.2: Communication using ActivityPub Protocol.

The protocol is built on several key concepts:

- **Actors** which represent users, groups, or applications that can send and receive activities.
- **Activities** which describe actions that actors take.
- **Objects** which represent the content being acted upon.

CHAPTER 3

LITERATURE REVIEW

3.1 Related Works

The Fediverse connects various decentralized social networks, allowing users to interact across different platforms. This section reviews some of the prominent projects within the Fediverse.

- **Mastodon**[2]
  Mastodon is a decentralized social network that operates on open-source software. It allows users to create their own servers (instances) and interact with users on other instances. Mastodon emphasizes user privacy and control over content, providing features like content warnings and robust moderation tools.

- **Pixelfed**[3]
  Pixelfed is a federated image-sharing platform similar to Instagram. It focuses on user privacy and data ownership, allowing users to share photos and interact with others without centralized control. Pixelfed supports ActivityPub, enabling interaction with other Fediverse platforms.

- **PeerTube**[4]
  PeerTube is a decentralized video hosting platform that uses peer-to-peer technology to distribute video content. It aims to provide an alternative to centralized video platforms like YouTube, giving users control over their content and reducing reliance on centralized servers. PeerTube instances can federate with each other, allowing for a distributed network of video content.

- **Lemmy**[5]
  Lemmy is a federated link aggregation and discussion platform similar to Reddit. It allows users to create and join communities, share links, and engage in discussions. Lemmy instances can federate with each other like it is the case with other social media in this list, enabling a decentralized network of communities that can still connect with one another.

- **snac**[6]

  SNAC is a decentralized, open-source social network that emphasizes user privacy and content control. It allows users to create their own servers (instances) and interact with others across the Fediverse. SNAC is a lightweight ActivityPub implementation with features like Mastodon API support, a simple web interface, and no database or JavaScript dependencies. Written in portable C, SNAC can be easily compiled and deployed on various platforms, providing a minimalistic alternative to mainstream social media.

3.2 Decentralization and Federation

Decentralization and federation are key concepts in the Fediverse. Decentralization refers to the distribution of data and control across multiple servers, reducing reliance on a single central authority. Federation allows different servers to communicate and interact with each other, creating a network of interconnected platforms.

3.2.1 Benefits of Decentralization

Decentralization offers several benefits, including:

- **Privacy and Control**: Users have greater control over their data and privacy settings, as there is no central authority collecting and monetizing user data.
- **Resilience**: Decentralized networks are more resilient to censorship and outages, as there is no single point of failure.
- **Community Governance**: Users can create and govern their own instances, fostering diverse and self-sustaining communities.

3.2.2 Challenges of Decentralization

Despite its benefits, decentralization also presents challenges:

- **Interoperability**: Ensuring seamless interaction between different platforms and instances can be complex.
- **Moderation**: Decentralized networks require robust moderation tools to manage

content and prevent abuse.

- **Scalability**: Decentralized systems must be designed to handle large numbers of users and high volumes of data.

3.3 Protocols and Technologies

The Fediverse relies on various protocols and technologies to enable decentralization and federation. Some of the key protocols include:

- **ActivityPub**[1]
  ActivityPub is a decentralized social networking protocol used by many Fediverse platforms. It enables users to follow, share, and interact with content across different instances.
- **WebFinger**[7]
  WebFinger is a protocol for discovering information about people and resources on the internet. It is used in the Fediverse to locate user profiles and instances.

3.4 Conclusion

The Fediverse represents a growing movement towards decentralized and federated social networks. By leveraging protocols like ActivityPub, platforms within the Fediverse offer users greater control over their data, enhanced privacy, and resilient communities. However, challenges such as interoperability, moderation, and scalability must be addressed to ensure the continued growth and success of decentralized social networks.

CHAPTER 4

METHODOLOGY

The development of our decentralized microblogging platform for scientific communication followed an Agile methodology, prioritizing iterative progress, flexibility, and team collaboration. This chapter outlines the key phases—Requirement Analysis, System Design, Development, Integration with Decentralized Protocols, and Testing—highlighting the processes and strategies that shaped the platform's realization.

