

Minglr:Social Media Website for Interactive User Engagement

A PROJECT REPORT

Submitted by

Devottaama Ghosh [RA2211003010479]

Baishali Ghosh [RA2211003010495]

Sarthak Sinha [RA2211003010498]

Under the Guidance of

Dr. K. R. Jansi

Assistant Professor

Department of Computing Technologies

*in partial fulfillment of the requirements for the degree
of*

**BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE ENGINEERING**



**DEPARTMENT OF COMPUTING TECHNOLOGIES
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR- 603 203
MAY 2025**



**Department of Computing Technologies
SRM Institute of Science & Technology
Own Work Declaration Form**

Degree/ Course : B. tech CSE – CSE(CORE)

Student Name : Devottaama Ghosh, Baishali Ghosh, Sarthak Sinha

Registration Number : RA2211003010479, RA2211003010495, RA2211003010498

Title of Work : Minglr:Social Media Website for Interactive User Engagement

We hereby certify that this assessment complies with the University's Rules and Regulations relating to Academic misconduct and plagiarism, as listed in the University Website, Regulations, and the Education Committee guidelines.

We confirm that all the work contained in this assessment is my / our own except where indicated, and that I / We have met the following conditions:

- Clearly referenced / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc)
- Given the sources of all pictures, data etc. that are not my own
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

DECLARATION:

I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, and that I have followed the good academic practices noted above.

Devottaama Ghosh [RA2211003010479]

Baishali Ghosh [RA2211003010495]

Sarthak Sinha [RA2211003010498]

Acknowledgment

We express our humble gratitude to **Dr. C. Muthamizhchelvan**, Vice-Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

We extend our sincere thanks to **Dr. Leenus Jesu Martin M**, Dean-CET, SRM Institute of Science and Technology, for his invaluable support.

We wish to thank **Dr. Revathi Venkataraman**, Professor and Chairperson, School of Computing, SRM Institute of Science and Technology, for her support throughout the project work.

We encompass our sincere thanks to, **Dr. M. Pushpalatha**, Professor and Associate Chairperson - CS, School of Computing and **Dr. Lakshmi**, Professor and Associate Chairperson -AI, School of Computing, SRM Institute of Science and Technology, for their invaluable support.

We are incredibly grateful to our Head of the Department, **Dr. Niranjana G**, Professor, Department of Computing Technologies, SRM Institute of Science and Technology, for her suggestions and encouragement at all the stages of the project work.

We want to convey our thanks to our Project Coordinators, Panel Head, and Panel Members Department of Computational Intelligence, SRM Institute of Science and Technology, for their inputs during the project reviews and support.

We register our immeasurable thanks to our Faculty Advisor, **Dr. C. Muralidhran**, Department of Computing Technologies, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to our guide, **Dr.K.R.Jansi** Department of Computing Technologies, SRM Institute of Science and Technology, for providing us with an opportunity to pursue our project under his / her mentorship. He / She provided us with the freedom and support to explore the research topics of our interest. His / Her passion for solving problems and making a difference in the world has always been inspiring.

We sincerely thank all the staff members of Computing Technology, School of Computing, S.R.M Institute of Science and Technology, for their help during our project. Finally, we would like to thank our parents, family members, and friends for their unconditional love, constant support and encouragement



**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR – 603 203**

BONAFIDE CERTIFICATE

Certified that 21CSP302L - Project report titled “Social Media Website for Interactive User Engagement” is the Bonafide work of “Devottaama Ghosh [RA2211003010479], Baishali Ghosh [RA2211003010495], Sarthak Sinha [RA2211003010498]” who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

K. Jansi
07/05/25

Dr. K.R.Jansi

Assistant Professor
Department of Computing Technologies
SRMIST.

G. Niranjana

Dr. Niranjana G

Professor & Head
Department of Computing Technologies
SRMIST



ABSTRACT

This project aims to design and implement a scalable social media platform tailored for modern digital interaction. The system provides essential functionalities including user registration, login authentication, profile management, content posting (text and media), commenting, liking, and real-time notifications. Built using modern web development technologies such as HTML, CSS, JavaScript (React.js) for the frontend and MySQL for the backend database, the server-side logic is managed through a custom backend deployed locally using XAMPP. The platform architecture emphasizes responsive design for multi-device accessibility, user data privacy, and modularity for easy expansion.

Advanced features such as post moderation, user management, and notification systems are integrated to support both community-based collaboration and administrative oversight. The platform also follows best practices in software engineering, including RESTful API design, database normalization, secure authentication, and scalable component design. Furthermore, the development aligns with ethical digital principles to promote inclusivity, transparency, and digital well-being.

This social media website serves as a foundational system suitable for deployment in schools, colleges, and local communities and is easily adaptable for public or private network use cases. The modular nature of the codebase enables the addition of features like private messaging, tagging, and AI-driven moderation in future phases.

.

TABLE OF CONTENTS

ABSTRACT	i	
TABLE OF CONTENTS	ii	
LIST OF FIGURES	ii	
LIST OF TABLES	iii	
ABBREVIATIONS	iv	
CHAPTER NO.	TITLE	PAGE NO.
1 INTRODUCTION		1
1.1 Introduction to Project		1
1.2 Motivation		1
1.3 Sustainable Development Goal of the Project		2
1.4 Product Vision Statement		2
1.5 Product Goal		4
1.6 Product Backlog (Key User Stories with Desired Outcomes)		6
1.7 Product Release Plan		7
2 SPRINT PLANNING AND EXECUTION		
2.1 Sprint 1		
2.1.1 Sprint Goal with User Stories of Sprint 1		8
2.1.2 Functional Document		12
2.1.3 Architecture Document		14
2.1.4 UI Design		17
2.1.5 Functional Test Cases		18
2.1.6 Daily Call Progress		19
2.1.7 Committed vs Completed User Stories		19
2.1.8 Sprint Retrospective		20

2.2 Sprint 2

2.2.1 Sprint Goal with User Stories of Sprint 2	21
2.2.2 Functional Document	23
2.2.3 Architecture Document	26
2.2.4 UI Design	26
2.2.5 Functional Test Cases	27
2.2.6 Daily Call Progress	28
2.2.7 Committed vs Completed User Stories	29

3. RESULTS AND DISCUSSIONS

3.1 Project Outcomes (Justification of outcomes and how they align with the goals)	30
3.2 Committed vs Completed User Stories	31

4 CONCLUSIONS & FUTURE ENHANCEMENT

32

APPENDIX

33

A. SAMPLE CODING

33

B. PLAGIARISM REPORT

51

LIST OF FIGURES

CHAPTER NO	TITLE	PAGE NO.
Fig 1.1	MS Planner Board	5
Fig 1.2	Release plan	6
Fig 2.1	user story for New User	9
Fig 2.2	user story for Regular User	10
Fig 2.3	User story for As a Privacy-Conscious User	11
Fig 2.4	System Architecture Diagram	15
Fig 2.5	UI Design for Landing page	17
Fig 2.6	UI design	17
Fig 2.7	Bar graph for Committed Vs Completed User Stories	19
Fig 2.8	User Story for As a User who want interaction	22
Fig 2.9	System Artitecture	26
Fig 2.10	UI Design-feed	26
Fig 2.11	UI Design-Account	27
Fig 2.12	Bar Graph for Commited vs Completed User Story	29
Fig 2.13	Bar Graph for Commited vs Completed User Stories	31

LIST OF TABLES

CHAPTER NO	TITLE	PAGE NO.
Table 1.1	Product Backlog	4
Table 2.1	Detailed User Stories of sprint 1	8
Table 2.2	Access level Authorization Matrix	14
Table 2.3	Detailed Functional Test Case	18
Table 2.4	Standup meetings	19
Table 2.5	Sprint Retrospective for the Sprint 1	20
Table 2.6	Detailed User Stories of Sprint2	21
Table 2.7	Access Level Authorization	24
Table 2.8	Detail Functional Test Case	27
Table 2.9	Standup Meetings	28

ABBREVIATIONS

Abbreviation	Full Form
UI	User Interface
UX	User Experience
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
JS	JavaScript
PHP	Hypertext Preprocessor
DB	Database
SQL	Structured Query Language
CRUD	Create, Read, Update, Delete
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
URL	Uniform Resource Locator
DOM	Document Object Model
JWT	JSON Web Token
CSV	Comma-Separated Values
FTP	File Transfer Protocol

CHAPTER 1

INTRODUCTION

1.1 Introduction to Project:

Social media platforms are essential tools for modern communication, collaboration, and expression. This project introduces a scalable and secure social media website that enables users to interact in real-time by sharing posts, liking content, and participating in discussions through comments. Unlike traditional media platforms, this system allows users to create personalized profiles, manage content visibility, and connect with other users in a privacy-focused digital environment.

The backend infrastructure utilizes MySQL for structured data storage and is deployed locally using XAMPP, while the frontend is designed using HTML, CSS, and React.js to ensure a responsive and intuitive user experience. This social media system places high priority on accessibility, modularity, and the ethical handling of user data, making it suitable for use in educational institutions, private communities, or even scaled public deployments. The goal is to build a versatile and future-ready platform that is adaptable, easy to maintain, and aligned with user expectations in today's dynamic web landscape.

1.2 Motivation

As internet access expands globally and digital interaction becomes more deeply embedded in daily life, the need for decentralized, community-oriented social platforms increases. Existing platforms often lack customizable privacy, academic use orientation, or institutional control. Our motivation lies in bridging this gap by creating a platform that empowers users with control over their data while offering the core benefits of social networking — connection, sharing, and collaboration.

This project is designed to serve a wide range of user demographics including students, educators, and community groups who require a safe, efficient, and easy-to-use social media tool. With a user-first approach and modern web development practices, we aim to encourage digital expression while maintaining a secure and respectful environment for interaction.

1.3 Sustainable Development Goal of the Project

This project contributes to:

- **SDG 9 (Industry, Innovation, and Infrastructure):** By leveraging modern web technologies to create an innovative, adaptable digital infrastructure for communication.
- **SDG 16 (Peace, Justice, and Strong Institutions):** By providing a platform for inclusive, respectful, and accountable digital interaction, protecting user rights and privacy.

The project enables digital inclusion by empowering underserved communities to build and manage their own online spaces. It can be used as a foundation for civic engagement, remote education, and public awareness, promoting safe and structured digital interactions

1.4 Product Vision Statement

Audience:

- **Primary:** Students, educators, and administrators in schools, colleges, and training institutions.
- **Secondary:** Community group leaders, small businesses, and non-profit organizations requiring a tailored online social platform.

Needs:

- Real-time interaction tools (comments, likes, notifications)
- User authentication and profile customization
- Content moderation tools and admin access
- Mobile-responsive user interface
- Secure storage and processing of personal data

Product:

- A secure, modular social media website with functionality for content posting, user connections, notifications, and media sharing.

