

PROFORMA FOR THE APPROVAL PROJECT PROPOSAL

PRN No. 1: Roll No 1:

PRN No. 2: Roll No 2:

1. Name of the Students:

Mr. Patil Somnath Nandu & Mr. Saroj Vicky Omprakash

2. Title of the Project:

Social Networking Site

3. Name of the Guide

Prof.Mrs.Ranjeeta Kapoor

4. Experience of the Guide

- Prof.Mrs.Ranjeeta Kapoor: having 17 years teaching experience.

Signature of the Students

Signature of the Guide

Date:

Date:

Signature of the Coordinator

Date:

Social Networking Site

A Project Report

Submitted in partial fulfilment of the

Requirements for the award of the Degree of

BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)

By

Name of The Students:

Mr. Patil Somnath Nandu

Seat Number:

Mr. Saroj Vicky Omprakash

Seat Number:

Under the esteemed guidance of :

Prof. Mrs. Ranjeeta Kapoor



DEPARTMENT OF INFORMATION TECHNOLOGY

SHRI. P.L. SHROFF COLLEGE OF ARTS & COMMERCE, CHINCHANI

(Affiliated to University of Mumbai)

CHINCHANI, 401503

MAHARASHTRA

2025-2026

SHRI. P.L. SHROFF COLLEGE OF ARTS & COMMERCE, CHINCHANI

(Affiliated to University of Mumbai)

CHINCHANI-MAHARASHTRA-401503

DEPARTMENT OF INFORMATION TECHNOLOGY



CERTIFICATE

This is to certify that the project entitled, "**Social Networking Site**", is bonafied work of **Patil Somnath Nandu** bearing Seat No.: (_____) and **Saroj Vicky Omprakash** bearing Seat No.: (_____) submitted in partial fulfilment of the requirements for the award of degree of **BACHELOR OF SCIENCE** in **INFORMATIONTECHNOLOGY** from University of Mumbai.

Internal Guide

Coordinator

Internal Examiner

External Examiner

Principal

College Seal

Date:

Abstract

Title: Social Networking Site

This project is a social media platform built using the **MERN stack** (MongoDB, Express.js, React, Node.js). It allows users to create profiles, share photos, videos, and stories, and connect with friends through likes, comments, and chat. To make the platform safer, it has a **Fake News Detection API**. The aim is to give users a fun, secure, and reliable social networking experience.

Main Features

1. User Accounts – Signup/login and create profile with bio and upload photos.
2. Post & Stories – Share images, videos, and short stories with captions.
3. Likes & Comments – Interact with posts by liking, commenting, and sharing.
4. Private Chat – Direct messaging between users.
5. Fake News Detection – API checks doubtful content.
6. Personalized Feed – See posts from friends, interests, and trending topics.
7. Privacy & Security – Encrypted messages and privacy settings for safety.

ACKNOWLEDGEMENT

We would like to express our heartfelt gratitude to all the individuals and organizations who contributed to the successful completion of the "*Social Networking Site*" project.

First and foremost, we extend our sincere appreciation to our faculty members for their dedication, expertise, and tireless efforts throughout the project's development. Your collaborative spirit and commitment were instrumental in bringing this innovative system to success.

We are deeply thankful to OpenAI and Google for providing access to the ChatGPT, which played a crucial role in building this project. Your help in AI research and development has been very valuable.

Finally, we acknowledge the support and encouragement of our friends and family, who stood by us throughout this journey.

This project would not have been possible without the efforts and support of all these individuals and organizations. We are grateful for the opportunity to work on this innovative project and are excited about the potential impact it may have on enhancing information retrieval and document generation for users.

Thank you all for your contributions and support.

Sincerely,

Patil Somnath Nandu

Saroj Vicky Omprakash

DECLARATION

We hereby declare that the project entitled, "*Social Networking Site*", done **at place where the project is done**, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge otherthan me, no one has submitted to any other university.

The project is done in partial fulfilment of the requirements for the award of degree of **BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as final semester project as part of our curriculum.

Name and Signature of the Students

Mr. Patil Somnath Nandu

**Mr. Saroj Vicky
Omprakash**

TABLE OF CONTENTS

Chapter 1: Introduction

1.1 Background	9
1.2 Objectives	9
1.3 Purpose and Scope.....	10
1.3.1 Purpose.....	10
1.3.2 Scope	10
1.4 Achievements.....	11
1.5 Organization of Report.....	11

Chapter 2: Survey of technologies

2.1 Existing System.....	12
2.2 Proposed System.....	13

Chapter 3: Requirement and Analysis

3.1 Problem Definition.....	14
3.2 Requirement Specifications.....	15
3.3 Planning and Scheduling.....	16
3.4 Hardware Requirements & Software Requirements.....	17
3.5 Preliminary Product Description	18
3.6 Conceptual Models	18

Chapter 4: System Design

4.1 Basic Modules.....	22
4.2 Data Design.....	23
4.3 Procedural Design	24
4.4 User interface Design.....	25
4.5 Security Issues.....	26

List of Figures

Figure No.	Figure Name	Page No.
1	Gantt Chart	16
2	E-R Diagram	18
3	Use case Diagram	20
4	Class Diagram	21

Chapter 1

Introduction

1.1 Background

Social media platform allows people to connect and stay entertained. However, fake news spreads quickly and can mislead users. This project, **Social Networking Platform with Fake News Detection**, combines social features with a built-in system to detect misleading content.

