|  |
| --- |
| **Full Stack Engineering**      Project Report  Semester-V (Batch-2023)    **Social Media Mini Platform**      **Supervised By: Submitted By:**  Rahul Sir Tushar Saxena -2310991152(G-4)  Vansh Thakur -2310991156(G-4)  Varun Choudhary -2310991158(G-4)  Parth Rana -2310991404(G-4)        **Department of Computer Science and Engineering**  **Chitkara University Institute of Engineering & Technology,**  **Chitkara University, Punjab** |

# ABSTRACT

The **Social Media Mini Platform** is a MERN-based web application designed to provide users with an engaging, secure, and interactive digital space. It enables individuals to create posts, share updates, like, comment, and connect with friends/followers in real time.

The platform focuses on **user privacy, scalability, and performance** by integrating JWT authentication, secure password encryption, and optimized MongoDB queries. With features like dynamic news feeds, real-time notifications, and interactive profile management, the application provides an intuitive and modern social networking experience.

Built using the **MERN stack (MongoDB, Express.js, React.js, Node.js)**, the project ensures responsiveness, scalability, and real-time interactions. It leverages WebSockets for live notifications and cloud deployment for accessibility. This report outlines the project’s problem definition, objectives, methodology, results, and scope for future enhancement.

.

# Table of Content

1. Abstract/Keywords

1. Introduction to the project

* 1. Background

* 1. Problem Statement

1. Problem Definition and Requirements

* 1. Methods

* 1. Programming/Working Environment

* 1. Requirements to run the application

1. Proposed Design / Methodology

1. Results

# Introduction

Current large-scale social media platforms are often overwhelming, heavy on resources, and raise privacy concerns. There is a need for a **mini social media platform** that:

* Focuses on core features (posts, likes, comments, follows)
* Ensures secure authentication and data privacy
* Provides real-time updates
* Remains scalable for growing user bases

1. **Background and Significance**

Social media has transformed how people communicate and share information. However, existing platforms often face **privacy issues, data overload, and lack of personalization**. Users seek lightweight, secure, and engaging platforms to stay connected.

The **Social Media Mini Platform** is designed to bridge this gap by providing essential social networking features in a **simple, secure, and scalable manner**.

1. **Objectives**

 Enable seamless **user engagement** through posts, likes, comments, and follows.

 Provide **real-time notifications** using WebSockets.

 Ensure **secure authentication** with JWT and password hashing.

 Build an **intuitive, responsive UI** using React.js.

 Create an **admin panel** for managing flagged content and user reports.

1. **Features and Functionality**

 **User Registration & Profile Management**

* + Create and manage user profiles.
  + Upload profile pictures and update bios.

 **News Feed & Posts**

* + Create, edit, and delete posts.
  + Like and comment in real-time.

 **Follow System**

* + Follow/unfollow users.
  + Personalized feed from followed accounts.

 **Notifications**

* + Real-time alerts for likes, comments, and follows.

 **Admin Panel**

* + Manage flagged posts and reported accounts.

1. **Technology Stack**

 **Frontend:** React.js (responsive UI, reusable components)

 **Backend:** Node.js + Express.js (APIs, authentication, notifications)

 **Database:** MongoDB (storing users, posts, comments, likes)

 **Authentication:** JWT, bcrypt.js

 **Real-Time Features:** WebSockets (Socket.IO)

 **Deployment:** Cloud hosting (AWS/Firebase/Heroku)

# Requirements

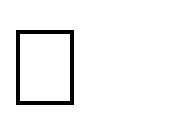
**Software Requirements:**

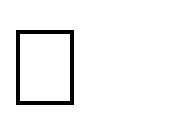
* **React.js for frontend**
* **Node.js + Express.js for backend**
* **MongoDB for database**
* **JWT for authentication**
* **Cloud hosting**

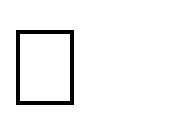
**Hardware Requirements:**

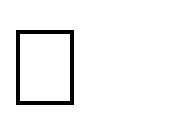
* **Minimum 8 GB RAM server**
* **SSD storage for fast queries**
* **Scalable cloud infrastructure**

