

DESIGN AND IMPLEMENTATION OF A SECURE COLLEGE-CENTRIC SOCIAL NETWORKING PLATFORM USING MERN STACK

¹Prof. Chaitali Chate, ²Prajwalagouda S. Dalawai, ³Prateek Pandarikar, ⁴Neha Vithob Nayak, ⁵Nikhil Hegadi

¹²³⁴⁵Department of Computer Science and Engineering

¹²³⁴⁵S. D. M. College of Engineering & Technology, Dharwad, Karnataka, India

Abstract—This paper presents a secure, college-centric academic social networking platform that addresses challenges such as fragmented communication, lack of institutional security, and unstructured collaboration in higher education environments. The system is developed using the MERN stack, incorporating MongoDB for scalable data storage, Express.js and Node.js for backend services, and React.js for a responsive user interface. JWT-based authentication ensures verified access restricted to institutional users, while real-time communication is enabled through Socket.io. The platform supports core academic networking functionalities including user profile management, content and post sharing, friend-based networking, real-time messaging, and notification services within a controlled ecosystem. The system architecture follows the Model–View–Controller (MVC) design pattern, enabling modular development, scalability, and maintainability. Experimental deployment demonstrates improved communication efficiency, secure data handling, and enhanced user engagement. The proposed platform provides a reliable foundation for academic collaboration and can be extended to support intelligent recommendation, mentorship, and analytics-driven academic services.

Index Terms—Academic Social Networking, College-Centric Platform, MERN Stack, JWT Authentication, Real-Time Communication, Web Application Security, MongoDB

I. Introduction

Effective communication and collaboration are essential in higher education; however, many institutions rely on generic social media and fragmented tools that lack academic focus, institutional security, and structured interaction. This often leads to privacy concerns, unverified access, and unorganized academic engagement.

To address these challenges, this paper presents **SDM HUB**, a secure, college-centric academic social networking platform designed for institutional communities. The platform enables verified users to connect through features such as profile management, content sharing, friend-based networking, real-time messaging, and notifications. Developed using the MERN stack and following the Model–View–Controller (MVC) architecture, SDM HUB provides a scalable and maintainable solution that enhances academic collaboration and supports a digitally connected campus ecosystem.

OBJECTIVES

1. To ensure secure, email-based authentication for verified access by students, faculty, and alumni.
2. To support academic and social content sharing for structured interaction within the college community.
3. To provide personalized news feeds displaying relevant posts, comments, and activities.

4. To enable real-time, secure messaging for instant user communication.
5. To implement a connection and notification system for friend requests and activity alerts.

II. Literature review

Several platforms have been developed to support academic communication and collaboration in higher education. CampusConnect (2022) utilized the MERN stack to provide basic features such as login, posts, and forums, but lacked real-time communication and mobile responsiveness. EduNet (Desai et al., 2021) implemented a Django-based portal focused on resource sharing and discussion boards; however, it offered limited user interaction features and lacked notification and friend systems.

CollabHub (2022) employed a REST-based architecture to support collaborative projects, but did not include direct messaging or social networking capabilities. CollegeCircle (2020) adopted a mobile-first approach using Firebase and Flutter, yet lacked role-based access control and scalability. Similarly, AcadShare Platform (Jain et al., 2023) enabled academic content sharing using Laravel and MySQL, but suffered from security and scalability limitations.

These studies highlight the need for a secure, scalable, and institution-centric academic networking platform with real-time interaction and structured access control. The proposed SDM HUB addresses these limitations by providing verified authentication, real-time communication, and a modular architecture tailored for academic environments.