The platform is developed using the **MERN stack (MongoDB, Express.js, React, Node.js)**.

1.2 Objectives

1. Build a Modern Social Platform

Develop an application where users can share photos, videos, and stories. Users can also like, comment, and message each other to stay connected.

2. Integrate Fake News Detection

Use a Fake News Detection API to automatically check any misleading or harmful content.

3. Enable Real-Time Interaction

Make sure users get instant updates on feeds, posts, and notifications for a smooth and engaging experience.

4. Ensure Privacy and Security

Keep user data safe with secure login and privacy settings.

1.3 Purpose and Scope

1.3.1 Purpose

- 1. Enhance Social Interaction** – Let users share photos, videos, and stories, and engage through likes, comments, and messages.
- 2. Combat Fake News** – Use a Fake News Detection API to check mislead content.
- 3. Ensure User Safety** – Keep login safe, protect privacy, and manage content to build trust.

1.3.2 Scope

- 1. User Management** – Store profiles, posts, likes, comments, and chat history securely.
- 2. Content Sharing** – Allow users to upload media, write captions, like, comment, and send messages.
- 3. Fake News Detection** – Automatically analyse text content report suspicious posts.
- 4. User-Friendly Interface** – Responsive design using React for smooth use on mobile and desktop.

1.4 Achievements

This project has several key goals:

- 1. Better Social Interaction** – Users can interact and share content in a community-focused environment.
- 2. Fake News Detection** – The system can detect misleading or unreliable content.
- 3. Improved Security** – User data is protected with secure login, privacy settings, and access controls.
- 4. Real-Time Updates** – Feeds, notifications, and interactions update instantly for a smooth experience.
- 5. Building Trust** – Provides a safe and transparent platform for responsible social interaction.

1.5 Report Organization

- 1. User Engagement** – Summary of how users interact on the platform.
- 2. Content Analysis** – Information about detected misleading content and overall trends.
- 3. Admin Tools** – Features for admins to monitor activity and manage flagged content.

Chapter 2

Survey of Technologies

2.1 Existing System

Existing social media platforms such as Instagram, Facebook, and Twitter (X) have revolutionized digital communication by enabling users to instantly share photos, videos, and opinions with a global audience. However, these platforms face a major challenge—the uncontrolled spread of fake news and misinformation.

This lack of automated verification allows false or misleading information to spread rapidly, influencing public opinion, creating confusion, and damaging trust among users.

Additionally, fake news detection on existing platforms is reactive; actions are usually taken after misinformation goes viral. The dependence on human moderators and third-party fact-checkers leads to delayed responses and inconsistent results. Furthermore, issues like data privacy breaches, limited transparency, and weak user control make these platforms less reliable and secure for genuine communication.

Key Problems with Existing Platforms:

1. **No Automated Fake News Check** – Platforms don't have built-in systems to verify if content is true or false.
2. **Slow and Reactive Moderation** – Fake news is detected only after it spreads widely.
3. **Manual Review** – Most checks are done by people, which takes time and may not be accurate.
4. **Data Privacy Issues** – Personal data is not always handled securely.

5. **Limited User Awareness** – Users are not warned about misleading or unverified posts.

2.2 Proposed System

The proposed “Social Networking Platform with Fake News Detection” is designed to overcome these problems by adding an API-based fake news detection system to a modern social media platform.

This system allows users to post, like, comment, and share content but with an extra layer of safety. When users post something, the fake news detection API checks the content and identifies whether it may contain false or misleading information. This helps stop the spread of fake news before it reaches many users.

The platform will be developed using the MERN stack MongoDB, Express.js, React.js, and Node.js which makes it fast, reliable, and suitable for both web and mobile devices.

Key Features:

- 1. API-Based Fake News Detection** – Uses an external API to check and flag suspicious or misleading posts automatically.
- 2. Real-Time Notifications** – Users get instant alerts about likes, comments, and fake news warnings.
- 3. Admin Dashboard and Reports** – Admins can view flagged posts, manage content, and see fake news trends.
- 4. Better Security and Privacy** – User data is protected with secure login and encryption.

Chapter 3

Requirements and Analysis

“This chapter explains the problems of existing platforms and the requirements for the proposed system”.

3.1 Problem Definition

- 1. Spread of Misinformation** – Users can post anything without verification, which allows fake news to spread quickly.
- 2. Reliance on Manual Moderation** – Human moderators or user reports are slow and cannot handle large content efficiently.
- 3. Privacy and Security Concerns** – Weak login and data protection can lead to unauthorized access and lower user trust.
- 4. Lack of Transparency and Trust** – Users don't always know when content is flagged, reducing confidence in the platform.

System Requirements to Solve These Problems

- 1.** Automate misinformation detection using APIs.
- 2.** Provide centralized content management and analytics for admins to monitor and manage content effectively.
- 3.** Ensure user privacy and data safety, creating a trustworthy online community.