4.1 Requirement Analysis

We began by identifying user needs and system requirements through a review of online academic forums, existing social media platforms, and federated system documentation. This research informed essential features: secure data sharing, privacy controls, scientific typesetting support, and decentralized communication. By analyzing the strengths and weaknesses of comparable platforms, we established core functionalities such as user profiles, microblog posts, and cross-instance federation.

4.2 Development

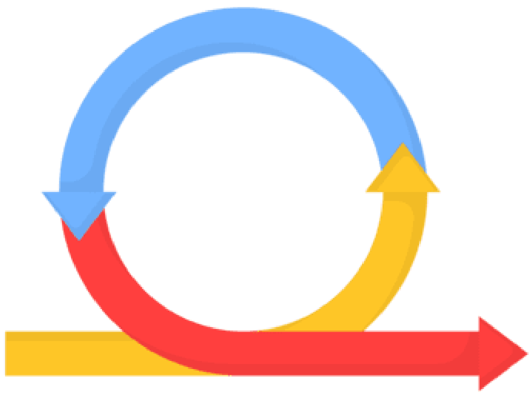

Figure 4.1: Agile Model

Development proceeded in iterative Agile sprints, split into backend and frontend streams.

This approach allowed continuous refinement based on emerging technical needs and team feedback. The backend handled authentication, content management, and federation, while the frontend focused on delivering an intuitive interface for scientific content creation and consumption.

4.3 System Architecture

The design phase centered on creating a scalable, federated architecture. We chose the ActivityPub protocol to enable decentralization, structuring the system around a backend for core logic and a mobile-first frontend for user interaction. Emphasis was placed on secure data exchange and usability across devices, with RESTful APIs as the backbone of component communication.

A modular architecture is used for scalability and maintainability. The figure below gives a high-level view of the system architecture.

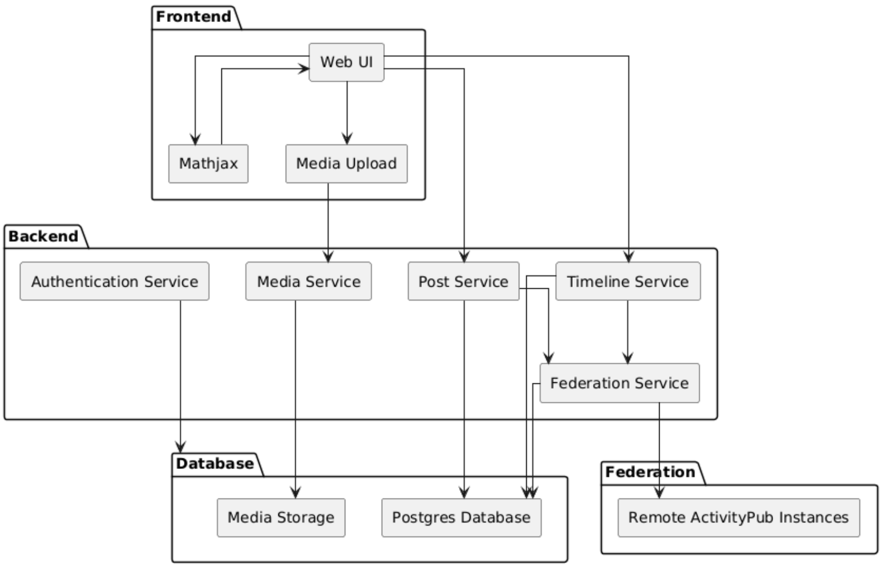

Figure 4.2: Component Diagram

4.4 Integration with Decentralized Protocols

Federation was a cornerstone of the platform, achieved through ActivityPub compliance. We implemented server-to-server communication and user activity management (e.g., posting, following) to ensure interoperability with other federated systems. This phase was validated iteratively to confirm seamless cross-platform interaction.

4.5 Testing

Testing was integrated throughout development to maintain functionality and reliability. We employed a multi-layered strategy, including unit tests for individual components and integration tests for system-wide interactions, ensuring robustness at every stage.

4.6 Technology Stack and Environment Setup

4.6.1 Backend

- **Language:** Node.js and Hono Server
- **Database:** Posgres, S3 Storage, Minio
- **Authentication:** OAuth

4.6.2 Frontend

- **Framework:** ReactNative, Expo, Mathjax

4.6.3 Environment

- **Deployment:** Docker
- **Development:** Docker and Nix

4.7 Detailed Implementation Steps

4.7.1 Backend Implementation

- **User Authentication:** Using OAuth.
- **Content Management:** CRUD operations for posts, comments, and likes.