III. System Architecture and Workflow

System Architecture

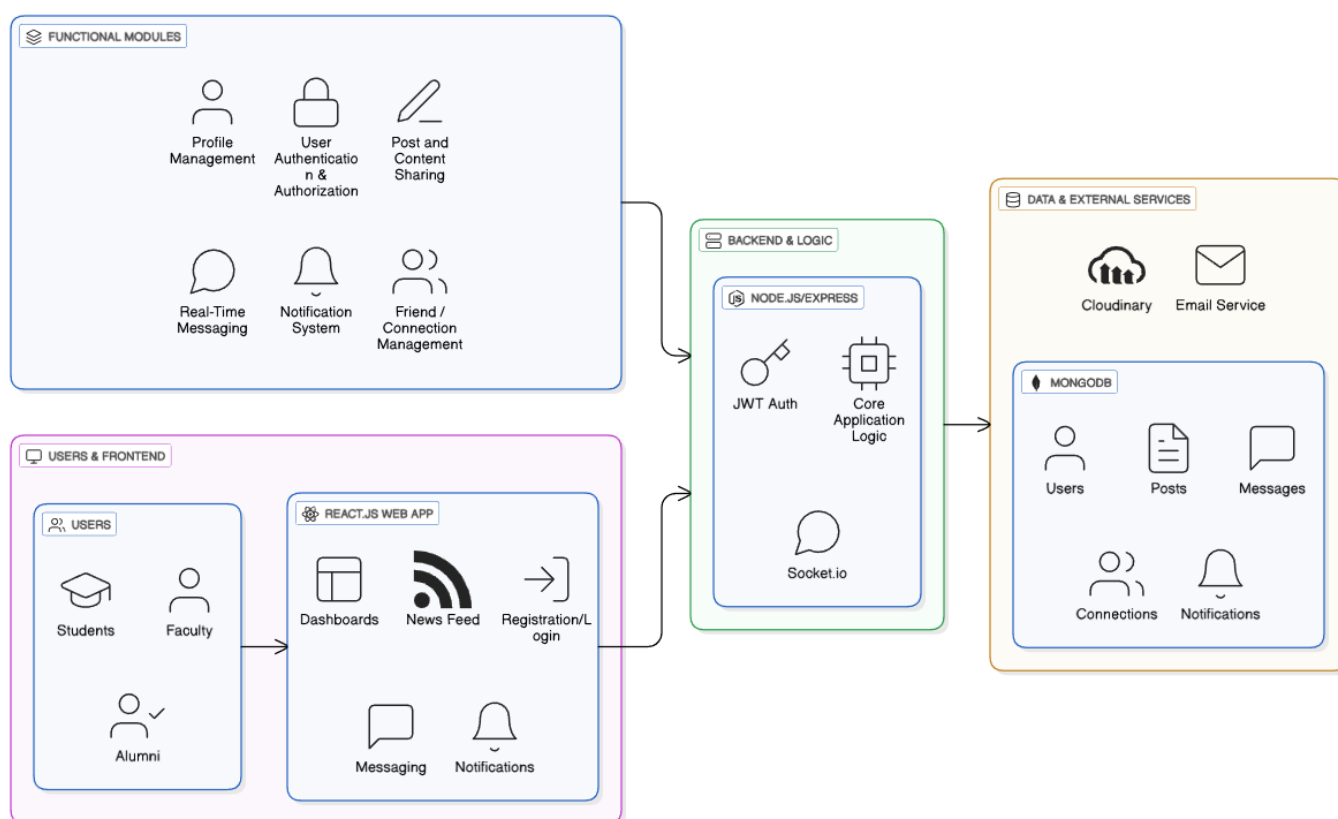


Fig. 1: System architecture

1. System Architecture and User Interface Layer

SDM HUB is designed as a modular, layered, college-centric academic social networking platform to ensure scalability, security, and maintainability. The system supports three primary user groups **Students, Faculty, and Alumni** who interact through a web-based frontend. The user interface is implemented using **React.js**, providing role-based dashboards that enable users to access platform features such as registration and login, profile management, news feed browsing, messaging, and notifications. This structured interface ensures seamless interaction, personalized content delivery, and efficient academic networking within the institutional ecosystem.

2. Backend Services and Core Functional Modules

The backend acts as the central processing unit of the system and is developed using **Node.js and Express.js**. It exposes secure application services to handle user authentication, content management, friend and connection handling, messaging, and notification workflows. Core functional modules include **user authentication and authorization, profile management, post and content sharing, real-time messaging, notification handling, and friend/connection management**. The backend validates user requests, enforces access control, and manages interactions between the frontend, database, and real-time services to ensure secure and reliable system operation.

3. Real-Time Communication and Application Logic Layer

To support instant interaction, SDM HUB integrates **Socket.io** within the backend for real-time messaging and notification delivery. This enables users to exchange messages and receive alerts for friend requests, post interactions, and updates without page reloads. **JWT-based authentication** is employed to ensure secure session handling and authorized access across all system components. The separation of core application logic and real-time communication enhances system performance and supports concurrent user interactions efficiently.

4. Data Management and External Services Integration

MongoDB is used as the primary NoSQL database to store user profiles, posts, messages, connections, and notification data, enabling flexible and scalable data management. For media handling, **Cloudinary** is integrated to support secure image and media uploads. An **email service** is incorporated to manage account verification and system alerts. User requests flow from the frontend to the backend services, where data is processed, stored, and retrieved from the database before being presented back to users through the frontend interface. This architecture ensures a seamless, secure, and efficient end-to-end workflow.

Flow Diagram

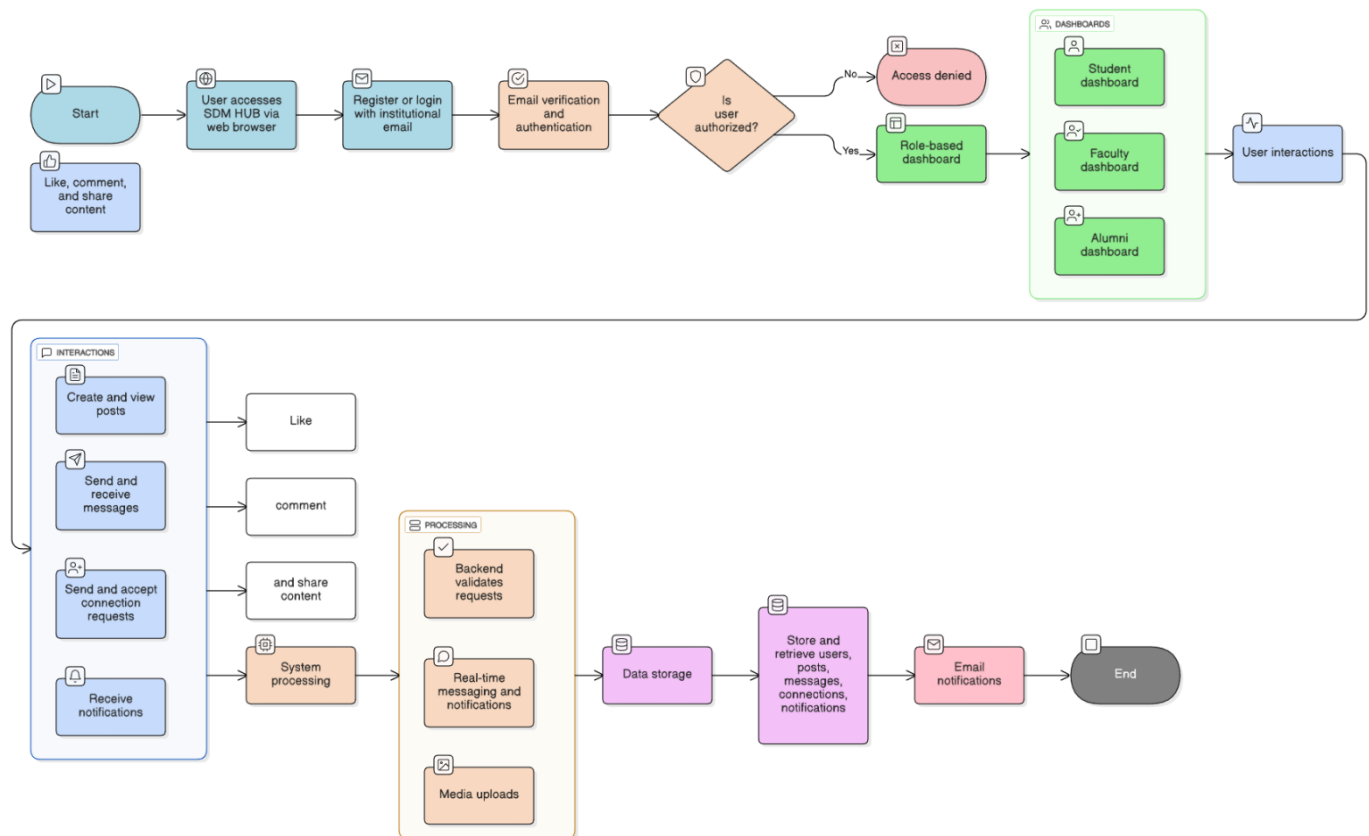


Fig. 2: Data Flow Diagram

It illustrates the end-to-end workflow of the system, beginning with user authentication and role verification. Authorized users access role-based dashboards to interact through posts, messaging, connections, and notifications, with all data securely processed and stored by the backend. The process begins when a user accesses **SDM HUB** through a web browser.