3.2 Requirements Specification

1. Functional Requirements

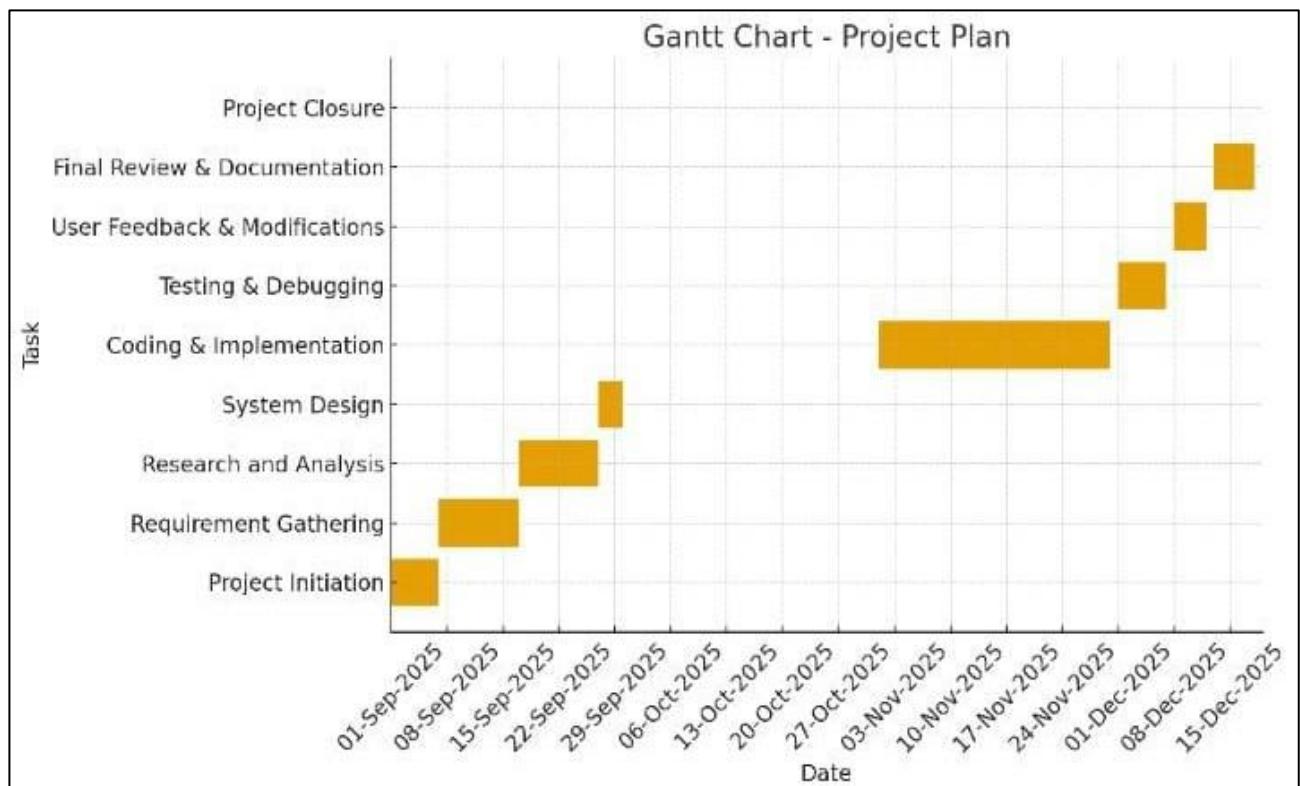
- 1. Content Sharing** – Users can share media and interact through likes, comments, and messages.
- 2. Content Storage** – Store data securely and manage content easily (MongoDB).
- 3. Fake News Detection** – Automatically detect and flag misleading content.
- 4. User Roles & Moderation** – Different roles (User, Moderator, Admin) with tools to manage flagged content.

2. Non-Functional Requirements

- 1. Security & Privacy** – Protect user data and use secure login.
- 2. Compatibility** – Works on all devices and browsers, mobile-friendly.
- 3. Performance** – Fast loading.
- 4. Scalability** – Can handle more users and content without slowing down.

3.3 Planning and Scheduling: GANTT chart:

Task	Start Date	End Date	Duration
Project Initiation	01/09/2025	07/09/2025	7 Days
Requirement Gathering	07/09/2025	17/09/2025	10 Days
Research and Analysis	17/09/2025	27/09/2025	10 Days
System Design	27/09/2025	31/09/2025	5 Days
Coding & Implementation	01/11/2025	30/11/2025	29 Days
Testing & Debugging	01/12/2025	07/12/2025	7 Days
User Feedback & Modifications	08/12/2025	12/12/2025	5 Days
Final Review & Documentation	13/12/2025	18/12/2025	6 Days
Project Closure	19/12/2025	21/12/2025	3 Days



3.4 Hardware Requirements and Software Requirements (Development Environment)

➤ Hardware Requirements

- 1. Laptop:** Acer aspire lite and Hp Victus used for development.
- 2. Processor:** AMD Ryzen 5 5625U and Intel i5 12450H for smooth coding and testing.
- 3. RAM:** 16GB for faster performance.
- 4. Storage:** 512GB SSD to store project files and software.
- 5. GPU:** AMD Vega 7 (integrated) and NVIDIA RTX 3050 (dedicated) for better performance, especially for heavy image/video tasks.

➤ Software Requirements

- 1. IDE:**
 - Visual Studio Code (VS Code) – Main IDE for writing, debugging, and testing frontend and backend code.
- 2. Languages & Libraries:**
 - Frontend: React.js with (HTML5, CSS3 & JS(E7)).
 - Backend: Node.js, Express.js.
 - Database: MongoDB.
- 3. Web API Integration:**
 - Fake News Detection API to check and flag misleading content in real-time.
- 4. Version Control:**
 - Git for source code management.