4.7.2 Frontend Implementation

- **User Interface:** Developed as a Single-Page Application (SPA) using React.

- **State Management:** Handled with Redux integrating with backend APIs.

4.8 Diagrams

4.8.1 Use Case Diagram

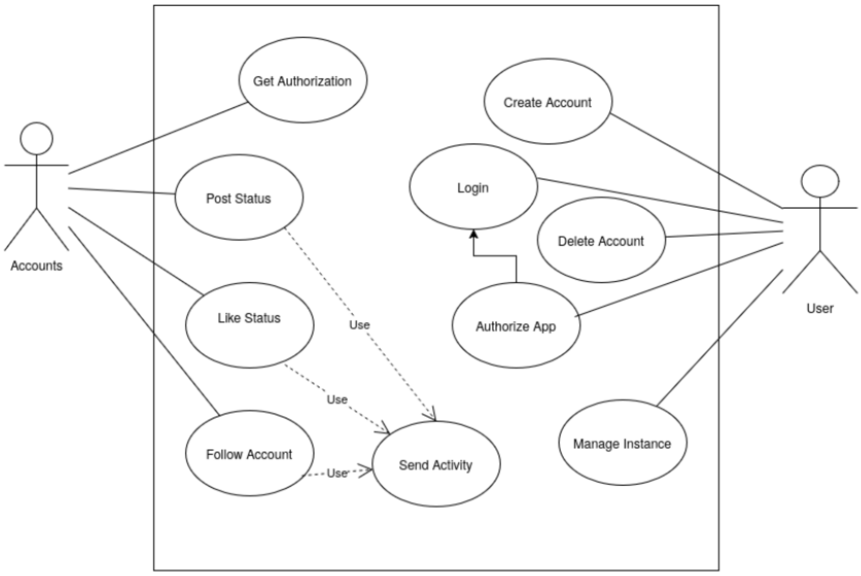

Figure 4.3: Use Case Diagram

4.8.2 ER Diagram

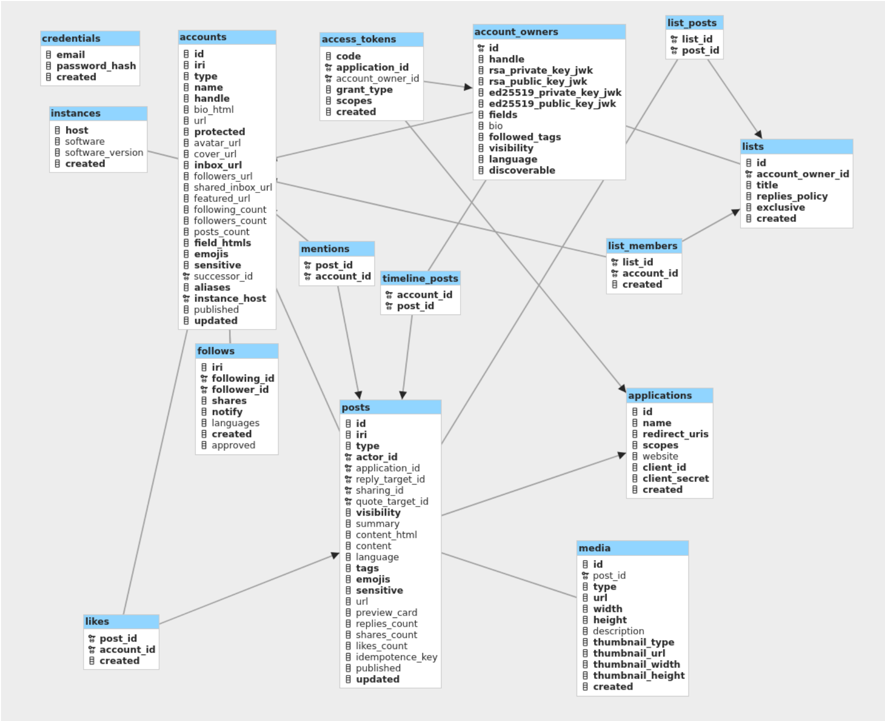

Figure 4.4: ER Diagram

CHAPTER 5

RESULTS

5.1 User & Account Creation

The platform follows a unique approach where each server is dedicated to a single user. However, within that server, the user can create and manage multiple accounts. This setup ensures full ownership and control over data while still enabling interaction within the federated network.