- The user registers or logs in using a **valid institutional email address**.
- The system performs **email verification and authentication** to validate the user.
- A decision is made to determine whether the user is authorized.
- **Unauthorized users** are denied access, while **authorized users** are granted access.
- Authorized users are redirected to **role-based dashboards** based on their role (Student, Faculty, or Alumni).
- All user requests are validated and processed by the **backend services**.
- Real-time messaging and notifications are handled through the system's communication layer.
- User data, posts, messages, connections, and notifications are **stored and retrieved from the MongoDB database**
- Users interact with the platform by:
 - Creating and viewing posts
 - Liking, commenting, and sharing content
 - Sending and receiving messages
 - Sending and accepting connection requests
 - Receiving real-time notifications
- Email notifications are sent for verification and important system alerts.

- The workflow continues as users interact with the platform for ongoing academic networking and collaboration.

Use – Case Diagram

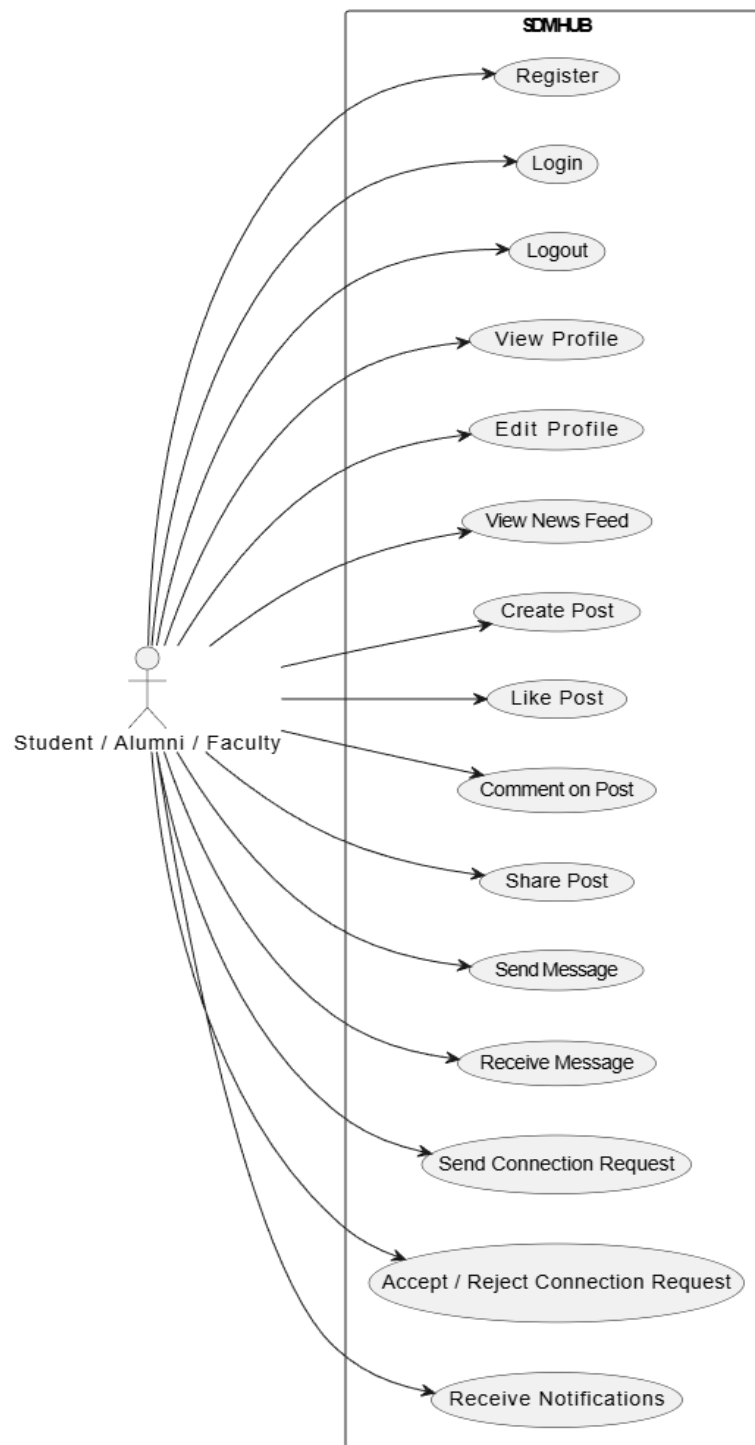


Fig. 3: Use Case Diagram

Project Schema

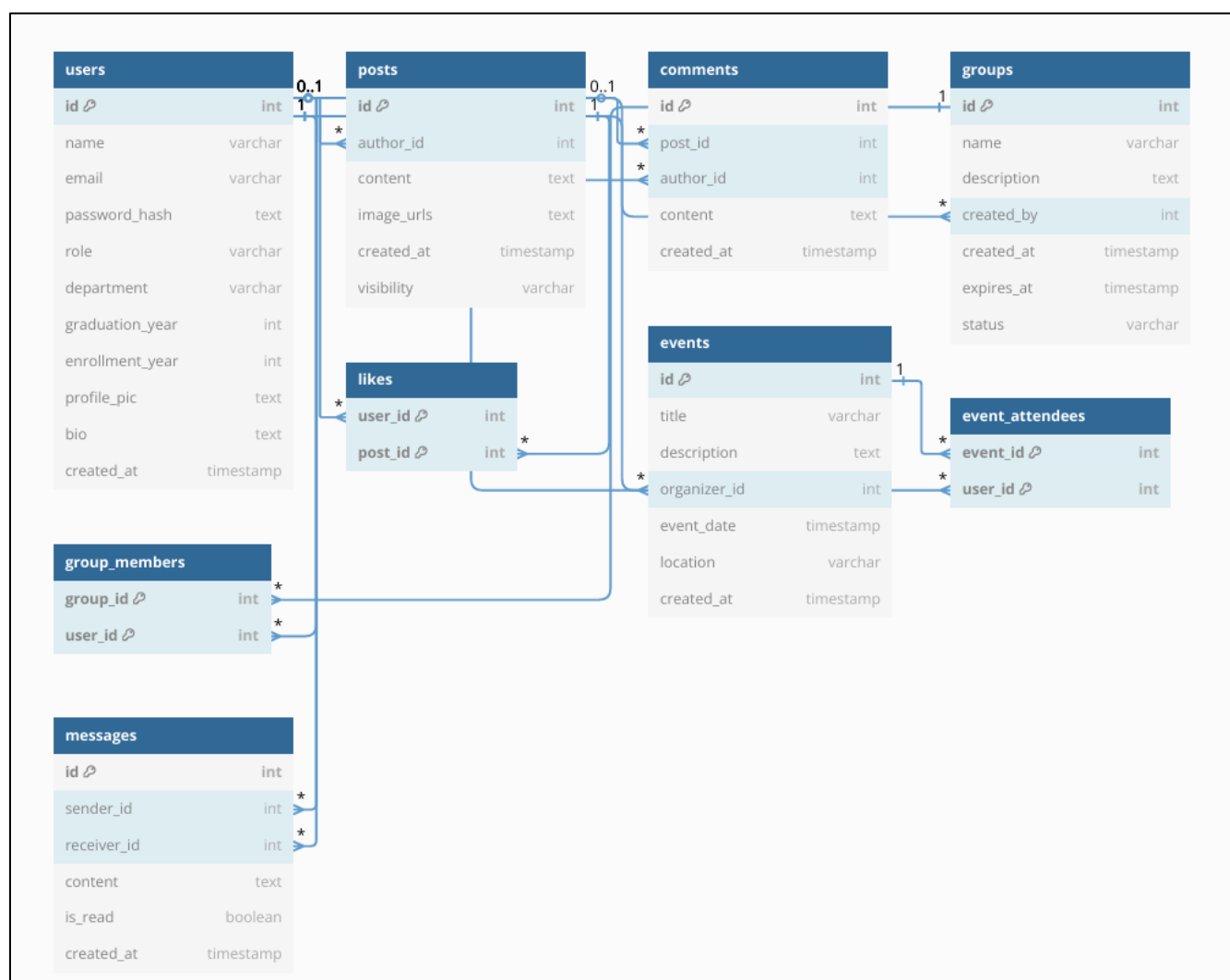


Fig. 4: Schema

IV. Implementation

The SDM HUB system is implemented as a secure, college-centric academic social networking platform using the MERN stack. The system is designed to support structured communication, collaboration, and networking among students, faculty, and alumni within an institution. Emphasis is placed on scalability, real-time interaction, data security, and ease of use.

¹ System Architecture

- The implementation is divided into three primary layers:

Presentation Layer: React.js-based responsive web interface providing role-based dashboards and interactive user experience.

Application Layer: Node.js and Express.js-based backend handling business logic, authentication, real-time communication, and API processing.

Data Layer: MongoDB NoSQL database used for efficient storage and retrieval of user data, posts, messages, connections, and notifications.

- Each module operates independently, enabling easier debugging and future enhancements.

² Technology Stack

- Frontend Framework:** React.js

- **Backend Framework:** Node.js with Express.js
 - **Database:** MongoDB (NoSQL)
 - **Real-Time Communication:** Socket.io
 - **Authentication:** email-based login with JWT tokens
 - **APIs:** RESTful APIs for frontend–backend interaction
 - **Security:** JWT authentication, password hashing
 - **Development Tools:** Visual Studio Code, Git, GitHub
 - **Testing Tools:** Postman, Thunder Client
 - **Design Tools:** Draw.io for UML/ER diagrams, Stitch for UI designs