- GitHub for repository hosting and collaboration.

5. Operating System:

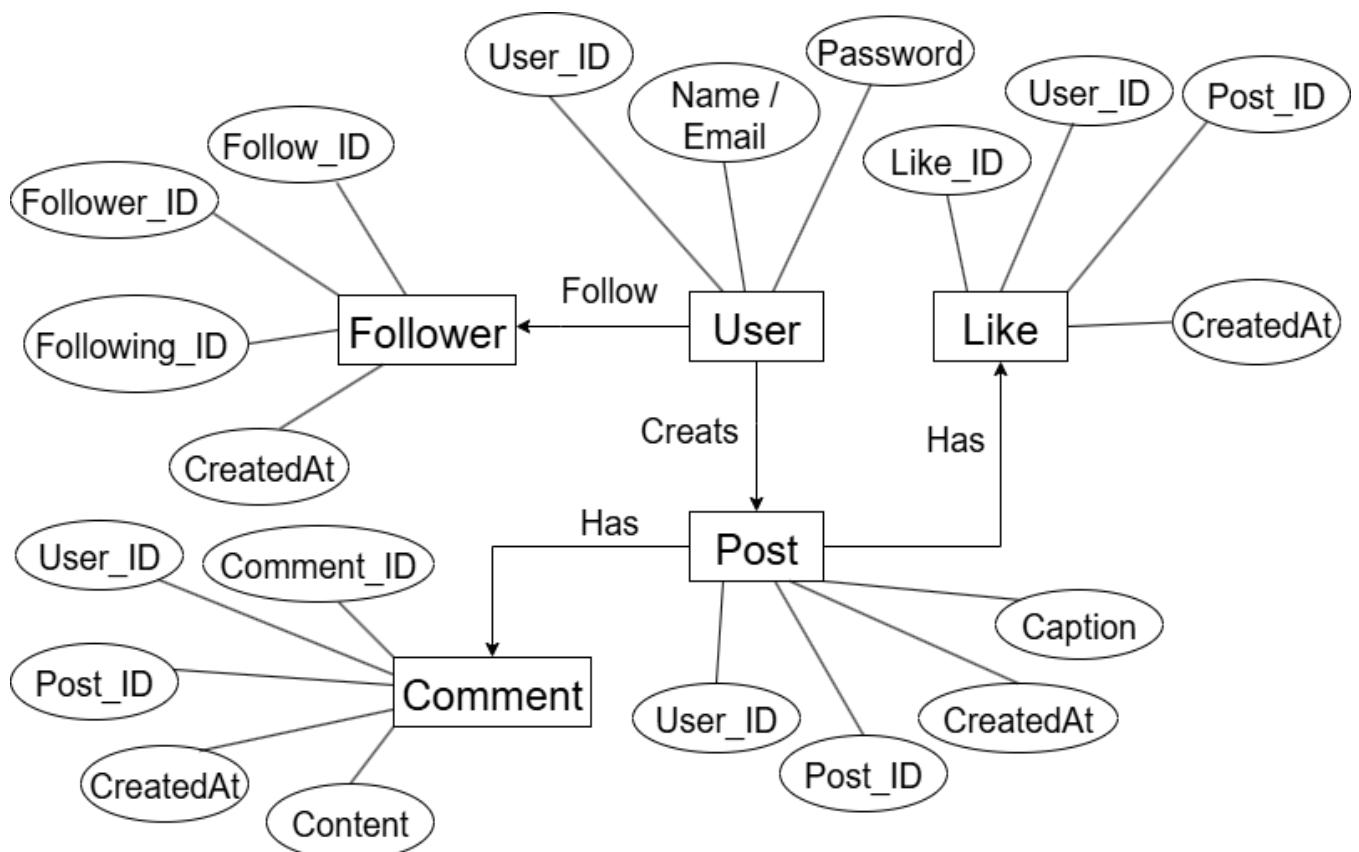
- Development done on Windows 11.
- The system is also compatible with macOS and Linux.

3.5 Preliminary Product Description

The Social Networking Platform with Fake News Detection is a web-based application that allows users to share content, interact, and stay informed safely. It provides secure login, real-time detection of misleading content, notifications, and a mobile-friendly interface. The system ensures accurate, safe, and smooth interaction while being easy to use on different devices.

3.6 Conceptual models:

Entity-Relationship Diagram:



Entity Attributes:

1. User

- User_ID (PK)
- Name / Email
- Password

2. Post

- Post_ID
- User_ID (FK)
- Caption
- CreatedAt

3. Comment

- Comment_ID
- User_ID (FK)
- Post_ID (FK)
- Content
- CreatedAt

4. Like

- Like_ID
- User_ID (FK)
- Post_ID (FK)
- CreatedAt

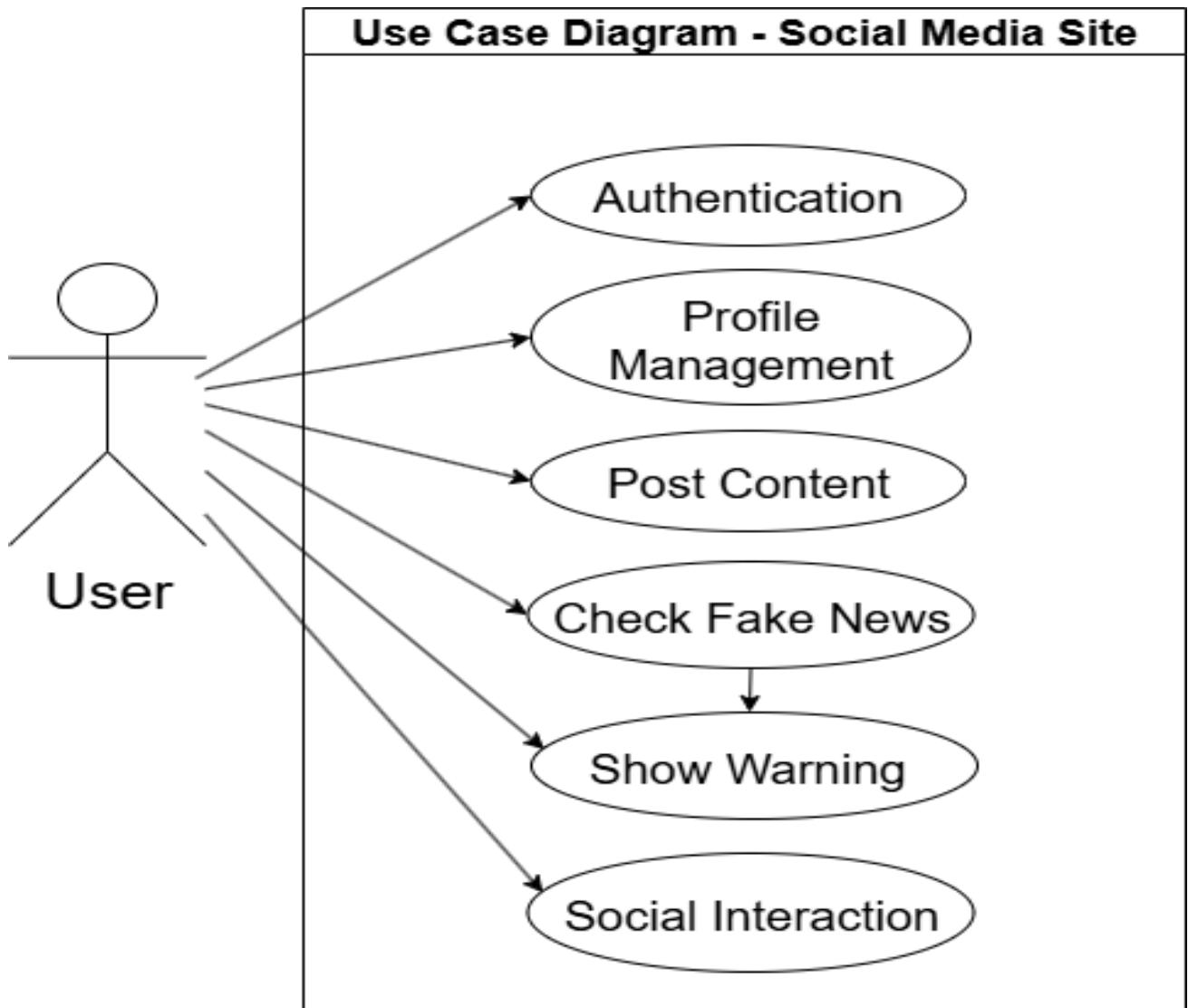
5. Follower

- Follow_ID
- Follower_ID (FK → User)
- Following_ID (FK → User)
- CreatedAt

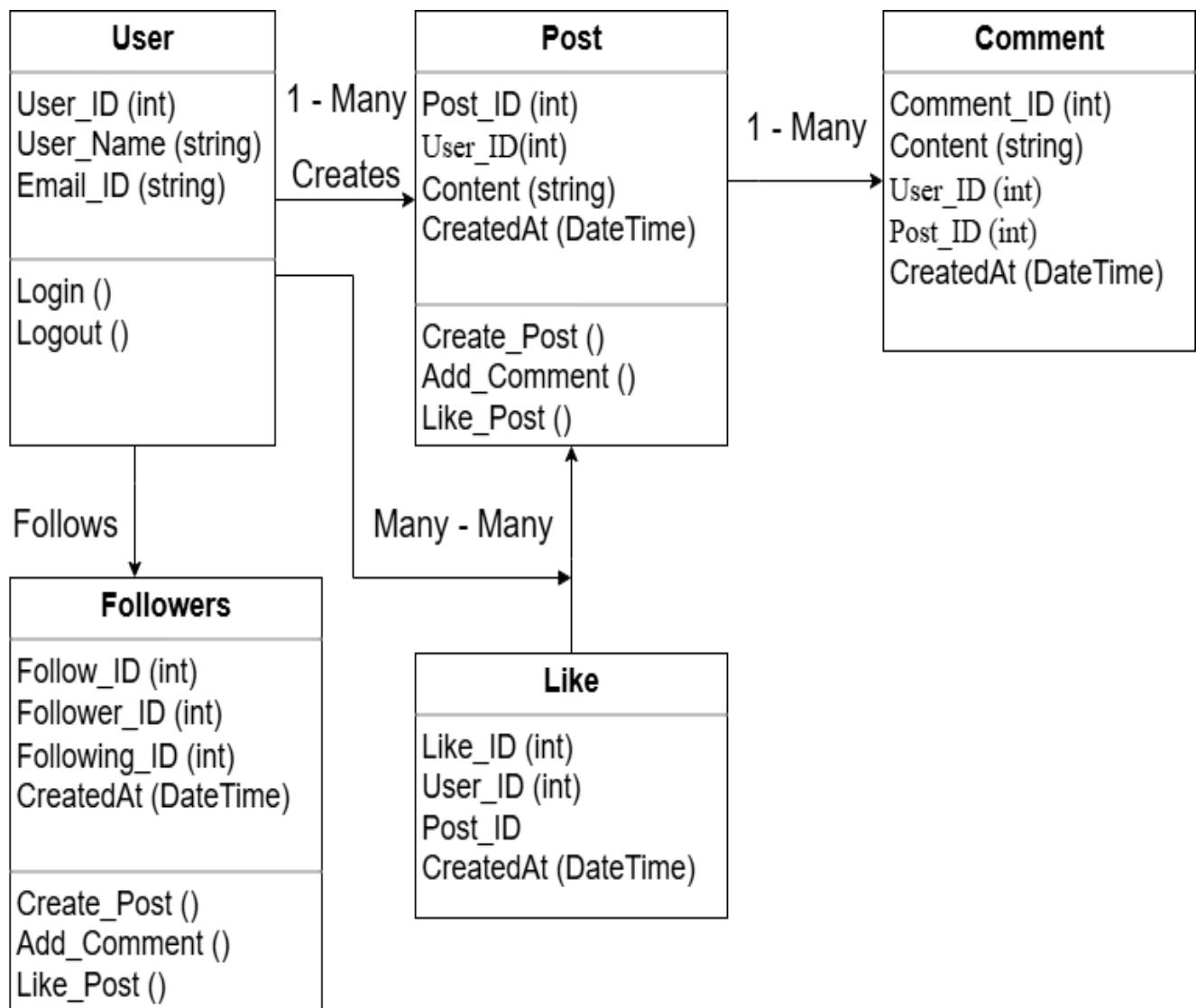
Relationships:

- 1. User – Post:** 1 to Many (a user can create many posts).
- 2. User – Comment:** 1 to Many (a user can write many comments).
- 3. Post – Comment:** 1 to Many (a post can have many comments).
- 4. User – Like – Post:** Many to Many (a user can like many posts, a post can be liked by many users).
- 5. User – Follower – User:** Many to Many (users can follow many users and be followed by).

Use Case Diagram:



Class Diagram:



Chapter 4

System Design

4.1 Basic Modules

To develop a Social Networking Site with Fake News Detection, the system is organized into well-structured modules to ensure scalability, security, and efficiency. These modules handle everything from user interactions to detecting misinformation.

1. UI module

- React-based screens for signup/login, profile, feed, post, comments, and messages.
- Shows fake-news results on posts (warning badge or score).

2. Posting module

- Create, edit, delete, share text/images/links.
- Live feed updates with WebSockets/Socket.io.
- Tags and hashtags for discovery.

3. Fake-news module

- Backend ML/API analyzes posts and links automatically.
- Flags risky content and shows a credibility score/status.

4. Social module

- Follow/unfollow or friend requests.
- Likes, comments, shares, notifications.
- Direct messages/chats.

5. Analytics module

- Admin reports: flagged posts, active users, trends.
- Export reports as PDF/CSV.

6. Testing module

- Unit tests for front-end and back-end (Jest/Mocha).
- API tests (Postman/Supertest) and logs for debugging.

7. Documentation module