5.1.1 User Creation

To set up a user, the individual must provide a valid email address and set a secure password. This step ensures authentication and account recovery options. Once the user is registered, they gain full control over their server and can proceed to create multiple accounts as needed.

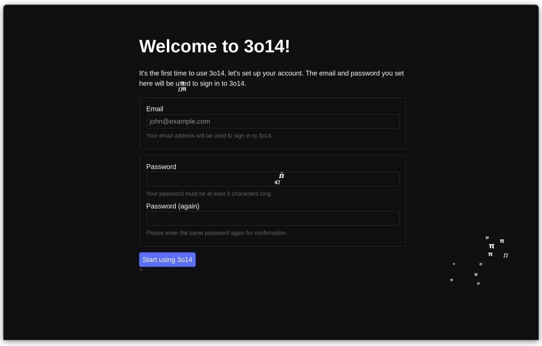

Figure 5.1: User Creation

5.1.2 Account Creation

Within their personal server, the user can create and manage multiple accounts. Each account can have distinct identities, preferences, and privacy settings while still operating under the same server. This feature allows for flexibility in communication, enabling users to maintain separate professional and personal identities, for example.

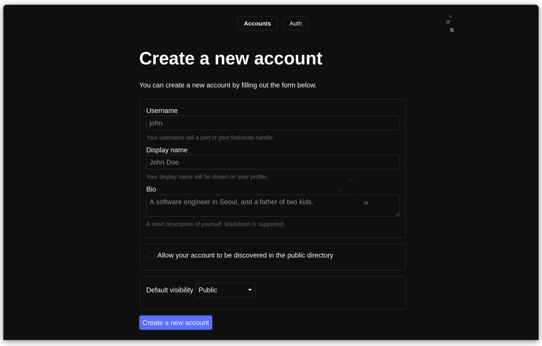

Figure 5.2: Account Creation

5.2 Posting Status With Latex

Users can include mathematical expressions within plaintext using LaTeX syntax while composing a post. As shown in Figure 5.3, the application provides a live preview of the rendered LaTeX output, allowing users to verify their equations before posting. Once published, the post is displayed in the timeline with properly formatted LaTeX expressions, as demonstrated in Figure 5.4. This feature ensures that complex mathematical notations, scientific formulas, and equations are easily readable and visually appealing in posts.