- **Module-Wise Implementation**
 - a. **1. User Authentication Module**
 - i. Supports institutional email-based user registration with role assignment
 - ii. Ensures secure login using JWT-based authentication and backend middleware validation.
 - iii. Implements password hashing before storing credentials in MongoDB.
 - iv. Maintains secure user sessions using token-based access control.
 - b. **2. User Profile Management Module**
 - i. Allows users to view, and edit academic and professional profile details.
 - ii. Supports secure profile image upload using cloud-based storage services.
 - iii. Stores profile information using Mongoose schemas for structured data handling.
 - iv. Reflects profile updates dynamically on the frontend interface.
 - c. **Friend Request and Network Module**
 - i. Enables users to send, accept, reject, and cancel connection requests.
 - ii. Maintains friend relationships securely within user data schemas.
 - iii. Calculates mutual friends using efficient set-based operations.
 - d. **Post and Content Management Module**
 - i. Allows users to upload and view posts containing text or media.
 - ii. Supports interaction through likes, comments, and content sharing.
 - iii. Stores posts with author metadata and timestamps in the database.
 - iv. Generates a personalized news feed with real-time content updates.
 - e. **Real-Time Messaging Module**
 - i. Enables secure one-to-one communication between connected users.
 - ii. Supports real-time message delivery using Socket.io.
 - iii. Stores chat history using Conversation and Message schemas.
 - f. **Notification Module**
 - i. Generates notifications for friend requests, post interactions.
 - ii. Uses a dedicated Notification schema for structured storage.
 - iii. Pushes real-time alerts to users via Socket.io.
 - iv. Allows users to mark notifications as read for better interaction tracking.

g. Search and Recommendation Module

- i. Allows users to search for other members using name or email.
- ii. Implements backend search APIs with indexed queries for efficiency.
- iii. Recommends users based on mutual friend relationships.
- iv. Ranks suggestions dynamically using mutual connection counts.

• Data Management

MongoDB is used as the primary data store for:

- User profiles and roles
- Posts, comments, and likes
- Messages and conversations
- Friend connections
- Notifications

The flexible schema design supports evolving platform requirements and efficient querying.

• Security and Privacy

- Role-based access control implemented throughout the system
- Secure authentication using JWT tokens
- All data transmission encrypted using HTTPS
- Passwords securely hashed before storage
- Only authenticated users can access internal platform features