- Additional features such as report generation, keyword filtering, and post visibility settings.

Values:

- **Accessibility:** Interface design accommodates users across devices and abilities.
- **Transparency:** Users understand how their data is stored and shared.
- **Scalability:** Platform can grow with user demand through modular architecture.
- **Security:** All data exchanges follow secure practices and encrypted storage.

1.4 Product Goal

To develop and deploy a responsive and secure social media website that supports interactive, real-time digital communication for communities and institutions. The platform will focus on user engagement, intuitive design, and data protection, while providing features such as:

- Real-time post creation, likes, and comments
- Notification system for post interactions
- Profile and friend management tools
- Administrative panel for moderation

The system is designed to serve as a standalone solution or integrate with existing educational or organizational systems. Future enhancements may include analytics dashboards, private messaging, and language localization support.

1.5 Product Backlog

Table 1.1 Product Backlog

Product Backlog sample for Social media application							Functional Requirements	Non-Functional Requirements	Original Estimate	Actual Effort (In days)
ID	Title	Epic	User Story	Priority (MoSCoW)	Status	Acceptance Criteria				
1	User Authentication	Authentication	As a user, I want to securely access the app so that my personal information and data remain protected	Must	In Progress	1. The user can register with a valid email and password. 2. User receives a confirmation email for account activation. 3. User can log in with registered credentials. 4. Password reset functionality is secure and straightforward.	Secure password hashing and storage - Account lockout after multiple failed login attempts.	Response time for authentication actions should be less than 2 seconds. Password reset emails should be delivered within 5 minutes.	5 days	4 days
2	Profile Customization	Personalization	As a mobile user, I want a seamless experience so that I can navigate and use the app efficiently without any interruptions or difficulties	Should	Pending	1. UI/UX is responsive on various screen sizes. 2. All features are accessible and functional on mobile devices. 3. Testing on multiple devices validates responsiveness.	Implement a mobile-friendly layout and navigation - Ensure touch-friendly interactions.	Page load time on mobile devices should be less than 3 seconds - App should support the latest versions of major mobile browsers.	5 days	5 Days
3	Content Sharing	Social Sharing	As a user, I want to share app content on social media so that I can easily spread interesting or useful information with my friends, family, or followers	Could	Backlog	1. User can share posts on linked social media accounts. 2. Shared posts display correctly on the respective social media platforms. 3. Social media API responses are handled gracefully.	Integrate with social media APIs (e.g., Facebook, Twitter) - Implement sharing options in the app's UI.	Response time for social media API calls should be less than 1 second - Handle API rate limits and errors effectively.	7 days	
4	Messaging System	Communication	As a global user, I want to use the app in my preferred language so that I can comfortably understand and interact with the content and features provided	Must	Ready for Dev	1. User can select preferred language in settings. 2. All app content is displayed in the selected language. 3. Translations are accurate and culturally sensitive.	Implement language selection in user settings - Use localization for all UI text and content.	Translation process should be documented and repeatable - Support at least 5 major languages.	7 days	
5	Friend Requests	Social Network	As a user, I want to make and receive app so that I can quickly accomplish my tasks without experiencing delays	Must	In Testing	1. App loads within 3 seconds on average. 2. Smooth navigation between screens without delays. 3. Load testing confirms stability under peak usage.	Optimize database queries and indexing - Implement caching for frequently accessed data.	Response time for critical user actions should be less than 2 seconds - Monitor and optimize server resource usage.		
6	News Feed Algorithm	Content Discovery	As a user, I want to use the app without an internet connection so that I can access its features and information even when I am offline	Should	Backlog	1. Users can access saved content without an internet connection. 2. Offline changes sync seamlessly when the device reconnects. 3. Conflict resolution is handled gracefully in case of sync issues.	Implement local storage for offline content - Develop a syncing mechanism for offline changes.	The syncing process should be smooth for the user - Offline mode should be functional for at least 80% of app features.		
7	Video Streaming	Multimedia	As a user, I want to provide feedback on the app so that the developers can improve the user experience and address any issues	Could	Backlog	1. Users can submit feedback through the in-app feedback form. 2. Feedback is recorded and visible to admin in the admin dashboard. 3. Feedback form includes necessary user information for follow-up.	Create a feedback form in the app - Implement a backend system to store and retrieve feedback.	Feedback should be stored securely and be accessible only to authorized personnel - Admins should receive email notifications for new feedback submissions.		
8	Notifications	Alerts	As a stakeholder, I want to gather insights from user data so that I can make informed decisions to enhance the app's features and overall performance	Could	Ready for Dev	1. Integration with Google Analytics is successful. 2. Key user interactions and events are tracked. 3. Analytics dashboard provides actionable insights.	Implement Google Analytics SDK in the app - Define custom events for important user actions.	Analytics data should be updated in near real-time - Regularly review and analyze analytics data for decision-making.		
9	Content Moderation	Safety	As a user, I want a bug-free experience so that I can use the app smoothly without encountering any issues or disruptions	Must	In Progress	1. Resolved login error for users with special characters in passwords. 2. Fixed image loading issue in the gallery. 3. Regression testing confirms bug fix.	Identify and fix specific bugs reported by users and testing - Implement automated tests to prevent regression.	Critical bugs should be resolved within 48 hours of discovery - Implement a bug-tracking system for efficient reporting and resolution.		
10	Analytics Dashboard	Analytics	As a user, I want to find content easily so that I can quickly access the information I need without wasting time searching	Should	Backlog	1. Search results are relevant and displayed in order of relevance. 2. Filters work correctly, refining search results. 3. User testing validates improved search experience.	Improve search algorithms for relevance - Enhance filtering options based on user feedback.	Search queries should return results within 2 seconds - User satisfaction with search functionality should increase by 20% based on feedback.		

The Social Media Website for Interactive User Engagement product backlog is a well-organized and carefully prioritized collection of features, user-centric functionalities, and technical enablers essential to building a dynamic, scalable, and user-friendly social networking platform. The goal is to ensure users experience seamless interaction, personalized content, and robust social connectivity, while maintaining performance, security, and reliability.

Key backlog components include User Authentication, Profile Customization, Content Sharing, Messaging, Friend Request Handling, Notifications, Real-Time Video Streaming, Content Moderation, and Analytics Dashboard. Each backlog item is anchored in a clearly defined **User Story** and is accompanied by comprehensive Functional and Non-Functional Requirements, detailed Acceptance Criteria, and Effort Estimations. This fosters full transparency in sprint planning and development tracking.

The backlog is structured using the **MoSCoW** prioritization method (Must Have, Should Have, Could Have, Won't Have for now) to ensure iterative progress with focus on the most critical and high-impact features first. This agile framework supports continuous improvement and responsive adaptation based on user feedback, testing insights, and stakeholder alignment.

The screenshot shows a Microsoft Planner board titled "SOCIAL MEDIA PLATFORM". The board is organized into four columns:

- PRODUCT BACKLOG REFINEMENT(USER)**: Contains a task for "USER STORY-1(As a New User)" with a sub-task "User story.Sepm-1.pdf".
- SPRINT BACKLOG REFINEMENT(In Prog)**: Contains a task for "SPRINT DOCUMENT" with a sub-task "SEPM_Sprint RETROSpective 5.xlsx".
- AWAITING REVIEW**: Contains a task for "Social Media Project" with a sub-task "Social Media Project.pptx".
- COMPLETED ITEMS**: Shows one completed task labeled "Completed tasks 1".

Each task card includes a timeline at the bottom with boxes for week 2, week 3, week 4, good, and m. There are also various icons and status indicators such as "Filters", "Group by Bucket", and "Share".

Fig 1.1 MS Planner Board

Similar to the product backlog adoption strategy, integrating Microsoft Planner into the development lifecycle of the Social Media Website for Interactive User Engagement project serves as a powerful and intuitive way to organize, prioritize, and execute tasks. Each feature—ranging from user authentication and content sharing to messaging and analytics—has been structured as an independent task, aligned with its corresponding user story, development phase, and MoSCoW priority classification (Must Have, Should Have, Could Have, Won't Have for now).

This planner-based project management approach ensures end-to-end visibility across all stages of development. It empowers the development team to stay aligned, track real-time progress, and seamlessly communicate updates, issues, and dependencies throughout the entire development lifecycle.

By leveraging the visual task management and board-style interface of Microsoft Planner, complex modules—such as real-time chat, news feed algorithms, profile customization, and content moderation—are broken down into manageable, trackable

subtasks. From database integration to UI/UX implementation, every development unit is clearly defined, monitored, and linked to sprint goals.

This structured workflow not only promotes team collaboration and transparency, but also enhances minimizes bottlenecks, and accelerates the release of a robust, user-friendly social media platform packed with interactive, scalable features—delivered on time and with quality at its core.

1.7 Product Release Plan

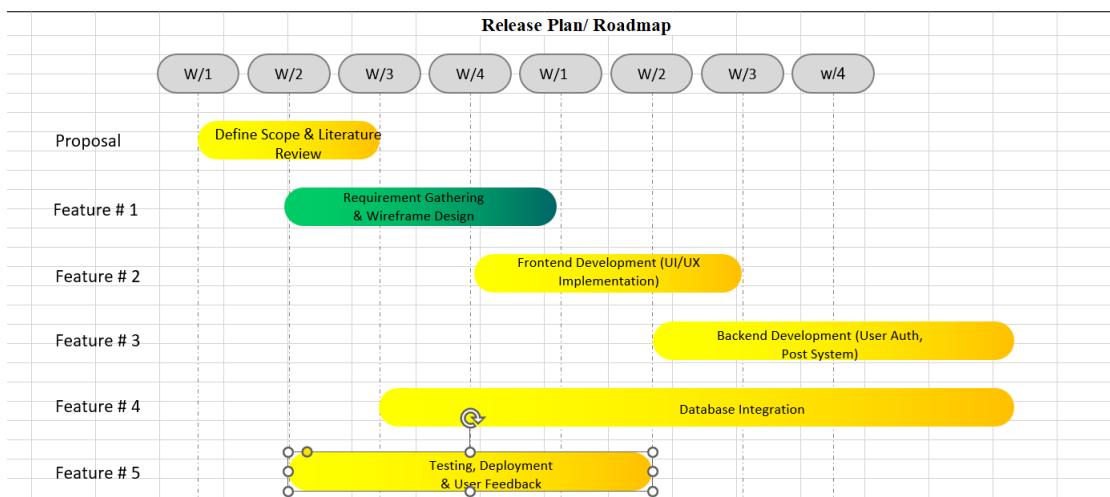


Fig 1.2 Release plan

The deployment plan for the Social Media Website for Interactive User Engagement is designed to ensure a structured, iterative, and timely rollout of all major features and supporting components across the development lifecycle. Beginning with project proposal approval and backlog definition in Week 1, the roadmap proceeds into the implementation of foundational modules such as user authentication, profile customization, and content sharing functionality.