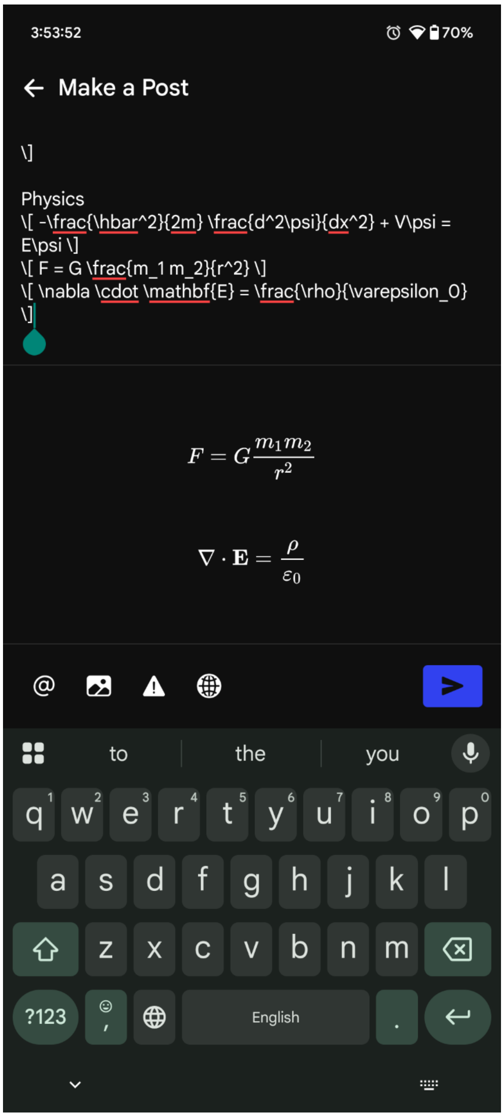

Figure 5.3: Live rendering of latex while writing post

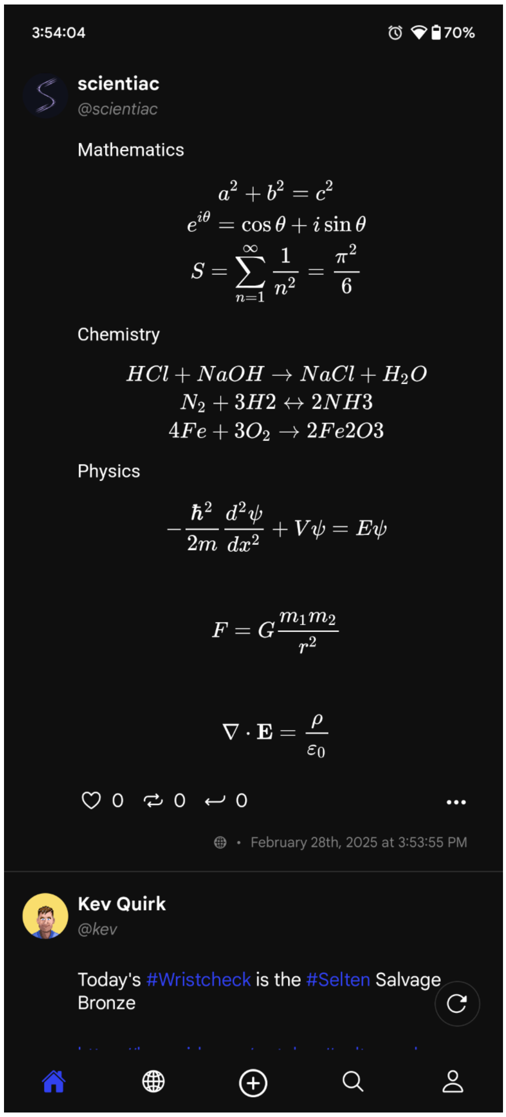

Figure 5.4: Rendered post with latex in timeline

5.3 Interaction with Remote Accounts

User, using their account, can interact with other users on different instances or services using ActivityPub[1] constructs like Follow, Like, Boost, Reply.

5.3.1 Searching and Following Remote Accounts

Users can search for remote accounts by entering the handle associated with an account on a different instance. As shown in Figure 5.5, the search results display matching accounts along with relevant details, allowing users to identify and interact with the desired profile.

Once a remote account is found, users can visit its profile page (Figure 5.6), where they can view posts, follow the account, and engage through actions like liking or boosting posts. This seamless integration enables cross-instance communication within the network.