V. Result and Discussion

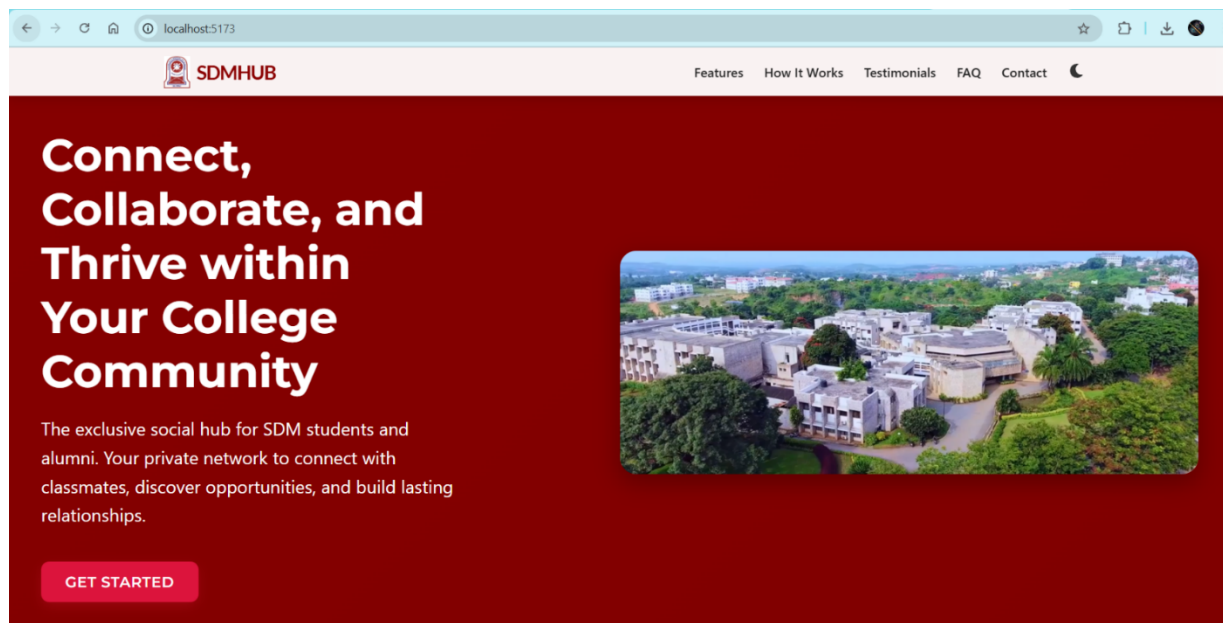
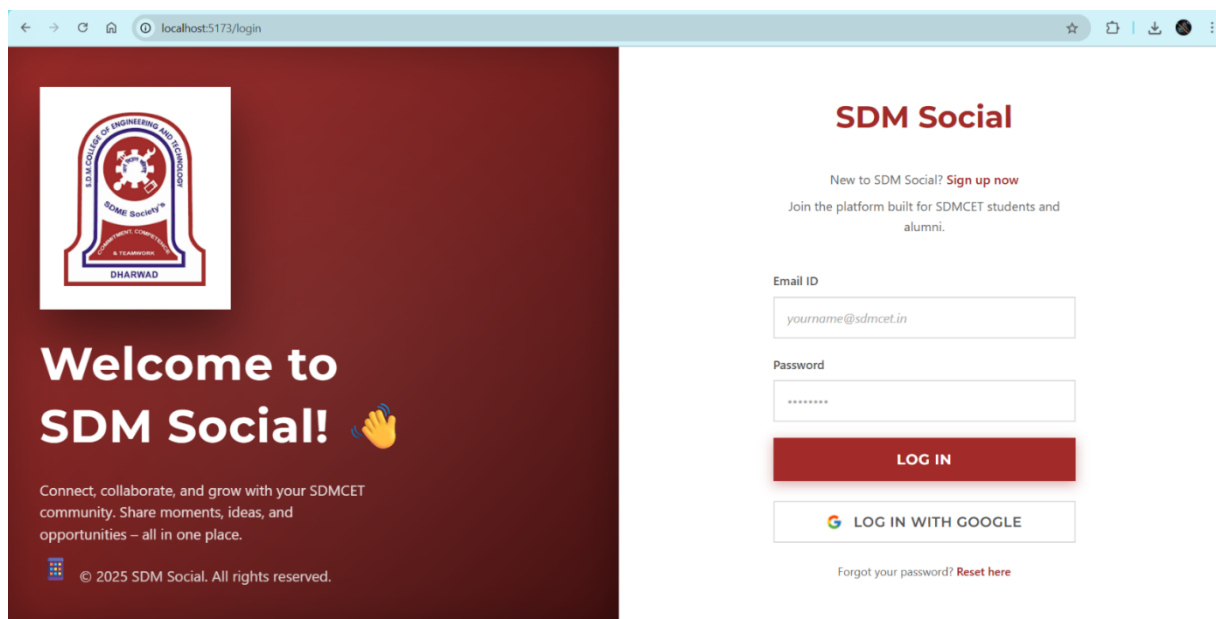


Fig. 5: Landing page



SDM Social

New to SDM Social? [Sign up now](#)

Join the platform built for SDMCET students and alumni.

Email ID

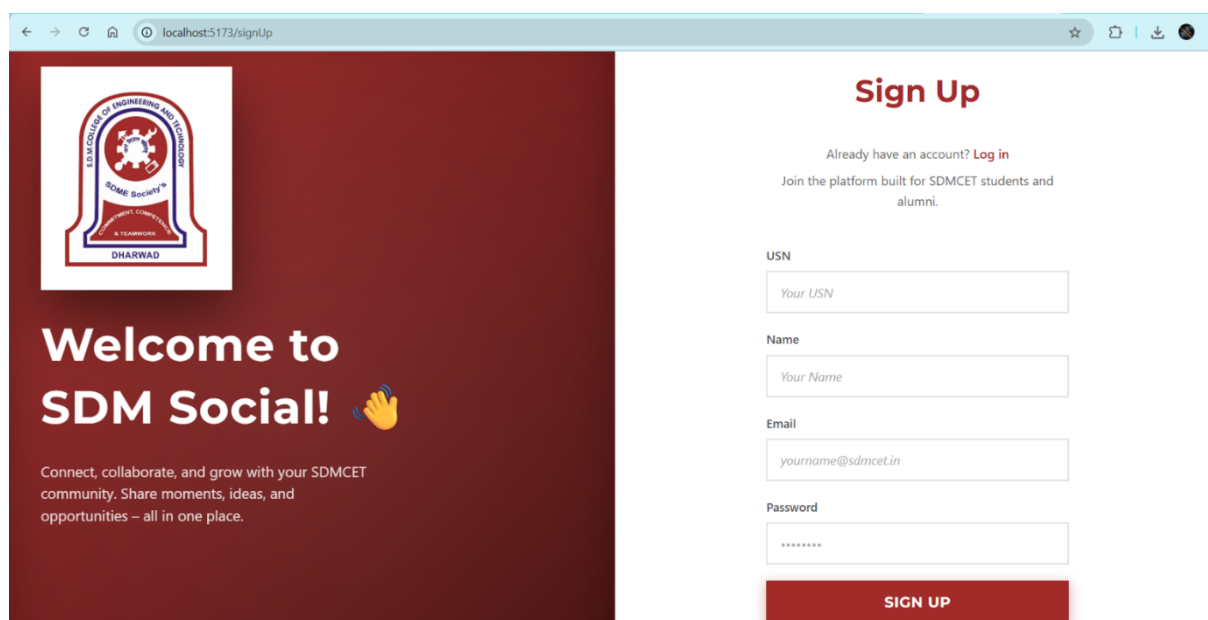
Password

LOG IN

LOG IN WITH GOOGLE

Forgot your password? [Reset here](#)

Fig. 6: Login page



Sign Up

Already have an account? [Log in](#)

Join the platform built for SDMCET students and alumni.