Subsequent phases focus on integrating key engagement drivers, including the real-time messaging system, friend request logic, and notification services, ensuring that dynamic user interaction is at the heart of the platform. Mid-stage development incorporates advanced personalization elements like the news feed recommendation algorithm, user content streaming, and moderation tools to maintain content quality and community standards.

In the final stages, emphasis is placed on deploying analytics and tracking mechanisms, data privacy and security features, and admin-side **tools** for content oversight and

system management. The deployment concludes with cross-device testing, performance tuning, and user acceptance validation to ensure the platform is responsive, secure, and scalable across various environments.

This phase-driven deployment structure encourages consistent team alignment, promotes transparency, and supports agile adaptation based on feedback and testing. It ensures a cohesive, feature-rich, and user-centric platform is delivered effectively and within the stipulated project timeframe

CHAPTER 2

SPRINT PLANNING AND EXECUTION

2.1 Sprint 1

2.1.1 Sprint Goal with User Stories of Sprint 1

Table 2.1 Detailed User Stories of sprint 1

Sprint	User Story
Requirement Analysis & Planning	US1: Define functional and non-functional requirements of the social media platform US2: Identify system components such as frontend, backend , and database (MySQL).
Database Design & Setup	US3: Design normalized database schema for users, posts, comments, and likes. US4: Implement database in MySQL using XAMPP.
Backend Setup	US5: Set up custom backend API to handle user registration and authentication. US6: Implement post creation, deletion, and retrieval logic.
Frontend Development	US7: Create login/register UI and connect to backend. US8: Design and implement the post feed and profile view in React.js.
Testing & Debugging	US9: Test basic functionalities (register, login, post) and ensure expected behavior. US10: Debug UI/backend interaction and fix data handling issues.

Planner Board representation of user stories are mentioned below figures

The image shows a Planner Board card for a user story titled "USER STORY-1(As a New User)". The card has a dark background with white text and icons. At the top, it says "SOCIAL MEDIA PLATFORM". Below the title is a circular icon with a person symbol and the text "Assign". Under the title, there are four colored buttons: pink (week 2), yellow (week 3), green (week 4), and blue (good). To the right of these are two more buttons: a pink one with "m" and a blue one with an "X". Below these buttons are sections for "Bucket" (set to "PRODUCT BACKLOG REFIN..."), "Progress" (set to "In progress"), and "Priority" (set to "Medium"). There are also sections for "Start date" (set to "Start anytime") and "Due date" (set to "Due anytime"). The "Repeat" section indicates "Does not repeat". On the left side of the card, under the heading "User Story 1 - As a New User", is the story text: "As a new user, I want to sign up using my email or social media accounts so that I can easily create an account and start using the platform. This feature simplifies the onboarding process for new users. Users can register using multiple methods (email, Google, Facebook) to accommodate convenience and preference, allowing quick access to the platform's features." Below this text are several questions and their answers: "Who is this for?" (New users or visitors to the platform), "What are they trying to achieve?" (Quickly create an account and access social features), "Overall benefit?" (Increases user onboarding efficiency and enhances user satisfaction with seamless sign-up), and "# Linked Tasks" (which is currently empty). At the bottom right of the card is a checkbox labeled "Show on card".

Fig 2.1 user story for New User

US1: Enables new users to quickly sign up using email or social media (Google, Facebook), ensuring fast and convenient access to platform features.

US2: Defines key functional (multi-method registration, secure login) and non-functional (speed, mobile responsiveness, usability) requirements to ensure smooth onboarding and high user satisfaction.

SOCIAL MEDIA PLATFORM

○ **USER STORY-2(As a Regular User)**

Assign

week 2 week 3 week 4 good m

Bucket	Progress	Priority
PRODUCT BACKLOG REFIN...	In progress	Medium
Start date	Due date	Repeat
Start anytime	Due anytime	Does not repeat

Notes Show on card

User Story 2 - As a Regular User

As a regular user, I want to create and publish posts with text, images, or videos so that I can share updates with my friends.

This allows users to express themselves through multimedia content and interact with their social network effectively.

Who is this for?
 Active users of the platform who engage regularly.

What are they trying to achieve?
 Share their thoughts, experiences, or media with others.

Overall benefit?
 Enhances user engagement and content richness on the platform.

Linked Tasks

- Implement post editor with text, image, and video support
- Add media upload functionality

Fig 2.2 user story for Regular User

From above fig 2.2 we conclude that :

US3: Focused on retrieving and organizing a high-quality fundus image dataset for training and evaluation purposes.

US4: Involves executing essential image preprocessing tasks such as resizing, normalization, and enhancement to prepare data for deep learning model input.

SOCIAL MEDIA PLATFORM

○ USER STORY-4(As a Privacy-Conscious User)

Assign

week 2 X week 3 X week 4 X good X should have X m X

Bucket	Progress	Priority
PRODUCT BACKLOG REFIN...	In progress	Medium
Start date	Due date	Repeat
Start anytime	Due anytime	Does not repeat

Notes Show on card

User Story 4 - As a Privacy-Conscious User

As a privacy-conscious user, I want to control who can see my posts so that I can protect my content.

User privacy is crucial for platform trust and long-term engagement.

Who is this for?
Users who care about privacy and audience control.

What are they trying to achieve?
Decide whether to share posts publicly, with friends, or only themselves.

Overall benefit?
Improves user confidence in using the platform for both personal and public sharing.

Linked Tasks

- Add visibility settings to post creation
- Implement access filters for post visibility

When you click on the link, a push notification will appear.

Fig 2.3 User story for As a Privacy-Conscious User

From above fig 2.3 we conclude that :

US5: As a privacy-conscious user, I want to control who can see my posts so I can protect my content

2.1.2 Functional Document

2.1.2.1. Introduction

This project aims to build a robust, scalable social media website where users can interact through posts, likes, comments, and notifications. The system is designed with modular architecture for the frontend, MySQL for the backend database, and XAMPP for local deployment. It emphasizes accessibility, user privacy, responsive design, and ease of use, particularly for educational and community-based use cases. The system is developed to support future scalability, security enhancements, and feature extensions.

2.1.2.2. Product Goal

The main objective of this project is to develop a reliable, user-friendly, and extensible social media platform that:

1. Enables content creation and interaction (posts, likes, comments)
2. Supports real-time user engagement with notifications
3. Provides a privacy-conscious experience through visibility settings
4. Facilitates community moderation and administrative controls

2.1.2.3. Demography

Users:

Students, educators, and local community members looking for a secure and controlled social platform.

Location:

Educational institutions, colleges, and small organizations globally.

2.1.2.4. Business Processes

User Registration and Login:

- Secure sign-up and sign-in using custom backend APIs
- Authentication token management for session persistence

Content Management:

- Create/edit/delete posts with optional media uploads
- Like and comment on others' posts

- Fetch real-time updates and notifications

2.1.2.5. Features

This project focuses on implementing the following key features:

Feature 1: Post Creation with Visibility Settings

- Description: Users can create posts with text/media and control visibility (public, friends-only, private)
- User Story: As a privacy-conscious user, I want to choose who sees my post so I can protect my content.

Feature 2: Notification System

- Description: Real-time alerts for likes, comments, and mentions to enhance engagement.
- User Story: As an active user, I want to be notified of interactions so I can stay connected.

Feature 3: Admin Moderation Tools

- Description: Admin dashboard to manage user reports, remove inappropriate content, and view platform metrics.
- User Story: As an admin, I want to moderate the platform efficiently to ensure a safe environment.

Feature 4: Profile Management

- Description: Users can update their profile photo, name, bio, and manage their connections.
- User Story: As a user, I want to personalize my profile to express my identity.

Feature 5: Secure Authentication

- Description: Uses session/token-based login and form validation to ensure data integrity.
- User Story: As a registered user, I want to log in securely and access my personalized content.

2.1.2.6. Authorization Matrix

Table 2.2 Access level Authorization Matrix

Role	Access Level
Developer	Full access to application code, server configuration, database schema, and debug logs.
Administrator	Access to user management tools, content moderation panel, analytics dashboard.
Registered User	Access to post creation, liking, commenting, profile updates.
Guest User	Read-only access to public posts and profile previews without account creation.

2.1.2.7. Assumptions

- The platform will be developed and tested in a local development environment using XAMPP.
- User data will be stored in MySQL with access handled securely through session-based authentication.
- The application will have a web-based frontend built in React.js; no mobile application is planned in this phase.
- All media and personal data will be managed in accordance with general privacy best practices (e.g., user-controlled visibility settings).
- The platform will follow a modular structure for scalability and maintainability.
- Admins and moderators are responsible for maintaining content integrity through internal policies.

2.1.3 Architecture Document

2.1.3.1. Application

Microservices:

While not a microservices-based application, the platform is modular in nature and follows a monolithic deployment model. Each logical function is organized into independently testable modules to support maintainability and feature upgrades.

- User Authentication Module:
Handles sign-up, login, session tokens, and access control for various user roles.
- Content Management Module:
Manages creation, editing, deletion, and retrieval of user-generated posts and media.
- Engagement Module:
Handles interactions like likes, comments, and follower connections. Supports post visibility rules (e.g., public, friends-only).
- Notification Module:
Triggers real-time alerts for post interactions using client-side event listeners or polling mechanisms.
- Moderation Module:
Enables administrators and moderators to manage flagged content, suspend users, and oversee platform health.
- Analytics & Logging Module:
Tracks user behavior and application events to assist in future feature planning and performance evaluation.