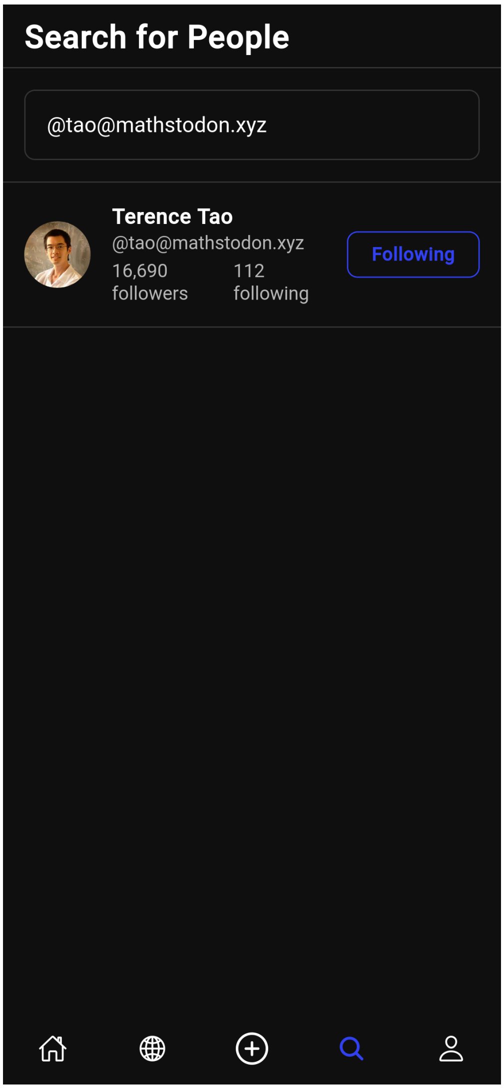

Figure 5.5: Searching and following remote accounts

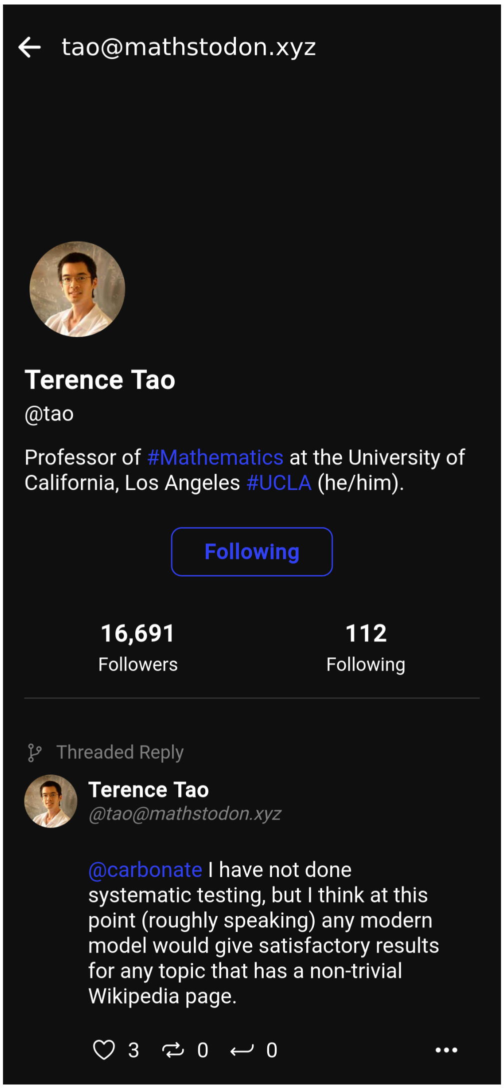

Figure 5.6: Profile page of remote account

5.3.2 Remote Posts: Like, Boost, Reply

Users can interact with remote posts in various ways, including liking, boosting (similar to retweeting), and replying. As shown in Figure 5.7, these actions allow users to engage with posts across different instances seamlessly.

Additionally, users can view all replies to a post in a threaded format. Figure 5.8 illustrates how responses are displayed, ensuring clear and organized discussions.