USN

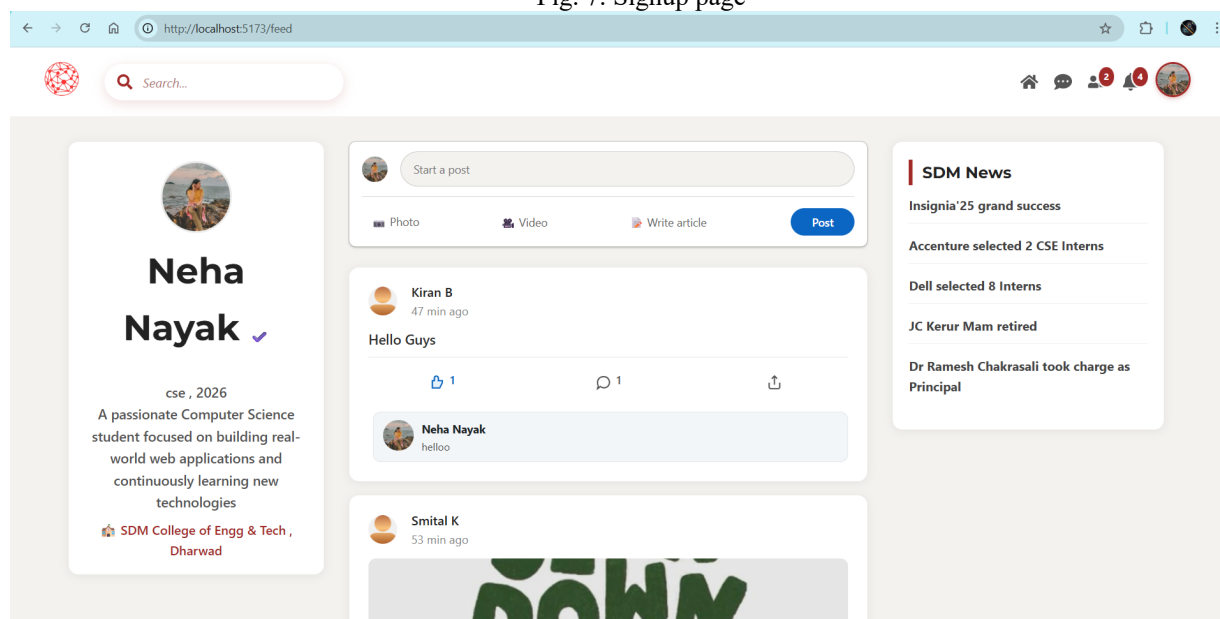
Name

Email

Password

SIGN UP

Fig. 7: Sign up page



Neha Nayak ✓
 cse , 2026
 A passionate Computer Science student focused on building real-world web applications and continuously learning new technologies
 SDM College of Engg & Tech , Dharwad