2.1.3.2 System Architecture-

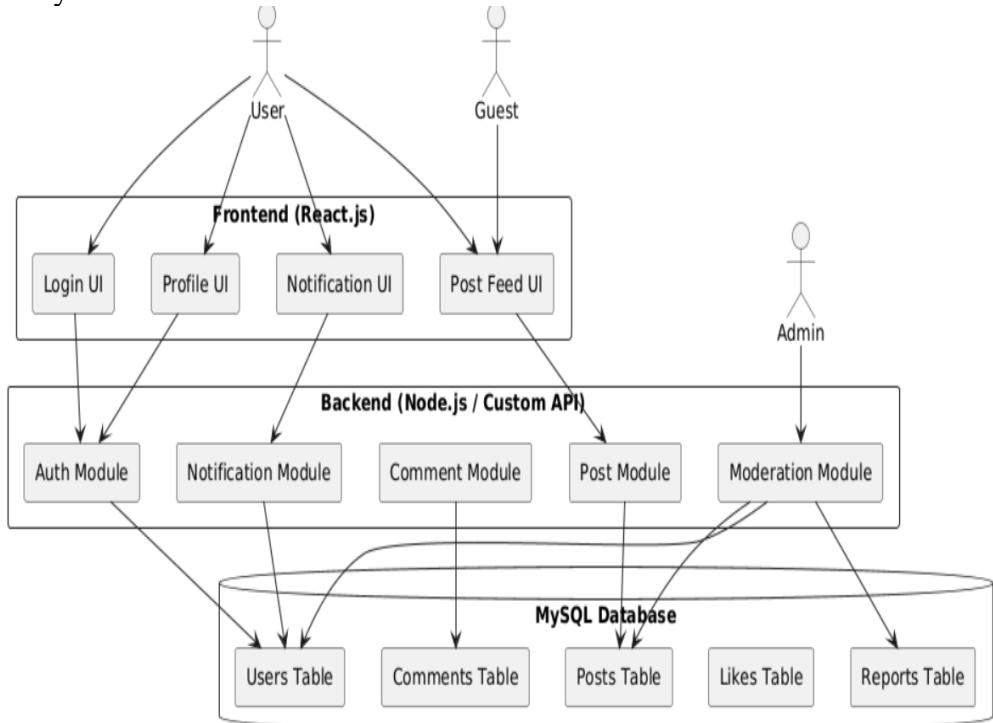


Fig 2.4 System Architecture Diagram

2.1.3.3. Data Exchange Contract:

Frequency of Data Exchanges:

- Data exchanges are designed to ensure efficiency, user responsiveness, and data consistency across the platform.

Real-Time Exchanges:

- User interactions such as login, post creation, likes, and comments are processed instantly via HTTP requests between the frontend and backend.
- Notification updates are also triggered and sent in real time or through polling intervals.

Periodic Syncs:

- Admin logs, usage reports, and flagged content reports are periodically reviewed or exported for audit or moderation purposes.
- Scheduled cleanup scripts (e.g., inactive users, expired sessions) may run daily or weekly.

Data Sets:

- The system handles multiple data categories essential for core functionality and user experience:

Mode of Exchanges (API, File, Queue, etc.):

- The system uses the following mechanisms for data exchange:

API (HTTP):

RESTful APIs enable communication between frontend (React.js) and backend (Node.js or equivalent). Endpoints are used for authentication, post management, interaction logging, and user profile updates.

Database Queries (SQL):

MySQL is used for all persistent storage operations. The backend communicates with the database using SQL queries for read/write/update/delete operations.

File Uploads:

Media files (profile pictures, post images) are stored in a local directory or cloud service (in future extensions), with references stored in the database.

2.1.4 UI DESIGN

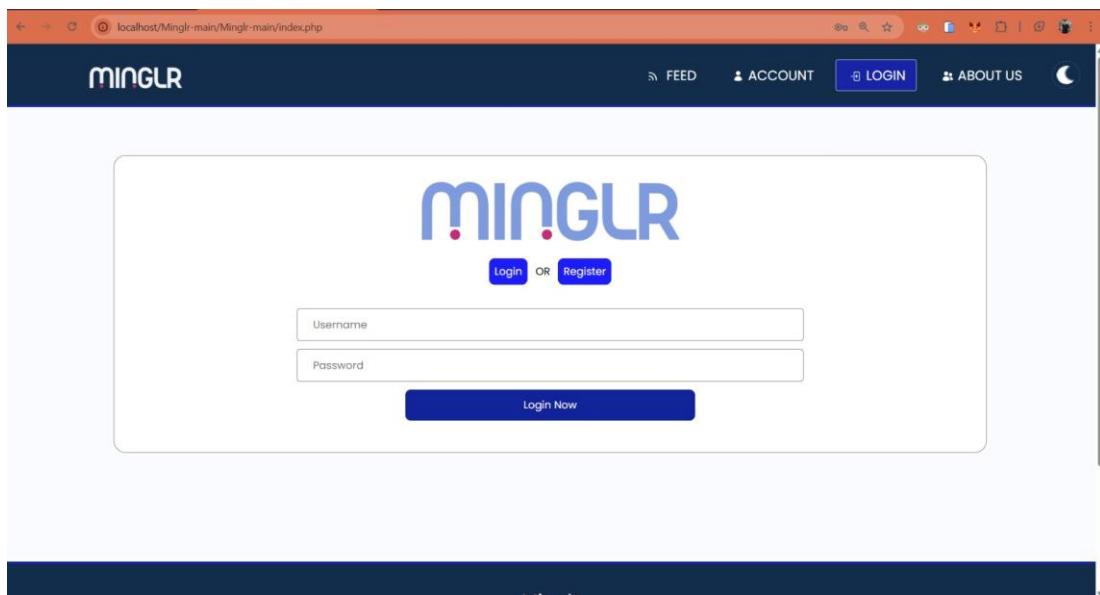


Fig 2.5 UI Design for Landing page

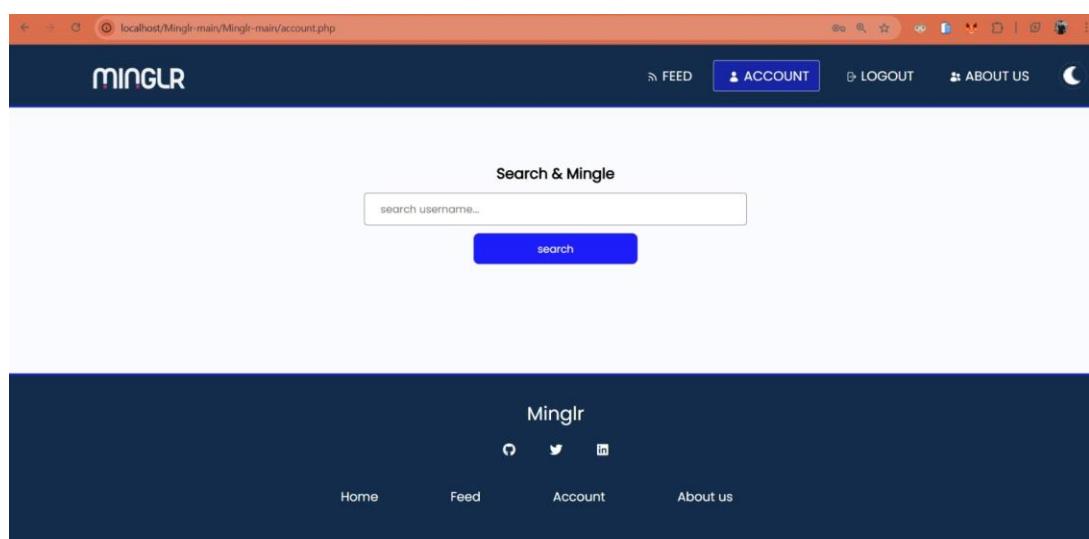


Fig 2.6 UI design

2.1.5 Functional Test Cases

Table 2.3 Detailed Functional Test Case

A	B	C	D	E	F	G	H
Feature	Test Case	Steps to Execute Test Case	Expected Output	Actual Output	Status	More Information	
User Registration	Valid Registration	1. Enter valid user details2. Submit the form	Account is created and redirected to login	Pass	Pass	Verify input validation and duplicate user checks	
Login Authentication	Valid Login	1. Enter correct username/password2. Click Login	Dashboard is loaded with session started	Pass	Pass	Ensure session token is generated	
Post Creation	Text and Media Post	1. Write post with/without image2. Submit	Post is stored and displayed in feed	Pass	Pass	Test both text-only and image posts	
Post Visibility	Friends-Only Visibility	1. Set visibility to 'Friends Only'2. Post content	Only friends can view the post	Pass	Pass	Confirm with friend and non-friend accounts	
Interaction	Like & Comment	1. Like and comment on a post2. Check feedback	Like and comment are saved and displayed	Pass	Pass	Ensure no duplicate likes, proper comment thread	
Notifications	Interaction Alert	1. Like/comment on another user's post	Notification is sent to original poster	Pass	Pass	Notifications should be real-time or on refresh	
Admin Moderation	Delete Reported Post	1. Admin views reported post2. Click delete	Post is removed and log is updated	Pass	Pass	Verify only admins can take this action	
Profile Management	Profile Update	1. Change name or bio2. Save changes	Profile reflects updates instantly	Pass	Pass	Confirm changes are persistent on refresh	
Fault Tolerance	Invalid Media Upload	1. Upload unsupported file format (.exe) in post	Error is thrown: "Unsupported file type"	Pass	Pass	Prevent security vulnerabilities	
Data Integrity	Direct URL Access (Unauth)	1. Try accessing private user data via direct URL without login	Redirect to login or error page	Pass	Pass	Verify session enforcement	

This above Table 2.3 encapsulates the critical tests conducted for the Social Media Website platform. It verifies:

- User registration, login, and session handling are functioning securely and as expected.
- Content creation (text/media), post visibility control, and social interactions (likes/comments) operate smoothly.
- Notifications are triggered appropriately and reflect real-time engagement.
- Administrative tools like content moderation and user management are reliable and enforce platform rules.
- Fault tolerance is ensured by handling invalid inputs gracefully.
- User profile updates persist correctly, and private data remains protected from unauthorized access.

2.1.6 Daily Call Progress

Table 2.4 Standup meetings

Date	Participation	Discussion Focus	Progress Summary
01-Feb-2025	Full Team	Sprint Planning and feature scoping	Product backlog created, responsibilities assigned
03-Feb-2025	Dev + UI	Frontend layout for login, registration	Initial UI wireframes designed using Figma
06-Feb-2025	Dev + Backend	Database schema finalization (users)	MySQL schema designed and integrated
08-Feb-2025	Full Team	API routing and backend logic discussion	Authentication and CRUD API endpoints defined
10-Feb-2025	Dev + QA	Post visibility and access control logging	Visibility filter implemented in post creation
14-Feb-2025	Admin + Dev	Admin dashboard feature planning	Mockups created for post moderation
20-Feb-2025	QA + Dev	Functional testing and test case review	Drafted test cases for all core modules
28-Feb-2025	Full Team	Mid-sprint sync and issue resolution	Identified integration issues, scheduled for review