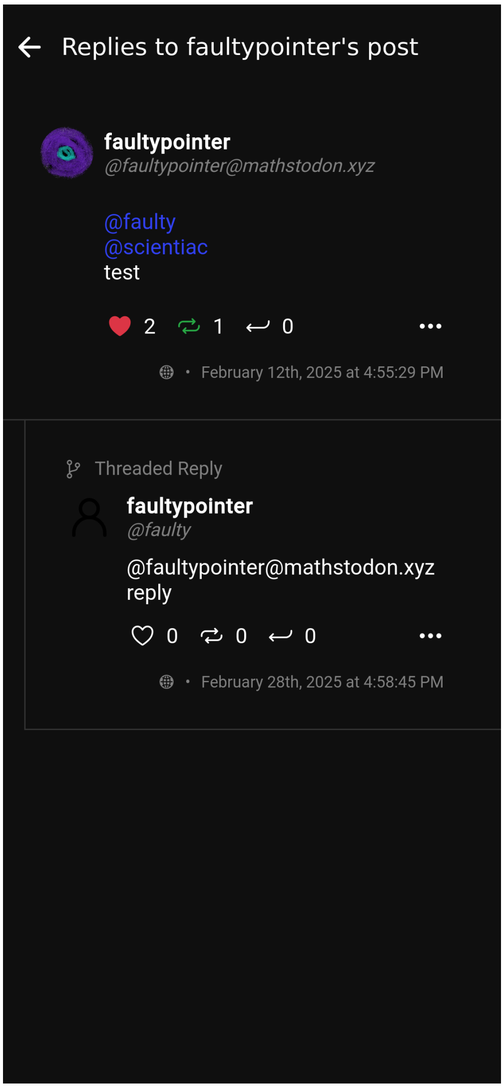

Figure 5.7: Liking, reboosting and replying to a post

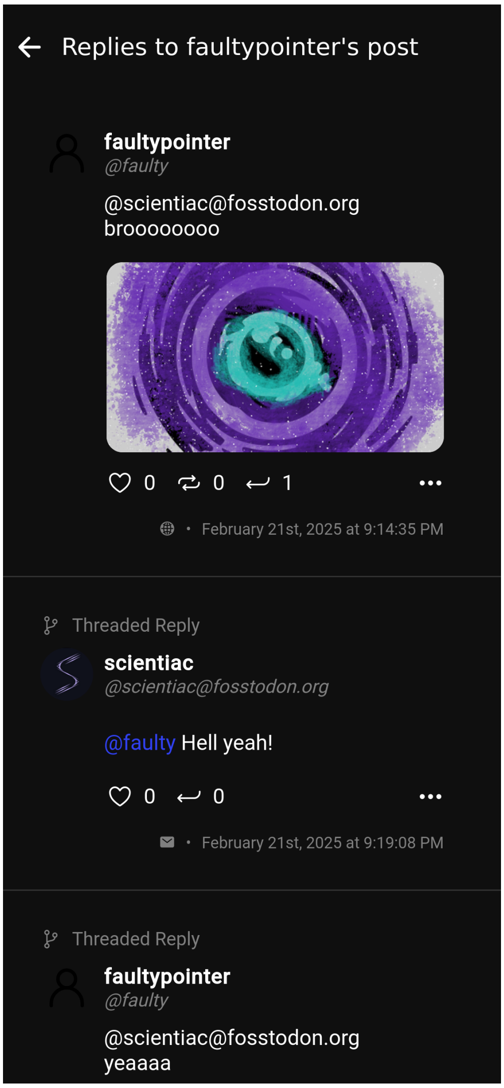

Figure 5.8: Viewing all the replies to a post

5.4 More: Notifications, Followers and Following

Users can keep track of their interactions through notifications, as shown in Figure 5.9, which provides updates on likes, replies, and follows.

Additionally, users can manage their connections through the Following and Followers list (Figure 5.10), where they can see the accounts they are following and those who follow them.