- Clear API docs, diagrams, and how-to guides.
- Developer notes for maintenance and upgrades.

8. Security module

- JWT login and role-based access (Admin/Moderator/User).
- HTTPS, encryption, anti-spam, and protection from SQL/NoSQL injection and XSS.

4.2 Data Design

The Data Design for the Social Networking Site with Fake News Detection describes how data will be structured, stored, and accessed within the system. Since the project is MERN-based, MongoDB will be used as the primary database. Collections are defined to handle users, posts, interactions, and fake news detection results.

Main Collections and Relationships

1. Users

- Fields: userId, username, email, passwordHash, profilePicture, bio, role, createdAt, updatedAt.
- Relations: one user makes many posts; users follow each other via a separate Follows collection (many-to-many).

2. Posts

- Fields: postId, userId (ref Users), content, createdAt, updatedAt, status: Pending/Verified/Flagged.
- Relations: one post has many comments; one post can have many fake-news reports over time.

3. Comments

- Fields: commentId, postId (ref Posts), userId (ref Users), content, createdAt.
- Relations: each comment belongs to one post and one user.

4. FakeNewsReports

- Fields: reportId, postId (ref Posts), credibilityScore (0–100), detectionResult (true/false), sourceVerification, createdAt.
- Relation choice: keep one latest report per post or store multiple reports; if single, enforce unique(postId).

5. Notifications

- Fields: notificationId, userId (receiver), type, message, createdAt, readAt (optional).
- Relation: each notification targets one user.

4.3 Procedural Design

The Procedural Design defines the workflows and processes for key features of the system. It explains how data flows between modules and how user actions are handled.

Key Processes:

1. Registration and login

- User fills form, backend validates, password is hashed, user record saved.
- Server returns a JWT; client stores it securely and sends it with future requests.