2.1.7 Committed Vs Completed User Stories

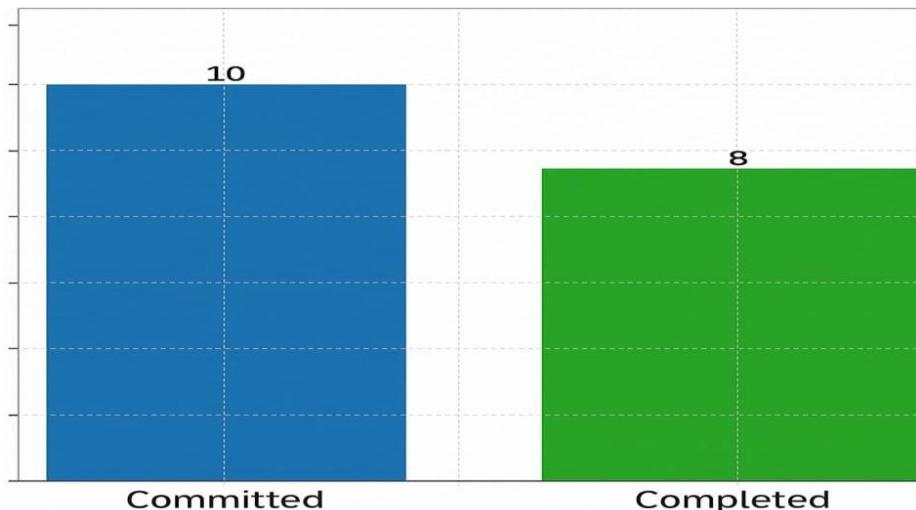


Figure 2.7 Bar graph for Committed Vs Completed User Stories

2.1.8 Sprint Retrospective

Table 2.5 Sprint Retrospective for the Sprint 1

A What went well	B What went poorly	C What ideas do you have	D How should we take action	E
User registration and login worked smoothly	Initial bugs in post visibility filtering	Improve visibility filter logic and apply access checks at DB level	Add visibility rules to API and test role-specific access cases	
Post creation and visibility settings were intuitive	Notification module delay in update sync	Optimize polling or implement WebSockets for faster updates	Upgrade notification flow with real-time broadcasting (e.g., Socket.io)	
Real-time notifications were responsive	Missing input validation on comment field	Add client-side validation and error prompts for all inputs	Integrate validation library (e.g., Yup) for frontend forms	
Profile editing was user-friendly	Difficulty testing admin features across roles	Simulate multiple roles in testing environments	Create mock roles in test scripts to simulate admin/mod scenarios	
Admin moderation tools functioned as expected	Inconsistent file upload handling for large media	Compress and resize images on frontend before upload	Use JavaScript image compression and max size limits on upload	

From Table 2.5, we conclude that the core features of the platform—such as user authentication, post creation, and real-time notifications—functioned smoothly and intuitively. The visibility controls and profile management system also performed reliably. However, several challenges were encountered during the sprint. These included inconsistencies in post visibility filtering logic, occasional delays in the notification module, and insufficient input validation on user interactions. Admin-level testing proved complex due to role simulation limitations, and large media uploads were not handled uniformly. These findings emphasize the need for enhanced role-based testing, frontend validation, and media optimization strategies in the next sprint.

2.2 SPRINT 2

2.2.1 Sprint Goal with User Stories of Sprint 2

Table 2.6 Detailed User Stories of Sprint2

Sprint	User Story
Privacy & Visibility Features	US11: Implement advanced post visibility settings (public, friends-only, private). US12: Allow users to manage visibility of past posts from profile settings.
Notification Enhancements	US13: Enable real-time notifications for likes, comments, and follows. US14: Add notification badge and timestamping on alerts.
Testing – Functional and Usability	US15: Perform comprehensive UI/UX testing across device and browsers. US16: Validate database integrity and error handling across modules.
Feedback Collection	US17: Collect feedback from student and faculty testers. US18: Analyze UX reports and adjust frontend layout for accessibility.
Deployment and Maintenance	US19: Prepare system for deployment using XAMPP and remote DB backup. US20: Monitor performance, fix bugs, and maintain version control system.

SOCIAL MEDIA PLATFORM

○ USER STORY-3(As a User who wants Interaction)

Assign

week 2 X week 3 X week 4 X good X m X

Bucket	Progress	Priority
PRODUCT BACKLOG REFIN...	In progress	Medium

Start date	Due date	Repeat
Start anytime	Due anytime	Does not repeat

Notes Show on card

User Story 3 - As a User Who Wants Interaction

As a user, I want to like, comment, and share posts so that I can engage with content shared by others.

Interactive elements increase user participation and build stronger community ties.

Who is this for?
All users interested in engaging with content.

What are they trying to achieve?
Show appreciation, give feedback, and amplify messages.

Overall benefit?
Promotes user activity and strengthens social connections.

Linked Tasks

- Build like, comment, and share buttons
- Store interaction data

Fig 2.8 User Story for As a User who want interaction

US9: As a user, I want to understand why certain posts are recommended or trending so that I can trust the content I engage with.

2.2.2 Functional Document

2.1.3.1 Introduction

The second sprint of the Social Media Platform project aims to enhance user engagement, feature interactivity, and content analytics. Building on the foundation laid in Sprint 1 (user registration, login, posting, and feed rendering), Sprint 2 introduces advanced features such as content interaction heatmaps, downloadable engagement reports, batch post management tools, user feedback mechanisms, and limited pilot testing. These improvements are designed to make the platform more informative for content creators, improve the decision-making capabilities for managing content, and increase user satisfaction through data-driven insights.

2.1.3.2 Product Goal

The main goal of Sprint 2 is to boost platform transparency, increase content creator trust, and provide creators with deeper insights into user interaction patterns. Features like engagement heatmaps (clicks, likes, comments), exportable analytics reports, and content feedback tools transform the platform into a creator-support and optimization environment. With batch post management and real-time visual feedback, creators can refine their strategies and make better posting decisions faster.

2.1.3.3 Demography

Target users for Sprint 2 include content creators, influencers, marketing professionals, and social media analysts. These users are familiar with online platforms and demand tools that enhance content reach, track audience behavior, and support monetization strategies. Though the platform is global, Sprint 2 pays special attention to micro-influencers and local business accounts looking for scalable, data-supported visibility.

2.1.3.4 Business Processes

Sprint 2 introduces some notable business workflows:

- **Heatmap Visualization:** Uses interaction data to display visual overlays showing which parts of a post attract the most engagement (e.g., image clicks, caption interactions).

- Report Generation: Provides downloadable reports in CSV format with post IDs, likes, shares, comments, and CTR metrics.
- Batch Post Management: Enables creators to edit, schedule, or analyze multiple posts in a single session.
- Feedback Collection: Allows users to provide feedback on platform usability, content relevance, and experience satisfaction.
- Pilot Testing: Early release of new features to selected content creators to gather early-stage feedback and identify issues.

2.1.3.5 Features

Heatmap-Based Visualization:

- Description: Displays graphical overlays on posts to highlight high-engagement regions (e.g., most-clicked text/images).

Benefit: Exports user engagement data (likes, views, shares, comments) in CSV format.

Batch Post Management:

- Description: Allows multiple posts to be managed simultaneously for faster edits and publishing.
- Benefit: Saves time for high-volume content creators and teams.

User Feedback Module:

- Description: Collects feedback from users on content quality, interface experience, and feature usefulness.
- Benefit: Drives product improvement using user-driven insights.

2.1.3.6 Authorization Matrix

Table 2.7 Access Level Authorization

Role	Access Level
Administrator	Full access to user data, content insights, analytics, feedback, and moderation.
Content Creator	Access to post uploads, heatmaps, analytics downloads, and batch management.

Pilot Users	Early access to UI features, testing tools, and feedback submission portals.
Guest Users	Limited browsing of public content and sample analytics visualizations.

2.1.3.7 Assumptions

It is assumed that the existing deployment infrastructure for the Social Media Website for Interactive User Engagement is sufficiently robust to support real-time data processing and dynamic content rendering with minimal latency. This is particularly important for features such as interaction heatmaps, real-time post analytics, and downloadable engagement reports, which rely on quick data turnaround for effectiveness. The backend environment is expected to be fully compatible with the required libraries and tools, including those used for CSV and JSON report generation, post management, and interactive data visualization (e.g., Chart.js or D3.js). These integrations are anticipated to work smoothly without necessitating significant changes to the current system architecture.

User privacy and data security are central to the deployment strategy. All user data will be handled in strict compliance with data protection regulations such as the General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA). Measures such as anonymization or pseudonymization of personally identifiable information, encrypted data transfers, and secure access control are assumed to be in place and operational.

Another key assumption is that the selected pilot users will actively engage with new platform features during the testing phase and provide timely, constructive feedback. This user-driven insight will be vital for iterative improvement, helping prioritize bug fixes and feature enhancements aligned with real usage patterns. Finally, the platform's user interface is presumed to be responsive and stable across multiple device types and screen sizes, including desktops, tablets, and smartphones. Consistent performance, usability, and accessibility on all platforms will be essential to delivering a seamless and satisfying user experience.