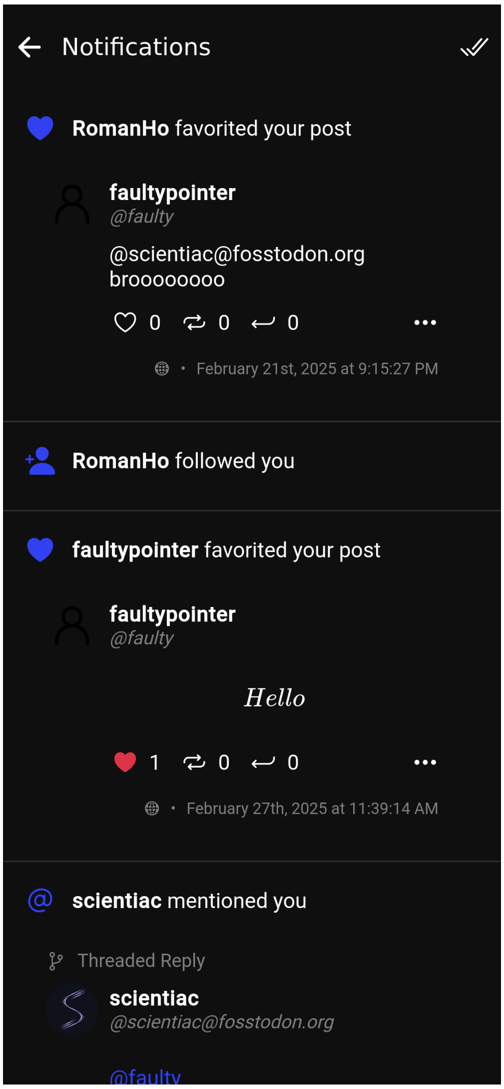

Figure 5.9: Notifications of an account

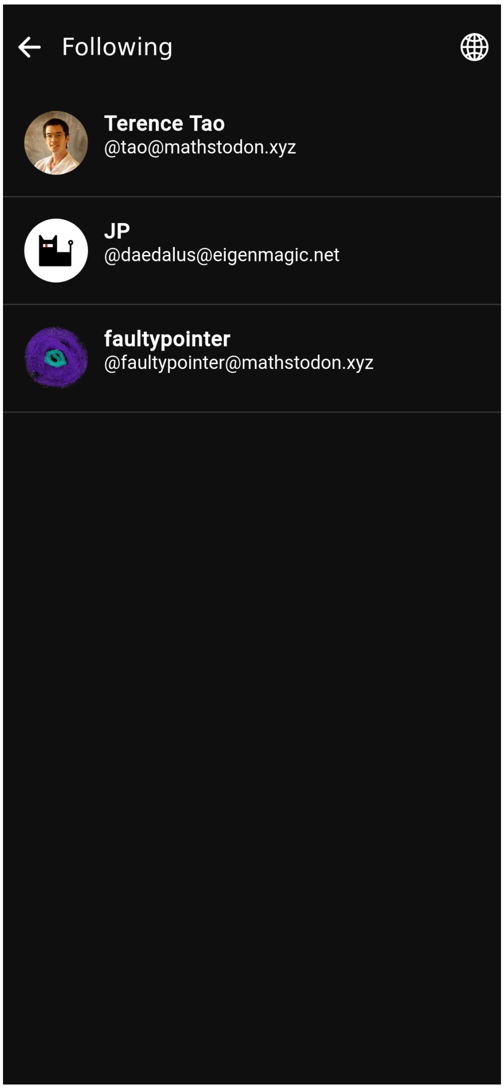

Figure 5.10: Following list

CHAPTER 6

CONCLUSION

In this project, we developed a basic social media server that implements parts of the [1] protocol. This allows users on our platform to interact with others on existing servers that follow the same protocol. In addition to our server, we also designed a clean and clutter-free frontend that supports not only our server but also other similar platforms, such as Mastodon[2].

Users of our frontend can write mathematical text within plain text posts by leveraging the power of LaTeX through MathJax[8].

Our server currently supports a single user, but with the ability to create multiple accounts for different purposes. Despite this single-user limitation, it is still possible to form a broader social network through ActivityPub, enabling communication across federated services. As a result, we have created a social media platform tailored for scientific and mathematical discussions while maintaining interoperability with the existing user base of ActivityPub-compliant services.

6.1 Further Improvments

We have implemented a basic social media platform with minimal interations. As such, there are few improvments to be made:

- The server can be extended to support multiple users.
- The frontend can be enhanced to allow editing of posted content and provide more granular privacy settings.
- The server can be improved with robust moderation tools, such as blocking and filtering.
- LaTeX/MathJax auto-completion can be added to enhance mathematical input in the frontend.

REFERENCES

[1] C. Webber and J. Tallon, "Activitypub," W3C. [Online]. Available: https://www.w3.org/TR/activitypub/

[2] E. Rochko, "Mastodon: Social networking that's not for sale," 2016. [Online]. Available: https://joinmastodon.org

[3] D. Supernault, "Pixelfed: A decentralized photo sharing platform," 2018. [Online]. Available: https://pixelfed.org

[4] "Peertube: An alternative to big tech's video platforms," Framasoft, 2017. [Online]. Available: https://joinpeertube.org

[5] "Lemmy: A federated link aggregator," LemmyNet, 2020. [Online]. Available: https://join-lemmy.org

[6] grunfink, "Snac: A simple, minimalistic activitypub instance," 2023. [Online]. Available: https://codeberg.org/grunfink/snac2

[7] P. E. Jones, G. Salgueiro, M. B. Jones, and J. Smarr, "Rfc 7033: Webfinger," Tech. Rep., September 2013. [Online]. Available: https://tools.ietf.org/html/rfc7033

[8] MathJax Consortium, "MathJax: A JavaScript display engine for mathematics," 2024. [Online]. Available: https://www.mathjax.org/

APPENDIX A

APPENDIX B