Start a post

Photo Video Write article **Post**

Kiran B
 47 min ago
 Hello Guys
 1 Like 1 Comment

Neha Nayak
 hello

Smital K
 53 min ago

SDM News

Insighia'25 grand success

Accenture selected 2 CSE Interns

Dell selected 8 Interns

JC Kerur Mam retired

Dr Ramesh Chakrasali took charge as Principal

Fig. 8: Feed page

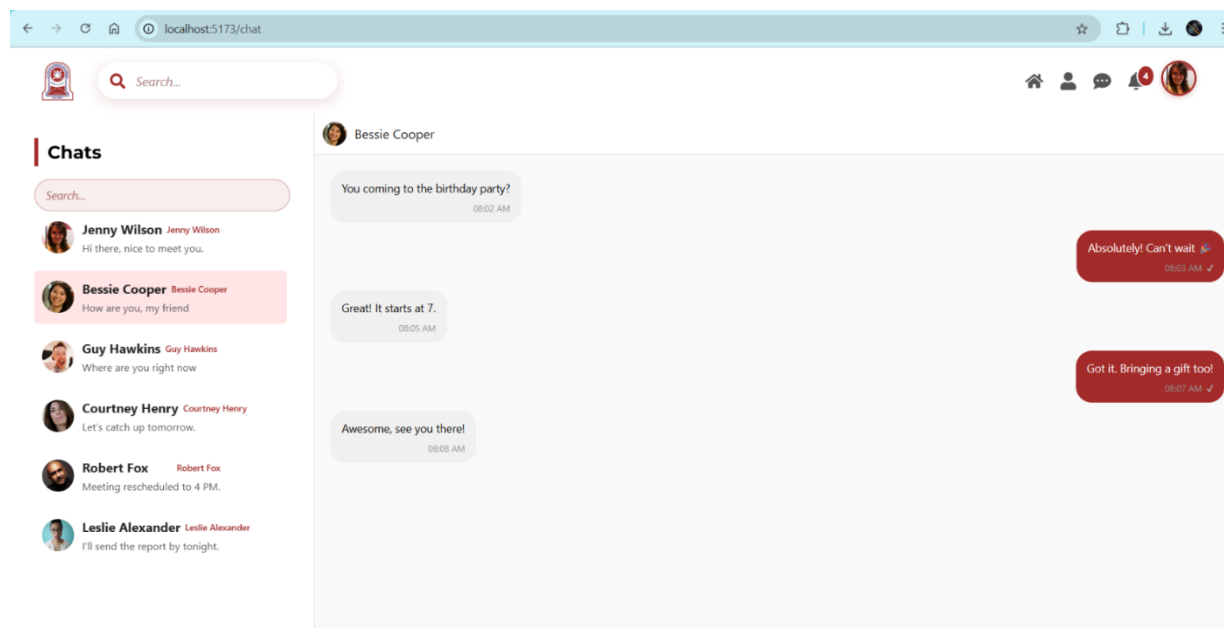


Fig. 9: Messaging and Content Sharing page

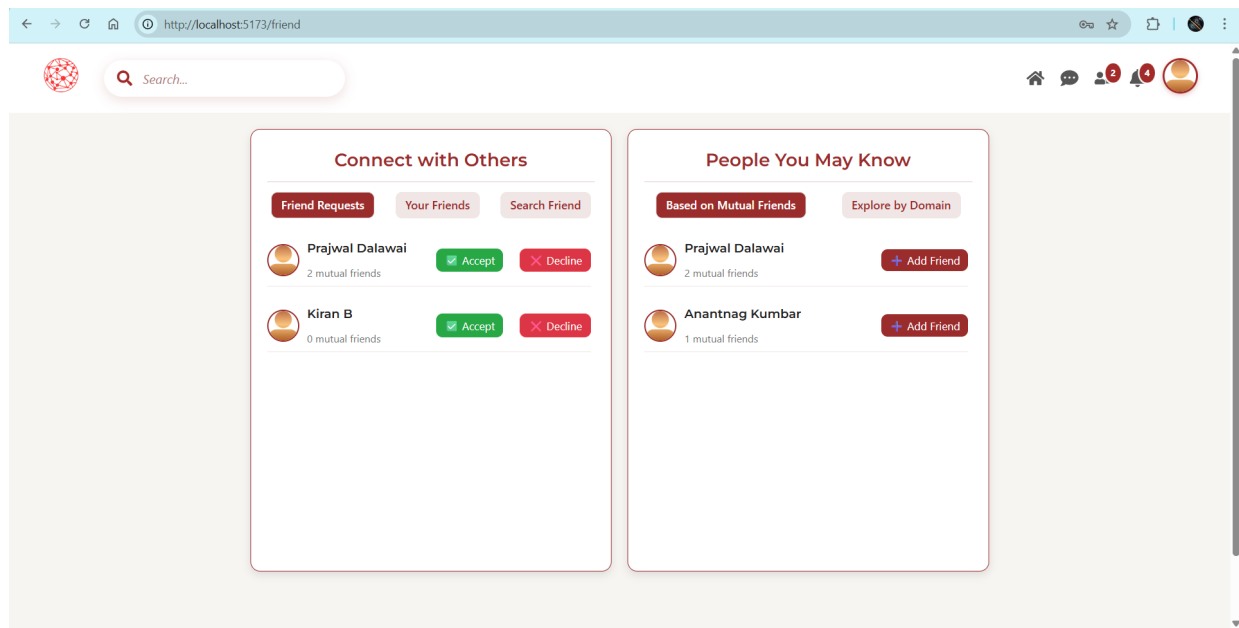


Fig. 10: Friend Request page

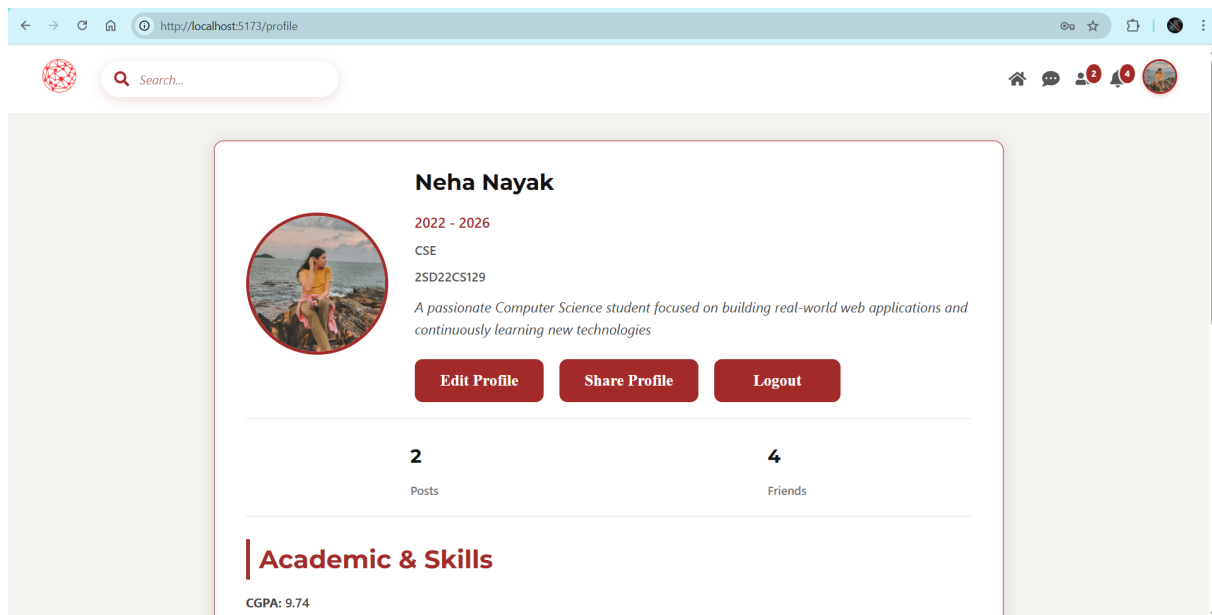


Fig. 11: Profile Page

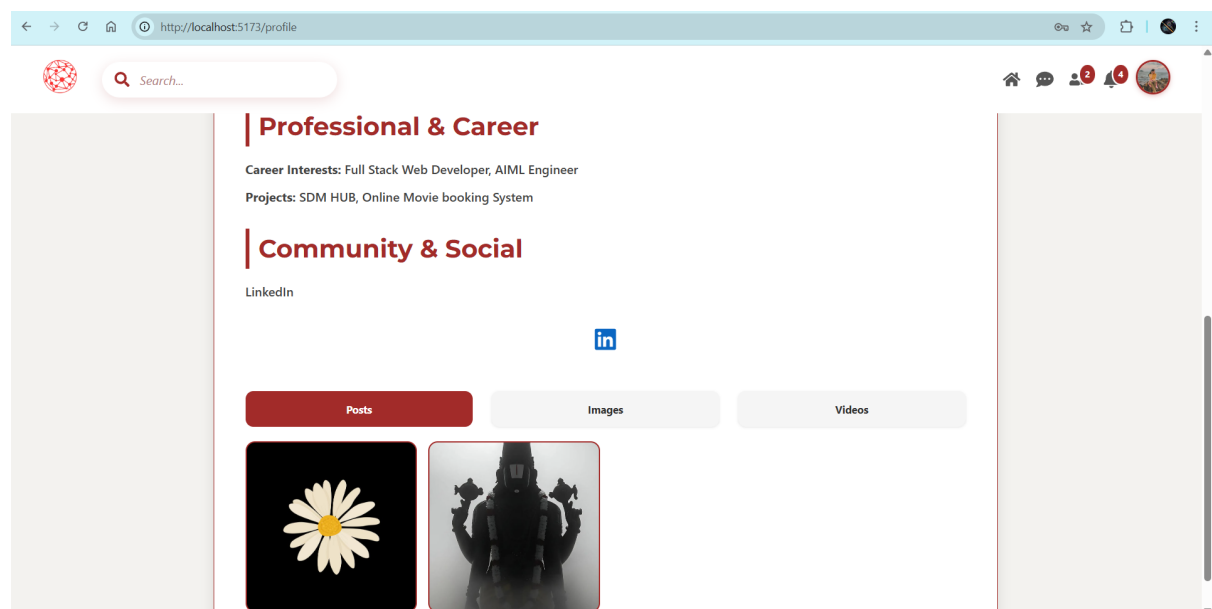


Fig. 12: Profile Page

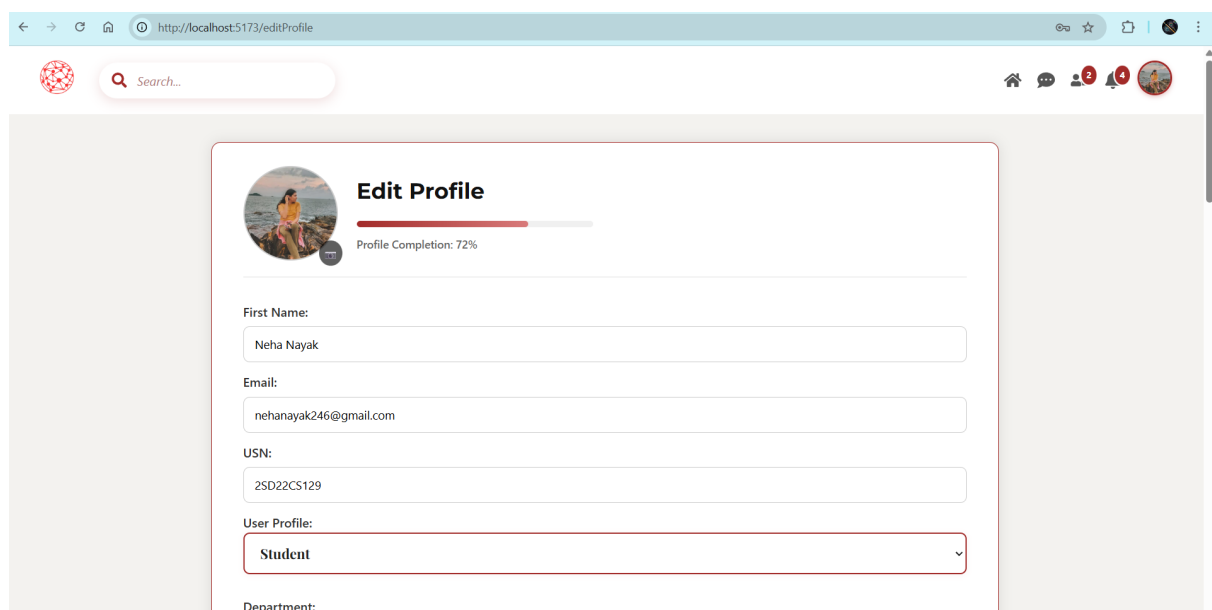


Fig. 13: Edit Profile page

VI. Conclusion

Effective communication and collaboration play a crucial role in higher education, directly influencing academic engagement, knowledge sharing, and professional networking among students, faculty, and alumni. However, traditional communication methods and generic social media platforms often lack institutional security, academic focus, and structured interaction, leading to fragmented communication and privacy concerns. The proposed system, SDM HUB, addresses these challenges by providing a secure, college-centric academic social networking platform tailored specifically for educational environments.

SDM HUB integrates essential academic networking features such as verified user authentication, profile management, content and post sharing, friend-based networking, real-time messaging, and notification services within a controlled ecosystem. The use of the MERN stack ensures scalability, modularity, and maintainability, while real-time communication enabled through Socket.io enhances user engagement and timely interaction. Role-based access for students, faculty, and alumni ensures secure participation and appropriate access to platform features.