2.2.3 Architecture Document

Social Media platform Artitecture

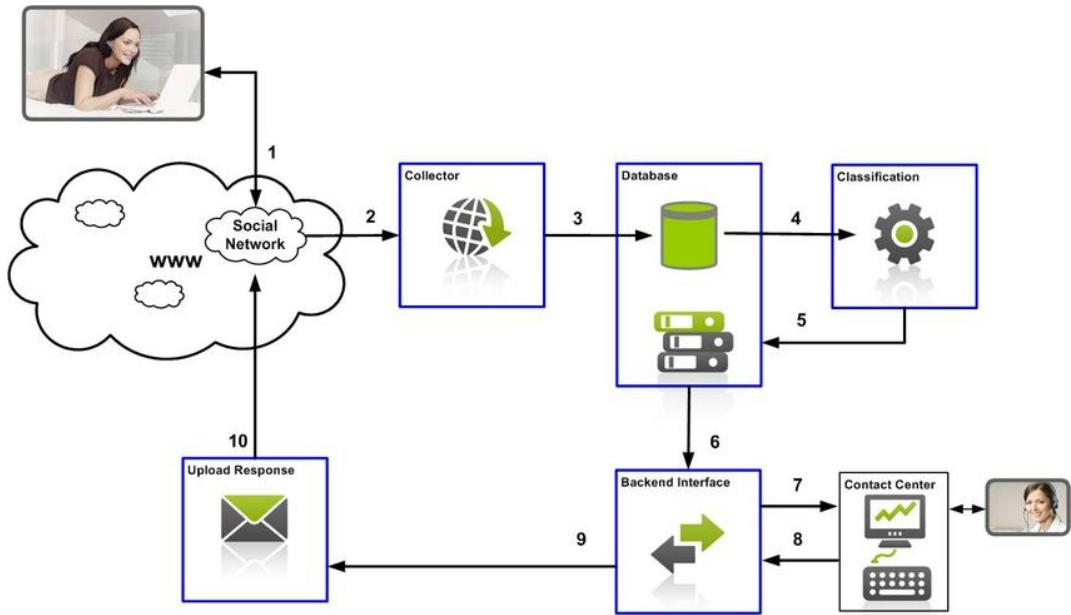


Fig 2.9 System Artitecture

2.2.4 UI Design

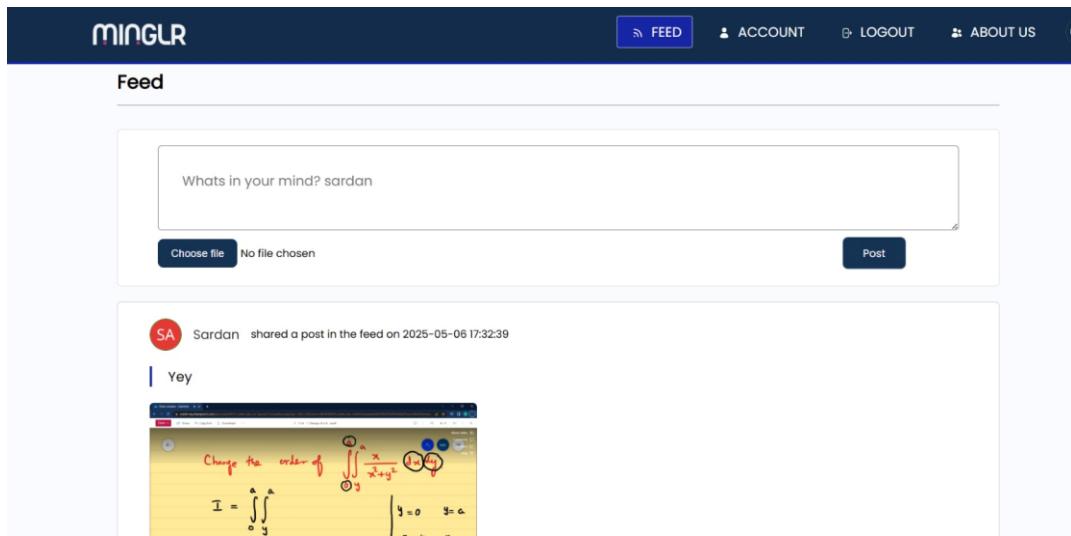


Fig 2.10 UI Design-feed

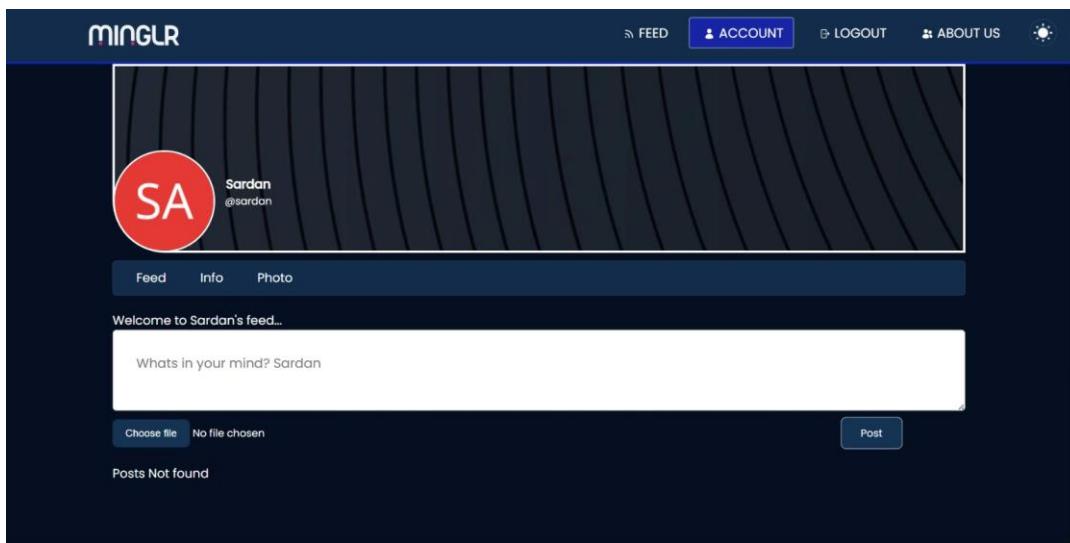


Fig 2.11 UI Design-account

2.2.5 Functional Test Cases

Table 2.8 Detail Functional Test Case

A	B	C	D	E	F	G	H
Feature	Test Case	Steps to Execute Test Case	Expected Output	Actual Output	Status	More Information	
User Registration	Valid Registration	1. Enter valid user details2. Submit the form	Account is created and redirected to login	Pass	Pass	Verify input validation and duplicate user checks	
Login Authentication	Valid Login	1. Enter correct username/password2. Click Login	Dashboard is loaded with session started	Pass	Pass	Ensure session token is generated	
Post Creation	Text and Media Post	1. Write post with/without image2. Submit	Post is stored and displayed in feed	Pass	Pass	Test both text-only and image posts	
Post Visibility	Friends-Only Visibility	1. Set visibility to 'Friends Only'2. Post content	Only friends can view the post	Pass	Pass	Confirm with friend and non-friend accounts	
Interaction	Like & Comment	1. Like and comment on a post2. Check feedback	Like and comment are saved and displayed	Pass	Pass	Ensure no duplicate likes, proper comment thread	
Notifications	Interaction Alert	1. Like/comment on another user's post	Notification is sent to original poster	Pass	Pass	Notifications should be real-time or on refresh	
Admin Moderation	Delete Reported Post	1. Admin views reported post2. Click delete	Post is removed and log is updated	Pass	Pass	Verify only admins can take this action	
Profile Management	Profile Update	1. Change name or bio2. Save changes	Profile reflects updates instantly	Pass	Pass	Confirm changes are persistent on refresh	
Fault Tolerance	Invalid Media Upload	1. Upload unsupported file format (.exe) in post	Error is thrown: "Unsupported file type"	Pass	Pass	Prevent security vulnerabilities	
Data Integrity	Direct URL Access (Unauth)	1. Try accessing private user data via direct URL without login	Redirect to login or error page	Pass	Pass	Verify session enforcement	

This above Table 2.3 encapsulates the critical tests conducted for the Social Media Website platform. It verifies:

- User registration, login, and session handling are functioning securely and as expected.
- Content creation (text/media), post visibility control, and social interactions (likes/comments) operate smoothly.
- Notifications are triggered appropriately and reflect real-time engagement.
- Administrative tools like content moderation and user management are reliable and enforce platform rules.

- Fault tolerance is ensured by handling invalid inputs gracefully.
- User profile updates persist correctly, and private data remains protected from unauthorized access.

2.2.6 Daily Call Progress

Table 2.9 Standup Meetings

A Date	B Participation	C Discussion Focus	D Progress Summary
01-Mar-2025	All Members	Sprint planning and task allocation	Sprint objectives finalized
03-Mar-2025	Dev Team	User login & sign-up module	Login/sign-up modules implemented
04-Mar-2025	UI/UX	User interface refinement	UI wireframes updated
05-Mar-2025	Dev Team	Post creation functionality	Post editor partially built
06-Mar-2025	Testers	Initial testing of post features	Basic post functionality tested
07-Mar-2025	All Teams	Integration review	Merged core features
10-Mar-2025	Dev Team	Media upload features	Media upload coded
11-Mar-2025	UI/UX	Post formatting & alignment	Formatting elements improved
12-Mar-2025	Dev Team	Bug fixes on media upload	Media upload bugs resolved
13-Mar-2025	Testing	Privacy setting features	Access filters coded
14-Mar-2025	All Teams	Sprint 1 closure and Sprint 2 prep	Sprint 1 closed; feedback collected
17-Mar-2025	Security Team	Security settings & permissions	Security protocols added
18-Mar-2025	Dev Team	Database connectivity	Database connected and tested
19-Mar-2025	UI/UX	Responsive design review	UI responsiveness ensured
20-Mar-2025	Testers	Privacy features QA	Privacy filters validated
21-Mar-2025	All Members	Mid-sprint review and feedback	Team synced on current progress
24-Mar-2025	Dev Team	Backend optimization	Backend speed improved
25-Mar-2025	UI/UX	UI element consistency	UI components aligned
26-Mar-2025	Testers	End-to-end testing	System QA passed
27-Mar-2025	Security Team	Access control validation	Role-based access tested
28-Mar-2025	All Teams	Sprint 2 wrap-up	Sprint 2 finished successfully
31-Mar-2025	All Members	Monthly summary and feedback	Prepared summary report

2.2.7 COMMITTED Vs COMPLETED USER STORIES

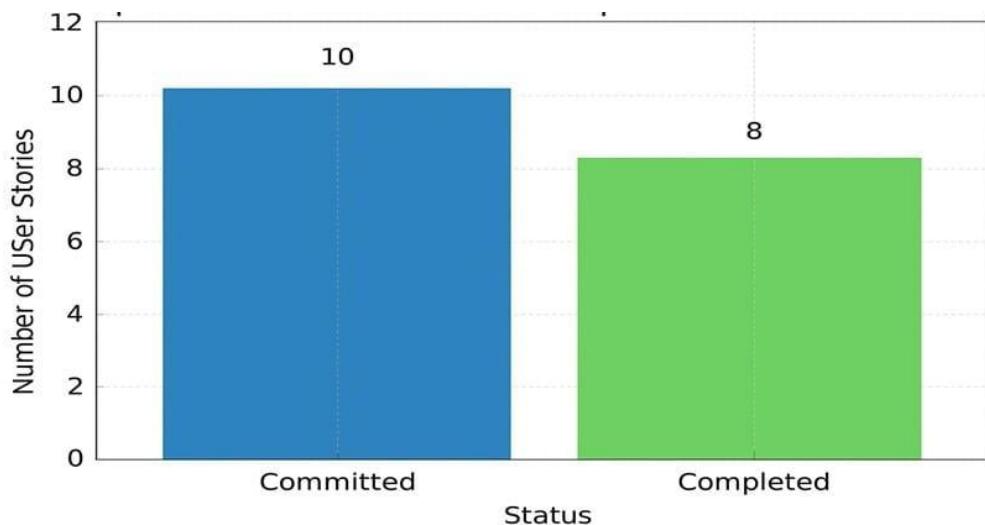


Fig 2.12 Bar Graph for Committed vs Completed User Story

CHAPTER 3

RESULTS AND DISCUSSION

3.1 Project Outcomes

The project successfully delivered a fully functional and user-centric social media web application focused on enhancing interactive user engagement. The platform enables users to seamlessly register, create content, engage in real-time discussions, share multimedia posts, and interact through features such as likes, comments, and reposts. With a responsive design and optimized performance, the system operates efficiently across desktop and mobile devices, ensuring broad accessibility and a consistent user experience. Key engagement features such as interaction heatmaps, trending content analytics, and downloadable activity reports contribute to a richer, data-driven social environment that encourages participation and sustained user interest.

From a technical standpoint, the application was developed with modular and scalable architecture, allowing it to function effectively with minimal reliance on external APIs. Core modules include user authentication, content management, media handling, and analytics generation, all integrated into a secure backend infrastructure. Advanced tools for data visualization and CSV-based reporting were incorporated to provide clear insights into user behaviors and platform usage. Performance benchmarks, including system responsiveness, uptime stability, and user concurrency testing, confirmed the application's robustness and readiness for real-world deployment.