2. Create and share post

- User writes text/adds images or link; API saves the post and marks status = Pending.
- The post is queued/triggered for fake-news check, then shown in feed with current status.

3. Fake-news detection flow

- Extract text and any URLs from the post.
- Call the detection service/model; get a score and label (Verified/Flagged).
- Save a report entry, update post. Status, and push a notification if flagged.

4. Interactions (likes, comments, follows)

- Like/comment requests are validated and stored; duplicates are blocked with unique rules.
- Owner gets a notification (e.g., “X liked/commented”), feed counters update in real time.

5. Moderator review

- Dashboard lists Flagged posts with details and scores.
- Actions: Approve (mark Verified), Delete, or Warn user; all actions are logged and notify the creator.

6. Reliability and performance

- Use background jobs/queues for detection so posting feels instant.
- Add rate limits and input validation to every endpoint; use indexes for feed and review queries.

4.4 User Interface Design

The User Interface (UI) is designed to be simple, clear, and interactive, giving a smooth experience for all types of users — Users, Moderators, and Admins.

Main Pages

1. Login & Signup Page

- Fields: Username, Email, Password.
- Buttons: *Login, Sign Up, Forgot Password*.
- Handles invalid login attempts securely.

2. User Dashboard (News Feed)

- Shows posts from friends and followed users.
- Each post displays content, credibility score, and a *fake news warning* if flagged.
- Users can *Like, Comment, or Share* posts.

3. Post Creation Page

- Fields for text, media upload, and link preview.
- Clicking *Post* starts the fake news detection process.

4. Admin/Moderator Panel

- Table showing flagged posts with filters (by user, date, score).
- Options to *Approve, Delete, or Ban/ Warn* a user.

5. Notifications Page

- Shows updates like new followers, comments, and post status.

Design Elements

- Buttons: Rounded and easy to identify.
- Forms: Show error or validation messages.
- Tables: Used for admin moderation.
- Colors: Neutral background, red/yellow for warnings, green for verified posts.
- Fonts: Modern and clean, with clear hierarchy.

4.5 Security Issues

Security is very important since users share personal data and posts. The platform includes several security measures:

1. User Authentication

- Use strong passwords with a mix of letters, numbers, and symbols.
- MFA (Multi-Factor Authentication) for admins/moderators adds extra protection.
- Limit failed login attempts to stop hackers.
- Use JWT tokens for secure login sessions.

2. Data Encryption

- Use HTTPS to encrypt all communication.
- Store passwords and sensitive data in encrypted form.
- Make sure API calls (like fake news detection) are also encrypted.

3. Input & API Security

- Validate and sanitize inputs to prevent SQL injection and XSS attacks.
- Limit API requests per user to avoid misuse.

4. Session Management

- Use secure cookies.
- Auto-logout users after inactivity.
- Invalidate tokens when users log out or change passwords.

5. Fake News Detection Integrity

- All posts are checked on the server before being published.
- Store detection results (credibility scores) securely so they can't be changed.

6. Secure Deployment

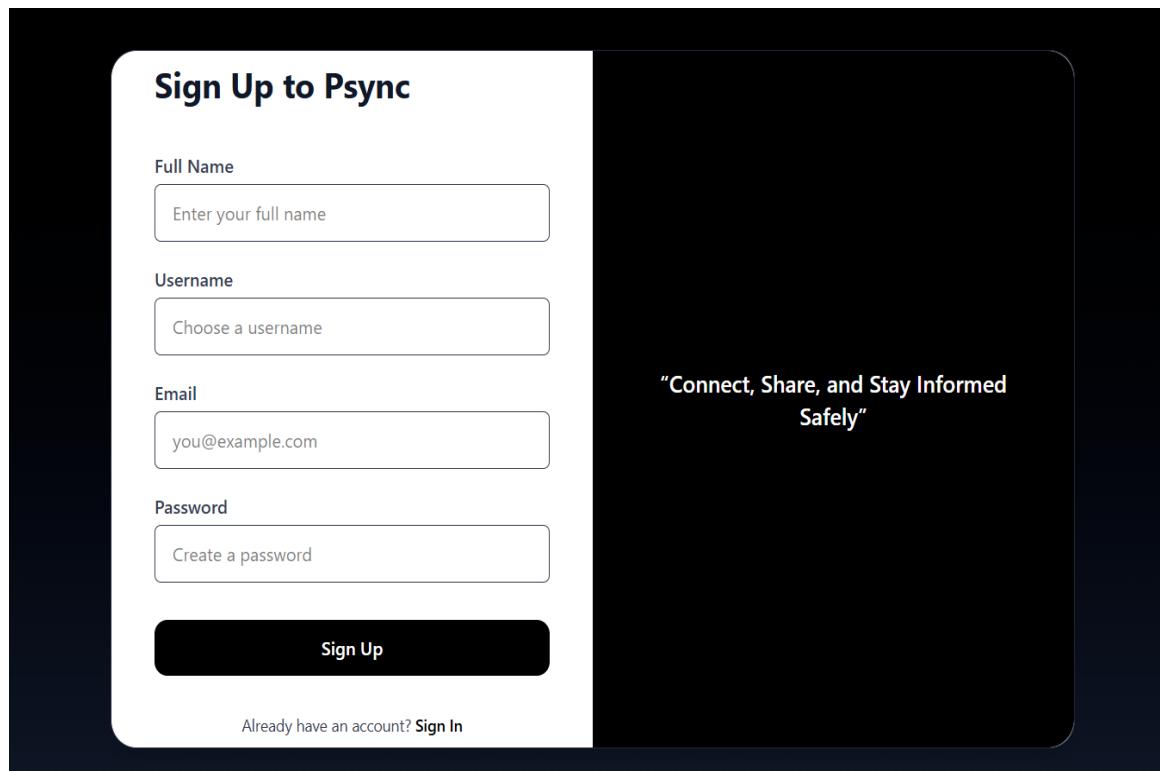
- Keep software updated (Node.js, MongoDB, etc.).
- Use firewalls and proxies for protection.
- Enable CORS to stop unauthorized access.