Overall, SDM HUB offers a reliable and efficient solution for structured academic communication and collaboration. By fostering a digitally connected campus environment, the platform enhances academic interaction, supports knowledge exchange, and strengthens institutional networking. The system can be further extended with advanced analytics, intelligent recommendations, and mentorship features to support evolving academic and professional needs.

VII. Acknowledgement

The authors would like to express their sincere gratitude to the faculty of the Department of Computer Science and Engineering, Shri Dharmasthala Manjunatheshwar College of Engineering and Technology (SDMCET), Dharwad, for their valuable guidance and continuous support throughout this work. The authors extend special thanks to Prof. Chaitali Dixit for her insightful guidance, encouragement, and constant motivation during the development of this project. The authors also wish to acknowledge the contributions of the project team members Prajwalagouda S. Dalawai, Prateek Pandarikar, Neha Vithob Nayak, and Nikhil Hegadi for their cooperation and collaborative efforts. Further appreciation is extended to the Head of the Department and the Principal of SDMCET, Dharwad (Autonomous, affiliated to Visvesvaraya Technological University, Belagavi, Karnataka, India), for providing the necessary facilities and a supportive academic environment to carry out this work.

References

- [1] A. Boyd and M. Ellison, "Social network sites: Definition, history, and scholarship," *Journal of Computer-Mediated Communication*, vol. 13, no. 1, pp. 210–230, 2007.
- [2] N. Alalwan, A. Rana, and Y. Dwivedi, "Factors influencing adoption of social networking sites in academic environments," *Computers & Education*, vol. 109, pp. 93–108, 2017.
- [3] M. Newman and J. Park, "Online social networks and their role in academic collaboration," *IEEE Computer*, vol. 46, no. 7, pp. 68–75, 2013.
- [4] MongoDB Inc., "MongoDB: A scalable NoSQL database," [Online]. Available: <https://www.mongodb.com>
- [5] R. Fielding, "Architectural styles and the design of network-based software architectures," Ph.D. dissertation, University of California, Irvine, 2000.
- [6] OpenJS Foundation, "Node.js documentation," [Online]. Available: <https://nodejs.org>