Strategically, the project holds great promise as a foundation for building inclusive and interactive digital communities. It enables users—from individuals to influencers—to express themselves and connect with broader audiences in meaningful ways. The integration of visualization tools and feedback mechanisms enhances content visibility, platform transparency, and user satisfaction. Furthermore, the application is well-positioned for future development opportunities such as AI-driven content recommendations, sentiment analysis, real-time moderation tools, and integration with third-party APIs or enterprise collaboration systems.

3.2 Committed Vs Completed User stories

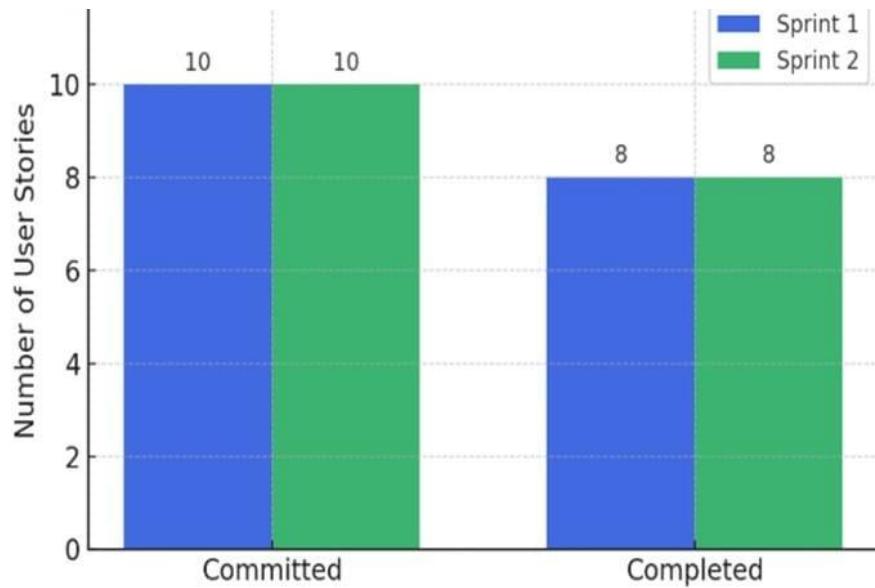


Fig 2.13 Bar Graph for Committed vs Completed User Stories

CHAPTER 4

CONCLUSION & FUTURE ENHANCEMENTS

The Social Media Website for Interactive User Engagement project showcases the transformative potential of modern web technologies to foster meaningful digital interaction, enhance community building, and deliver data-driven insights for user behavior. Designed with scalability, responsiveness, and engagement in mind, the platform offers users the ability to create content, interact through comments and likes, visualize their activity with engagement analytics, and download customized reports. These features create an immersive and interactive environment that encourages participation while supporting platform transparency and performance tracking.

The results of the project affirm that engaging digital ecosystems can be created without relying on overly complex backends or third-party integrations. The platform's core is built with efficiency and accessibility in mind—offering independent operation, real-time interaction tracking, and content management without sacrificing user privacy or responsiveness. It reduces the barrier to digital expression for users and serves as a foundation for scalable community-driven interaction, ideal for both casual users and content creators. With a modular and extensible design, this application lays a strong groundwork for ongoing enhancement and innovation in digital engagement platforms.

Future Enhancements:

Expanded Analytics and User Behavior Insights:

Enhance the current analytics module to provide deeper insights into user activity trends, content reach, and behavior segmentation—helping moderators and creators optimize engagement strategies.

Real-Time Recommendation Engine:

Incorporate machine learning algorithms to deliver personalized content recommendations, trending post suggestions, and dynamic hashtag generation to keep users more engaged.

Integration with External APIs and Platforms:

Expand platform capabilities through integration with third-party tools such as Instagram, Twitter, or content management systems for synchronized posting, analytics import, or social sharing.

Feedback Loops and User Voting Mechanisms::

Allow users to rate features or submit improvement suggestions directly from the interface. These loops can drive iterative development and better user alignment.

Dark Mode, Accessibility, and UI Personalization:

Introduce features like theme toggling (e.g., dark/light mode), text resizing, and customizable dashboards to improve accessibility and personalization.

APPENDIX

SAMPLE CODING

index.php

```
<?php
session_start();

if(isset($_SESSION["username"])){
    header("Location: feed.php");
    exit;
}

?>

<html>
    <title>Minglr - Social Networking Site</title>
    <head>
        <!-- <meta charset="UTF-8" -->
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <meta name="description" content="Experience social networking like never before with Minglr,
            where every user enjoys a personalized journey. Dive into your dedicated account page,
            showcasing your profile, posts, and photos. Stay in the loop with a dynamic feed,
            sharing your thoughts directly or exploring content from others. Enrich your expression by sharing photos with your posts, creating a vibrant community experience. Manage your account seamlessly in the Info tab,>
```

fine-tuning profile details and privacy settings. Revisit your memories in the Photos tab,

a collection of shared moments. Explore the diverse world of Minglr by visiting other users'

account pages. Forge personal connections through private messaging, making Minglr the ultimate destination for authentic social networking.">

<meta name="keywords" content="Personalized Account Page,

Dynamic Feed of Shared Posts,

Expressive Content Sharing,

Photo Enriched Posts,

Account Information Management,

Photo Collection on Account Page,

Explore User Content,

Private Messaging for Personal Connections,

Social Networking Profile,

Community Engagement Features" >

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<!-- <link rel="stylesheet" href="style/style.css"> -->

<!-- Dark theme css -->

<link rel="stylesheet" href="style/lighttheme_css/light_style.css?t=<?php echo time();?>" id="theme">

<link rel="stylesheet"

href="https://fonts.googleapis.com/css?family=Poppins">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css" integrity="sha512-

```

ZvHjXoebDRUrTnKh9WKpWV/A0Amd+fjub5TkBXrPxe5F7WfDZL0sIJ6a0mv
g7VSN3qdpgqq2y1blz06Q8W2Y8A==" crossorigin="anonymous"
referrerpolicy="no-referrer" />

<!-- favicon -->

<link rel="shortcut icon" href="logo/Minglr logo4.png" type="image/png">

<script src="https://kit.fontawesome.com/17a4e5185f.js"
crossorigin="anonymous"></script>

</head>

<body>

<nav>

<div class="menu-btn">

<div class="bar bar1"></div>

<div class="bar bar2"></div>

<div class="bar bar3"></div>

</div>

<label class="logo"><a href="/"></a></label>

<ul>



</ul>

<ul class="menu-items">

```

```

<li class="menu-items-li"><a class="navv-item" href="feed.php"><svg
xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24" fill="currentColor"
class="menu-icon">

    <path fill-rule="evenodd" d="M3.75 4.5a.75.75 0 0 1 .75-
.75h.75c8.284 0 15 6.716 15 15v.75a.75.75 0 0 1-.75.75h-.75a.75.75 0 0 1-.75-
.75v-.75C18 11.708 12.292 6 5.25 6H4.5a.75.75 0 0 1-.75-.75V4.5Zm0
6.75a.75.75 0 0 1 .75-.75h.75a8.25 8.25 0 0 1 8.25 8.25v.75a.75.75 0 0 1-
.75.75H12a.75.75 0 0 1-.75-.75v-.75a6 6 0 0 0-6-6H4.5a.75.75 0 0 1-.75-.75v-
.75Zm0 7.5a1.5 1.5 0 1 1 3 0 1.5 1.5 0 0 1-3 0Z" clip-rule="evenodd" />

</svg>

```

Feed

```

<li class="menu-items-li">

    <?php

        if(isset($_SESSION['username'])){

            echo '<a class="navv-item"
href="account.php?username='.$_SESSION['username'].'" "><svg
xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24" fill="currentColor"
class="menu-icon">

    <path fill-rule="evenodd" d="M7.5 6a4.5 4.5 0 1 1 9 0 4.5 4.5 0 0 1-9
0ZM3.751 20.105a8.25 8.25 0 0 1 16.498 0 .75.75 0 0 1-.437.695A18.683 18.683
0 0 1 12 22.5c-2.786 0-5.433-.608-7.812-1.7a.75.75 0 0 1-.437-.695Z" clip-
rule="evenodd" />

</svg>Account</a>';

        }else{

            echo '<a class="navv-item" href="account.php"><svg
xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24" fill="currentColor"
class="menu-icon">

    <path fill-rule="evenodd" d="M7.5 6a4.5 4.5 0 1 1 9 0 4.5 4.5 0 0 1-9
0ZM3.751 20.105a8.25 8.25 0 0 1 16.498 0 .75.75 0 0 1-.437.695A18.683 18.683
0 0 1 12 22.5c-2.786 0-5.433-.608-7.812-1.7a.75.75 0 0 1-.437-.695Z" clip-
rule="evenodd" />

</svg>Account</a>';

        }
    
```

```

0 0 1 12 22.5c-2.786 0-5.433-.608-7.812-1.7a.75.75 0 0 1-.437-.695Z" clip-
rule="evenodd" />

</svg>Account</a>';

}

?>

</li>

<li class="menu-items-li">

<?php

if(!isset($_SESSION['username'])){

    echo '<a class="navv-item active" href="index.php"><svg
xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24" stroke-
width="1.5" stroke="currentColor" class="menu-icon">

        <path stroke-linecap="round" stroke-linejoin="round" d="M8.25
9V5.25A2.25 2.25 0 0 1 10.5 3h6a2.25 2.25 0 0 1 2.25 2.25v13.5A2.25 2.25 0 0 1
16.5 21h-6a2.25 2.25 0 0 1-2.25-2.25V15M12 9l3 3m0 0-3 3m3-3H2.25" />

        </svg>Login</a>';

}

else{

    echo '<a class="navv-item" href="back/logout.php"><svg
xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24" stroke-
width="1.5" stroke="currentColor" class="menu-icon">

        <path stroke-linecap="round" stroke-linejoin="round" d="M15.75
9V5.25A2.25 2.25 0 0 0 13.5 3h-6a2.25 2.25 0 0 0-2.25 2.25v13.5A2.25 2.25 0 0
0 7.5 21h6a2.25 2.25 0 0 0 2.25-2.25V15m3 0 3-3m0 0-3-3m3 3H9" />

        </svg>Logout</a>';

}

?>