7. User Privacy & Data Protection

- Follow privacy laws (like GDPR).
- Allow users to manage or delete their data anytime.

➤ Psync Webpages:

1. Login Page



The image shows a 'Sign Up to Psync' form on a dark background. The form includes fields for Full Name, Username, Email, and Password, each with a placeholder text. A 'Sign Up' button is at the bottom, and a link to 'Sign In' is at the bottom right. To the right of the form, there is a slogan: "Connect, Share, and Stay Informed Safely".

Sign Up to Psync

Full Name
Enter your full name

Username
Choose a username

Email
you@example.com

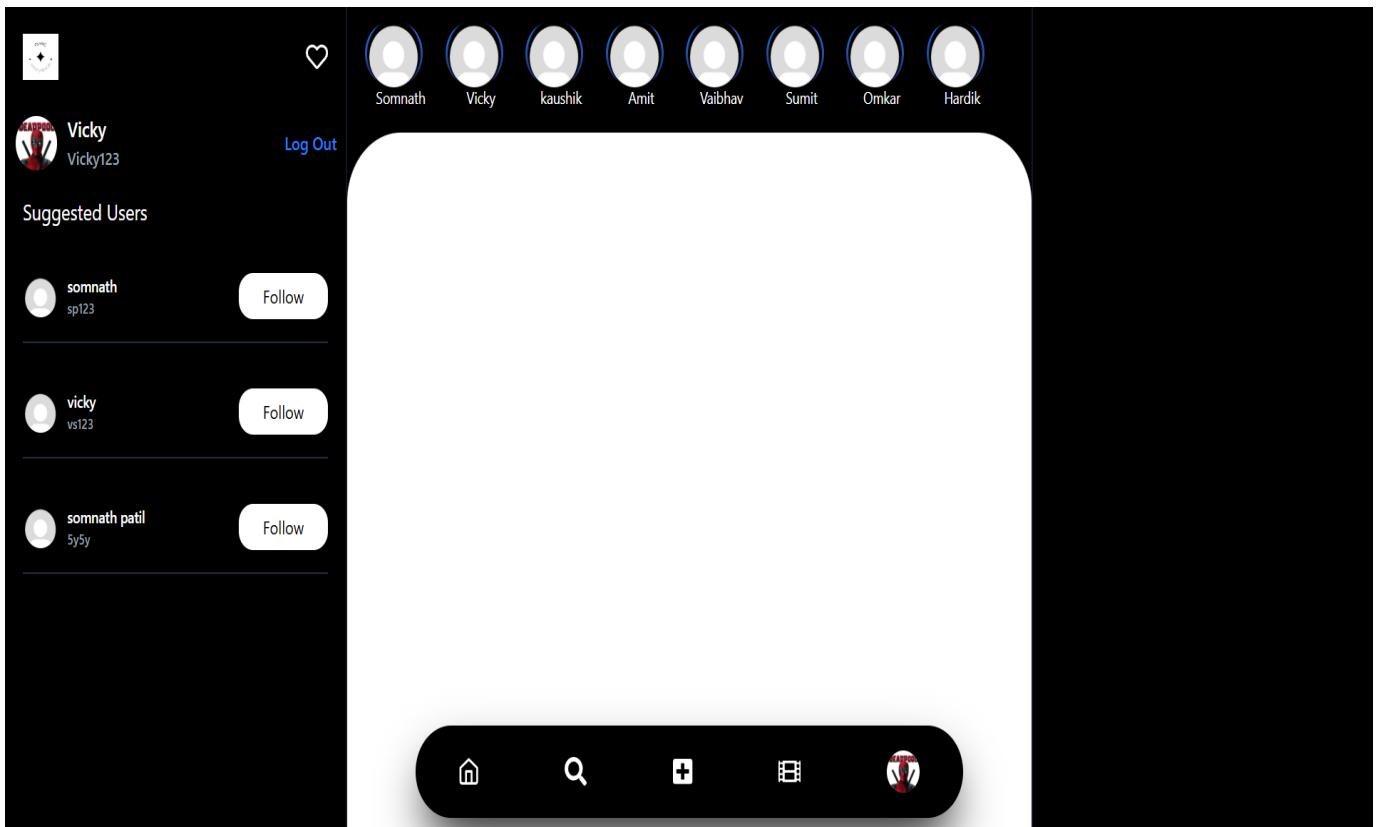
Password
Create a password

Sign Up

Already have an account? [Sign In](#)

"Connect, Share, and Stay Informed
Safely"

2. Home Page



3. Profile Page