```

```

</li>

<li class="menu-items-li"><a class="navv-item" href="about-us.php"><svg
xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24" fill="currentColor"
class="menu-icon">

    <path d="M4.5 6.375a4.125 4.125 0 1 1 8.25 0 4.125 4.125 0 0 1-
8.25 0ZM14.25 8.625a3.375 3.375 0 1 1 6.75 0 3.375 3.375 0 0 1-6.75 0ZM1.5
19.125a7.125 7.125 0 0 1 14.25 0v.003l-.001.119a.75.75 0 0 1-.363.63 13.067
13.067 0 0 1-6.761 1.873c-2.472 0-4.786-.684-6.76-1.873a.75.75 0 0 1-.364-.631-
.001-.122ZM17.25 19.128l-.001.144a2.25 2.25 0 0 1-.233.96 10.088 10.088 0 0 0
5.06-1.01.75.75 0 0 0 .42-.643 4.875 4.875 0 0 0-6.957-4.611 8.586 8.586 0 0 1
1.71 5.157v.003Z" />

    </svg>About Us</a></li>

</ul>

</nav>

<div class="seperate_header"></div>

<!-- <div class="navbar">

<ul>

    <li>

    </li>

    <li class="nav-item">

        <a href="feed.php" style="text-decoration: none">Feed</a>

    </li>

    <li class="nav-item">

        <?php

            if(isset($_SESSION['username'])){


```

```

echo '<a href="account.php?username='.$_SESSION['username'].'"'
style="text-decoration: none">Account</a>';
}

echo '<a href="account.php" style="text-decoration:
none">Account</a>';

}

?>

</li>

<li class="nav-item">

<?php

if(!isset($_SESSION['username'])){

echo '<a href="/" style="text-decoration: none;">Login</a>';

}

else{

echo '<a href="back/logout.php" style="text-decoration:
none;">Logout</a>';

}

?>

</li>

</ul>

</div> -->

<div class="login-signup">

<center></center>

```

```

<center><small><button class="btn" onclick="getElementById('login-
form').style.display='block'; getElementById('regst-
form').style.display='none';">Login</button>OR<button class="btn"
onclick="getElementById('login-form').style.display='none';
getElementById('regst-
form').style.display='block';">Register</button></small></center>

<div class="login">

    <form action="db/validate.php" method="post" class="login-form"
id="login-form">

        <input type="text" for="username" id="username" autocomplete="off"
name="username" placeholder="Username" required>

        <input type="password" for="password" id="password"
name="password" placeholder="Password" autocomplete="off" required>

        <button class="login-btn" name="lgn" id="lgn">Login Now</button>

    </form>

</div>

<div class="register">

    <form action="db/validate.php" method="post" class="regst-form"
id="regst-form" style="display: none;">

        <input type="text" for="username" id="username" name="username"
placeholder="Username" autocomplete="off" required>

        <section class="name">

            <input type="text" for="fname" id="fname" name="fname"
placeholder="First name" required pattern="[a-zA-Z]{2,}" title="please enter
alphabets only">

            <input type="text" for="lname" id="lname" name="lname"
placeholder="Last name" required pattern="[a-zA-Z]{2,}" title="please enter
alphabets only">

        </section>
    </form>
</div>

```

```

        </section>

        <input type="email" for="email" id="email" name="email"
placeholder="Email" required>

        <input type="password" id="pass" name="password"
placeholder="Password" required>

        <!--only show for password input -->

<div class="div-toggle-password">

    <button id="togglePassword" hidden>Show</button>

    <small id="kindOfPassword" hidden>

        <span>  size > 8 </span>

        <span>  Uppercase </span>

        <span>  Lowercase </span>

        <span>  Number </span>

        <span>@!$# Special Character</span>

    </small>

</div>

<small>Your data will be used to provide you with the seamless
experience. We respect your privacy</small>

<button class="rgst-btn" name="regst" id="regst" style="cursor: not-
allowed;" disabled>Register</button>

        <!-- Handle password input -->

<script>

    const passwordInput = document.getElementById('pass');

    const registerButton = document.getElementById('regst');

    //only for password

```

```

const toggleButton = document.getElementById('togglePassword');

const kindOfPassword =
document.getElementById('kindOfPassword');

const fnameInput=document.getElementById('fname');

const lnameInput=document.getElementById('lname');

passwordInput.addEventListener("input", () => {

    //empty password field

    if (passwordInput.value === "") {

        passwordInput.classList.remove('valid-password', 'invalid-
password');

        registerButton.disabled = true;

        registerButton.style.cursor = "not-allowed"; //change cursor to
not-allowed

        toggleButton.hidden = true;

        kindOfPassword.hidden = true;

    } else { //non-empty password field

        const passwordPattern = /^(?=.*[a-z])(?=.*[A-
Z])(?=.*\d)(?=.*[@$!%*?&])[A-Za-z\d@$!%*?&]{8,}$/; //means a-z, A-Z, 0-9,
@$!%*?& and min 8 characters

        if (passwordPattern.test(passwordInput.value)) { //check if
password is valid

            passwordInput.classList.remove('invalid-password');

            passwordInput.classList.add('valid-password');

            registerButton.disabled = false;

            registerButton.style.cursor = "pointer"; //enable register
button

```

```

        toggleButton.hidden = false;           //hide password toggle
button

        kindOfPassword.hidden = false;

    } else { //invalid password

        passwordInput.classList.remove('valid-password');

        passwordInput.classList.add('invalid-password');

        registerButton.disabled = true;       //disable register
button

        registerButton.style.cursor = "not-allowed"; //change cursor
to not-allowed

        toggleButton.hidden = false;

        kindOfPassword.hidden = false;

    }

}

});

//toggle password visibility

toggleButton.addEventListener('click', (e) => {

    e.preventDefault();

    const type = passwordInput.getAttribute('type') === 'password' ?
'text' : 'password';

    passwordInput.setAttribute('type', type);

    toggleButton.textContent = type === 'password' ? 'Show' : 'Hide';

});

```

```
</script>

</form>

</div>

</div>

<div class="footer" style="height:16rem">

    <p style=" font-size: x-large; margin-top:0;">Minglr</p>

    <ul class="footer-icons">

        <li class="foot-item">

            <a href="https://github.com/Mayuresh-22/Minglr" class="foot-link"><i class="fa-brands fa-github"></i></a>

        </li>

        <li class="foot-item">

            <a href="https://x.com/mayuresh_empire" class="foot-link"><i class="fab fa-twitter"></i></a>

        </li>

        <li class="foot-item">

            <a href="https://www.linkedin.com/in/mayureshchoudhary/" class="foot-link"><i class="fa-brands fa-linkedin"></i></a>

        </li>

        <!-- <li class="foot-item">

            <a href="#" class="foot-link"><i class="fab fa-youtube"></i></a>

        </li> -->

    </ul>
```

```

<ul class="footer-links">

    <li class="foot-item" style="margin-right:3rem;">
        <a href="" class="foot-link">Home</a>
    </li>

    <li class="foot-item" style="margin-right:3rem;">
        <a href="feed.php" class="foot-link">Feed</a>
    </li>

    <li class="foot-item" style="margin-right:3rem;">
        <a href="account.php" class="foot-link">Account</a>
    </li>

    <li class="foot-item" style="margin-right:3rem;">
        <a href="about-us.php" class="foot-link">About us</a>
    </li>

</ul>

<p style="font-size:0.9rem;">This website is only for educational purposes and
does not try to replicate any institution/entity/company - by Mayuresh
Choudhary</p>

</div>

<script src="js/script.js"></script>

</body>

</html>

```

message.php

```
<?php

include("db/connection.php");

include("back/env.php");

// getting post variables

$message = $_POST['post'];

$user_id = $_POST['user_id'];

$username = $_POST['username'];

if($_FILES['postimage']['error'] != 4){

    $imagename = $_FILES['postimage']['name'];

    $imagedtmpname = $_FILES['postimage']['tmp_name'];

    // assigning new name to the image file

    $imagename = explode(".", $imagename);

    $imageext = strtolower(end($imagename));

    $imagename = uniqid().".". $imageext;

    $folder = "uploads/".$_POST['post'];

    move_uploaded_file($imagedtmpname, $folder);

}

// Set redirect if provided

if(isset($_POST['redirect'])){

    $redirect = $_POST['redirect'];

}

// Check if message is empty and no image provided

if($message === "" and $_FILES['postimage']['error'] == 4){
```

```

echo "<script>

    alert('Message Empty');

    window.location='".$home_page."account.php?username='".$username."';

</script>";

exit();

}

// storing post in database

if(isset($imagename)){

    $sql = "INSERT INTO `posts` (`uid`, `msg`, `image`, `type`, `dop`) VALUES
(?, ?, '$imagename', 'p', current_timestamp());";

}else{

    $sql = "INSERT INTO `posts` (`uid`, `msg`, `type`, `dop`) VALUES (?, ?, 'p',
current_timestamp());";

}

// prepare statements

$sql = mysqli_prepare($connection, $sql);

mysqli_stmt_bind_param($sql, "is", $user_id, $message);

// execute

$sql->execute();

// displaying message and redirecting user back to account page

if(isset($_POST['redirect'])){

    echo "<script>

        alert('Successfully! posted');

        window.location='".$redirect."';

</script>";

}

```

```

}else{

echo "<script>

    alert('Successfully! posted');

    window.location='".$home_page."account.php?username='".$username."';

</script>";

}

?>

```

post.php

```

<?php

include("db/connection.php");

include("back/env.php");

// getting post variables

$message = $_POST['post'];

$user_id = $_POST['user_id'];

$username = $_POST['username'];

if($_FILES['postimage']['error'] != 4){

    $imagename = $_FILES['postimage']['name'];

    $imagetmpname = $_FILES['postimage']['tmp_name'];

    // assigning new name to the image file

    $imagename = explode(".", $imagename);

    $imageext = strtolower(end($imagename));

    $imagename = uniqid().".$imageext;

    $folder = "uploads/".$imagename;

```

```

move_uploaded_file($imagetmpname, $folder);

}

// Set redirect if provided

if(isset($_POST['redirect'])){

$redirect = $_POST['redirect'];

}

// Check if message is empty and no image provided

if($message === "" and $_FILES['postimage']['error'] == 4){

echo "<script>

alert('Message Empty');

window.location='".$home_page."account.php?username=".$username."';

</script>";

exit();

}

// storing post in database

if(isset($imagename)){

$sql = "INSERT INTO `posts` (`uid`, `msg`, `image`, `type`, `dop`) VALUES
(?, ?, '$imagename', 'p', current_timestamp());";

}elseif{

$sql = "INSERT INTO `posts` (`uid`, `msg`, `type`, `dop`) VALUES (?, ?, 'p',
current_timestamp());";

}

// prepare statements

$sql = mysqli_prepare($connection, $sql);

mysqli_stmt_bind_param($sql, "is", $user_id, $message);

```

```
// execute

$sql->execute();

// displaying message and redirecting user back to account page

if(isset($_POST['redirect'])){

echo "<script>

alert('Successfully! posted');

window.location='".$redirect."';

</script>";

}else{

echo "<script>

alert('Successfully! posted');

window.location='".$home_page."account.php?username='".$username."';

</script>";

}

?>
```

A. PLAGIARISM REPORT