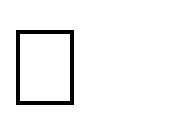
# Proposed Design / Methodology

 **Frontend: React with modular components, state management, and responsive design.**

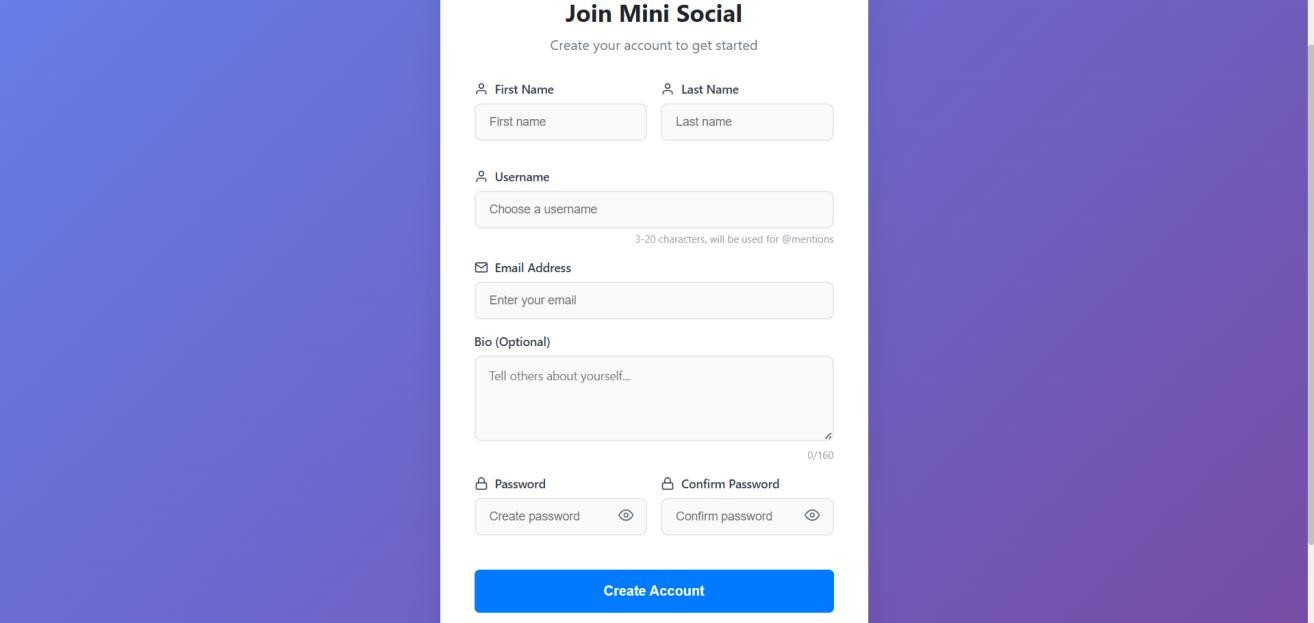
 **Backend: Node.js + Express.js with secure APIs and middleware for authentication.**

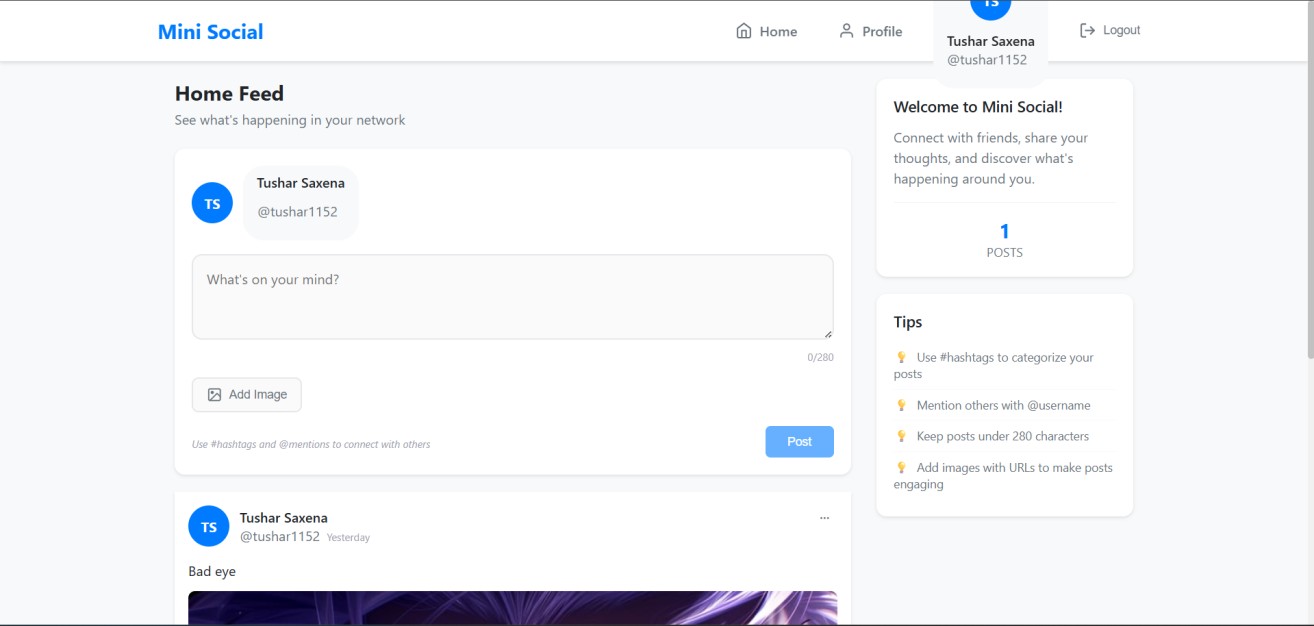
 **Database Models: Users, Posts, Comments, Likes, Follows, Reports.**

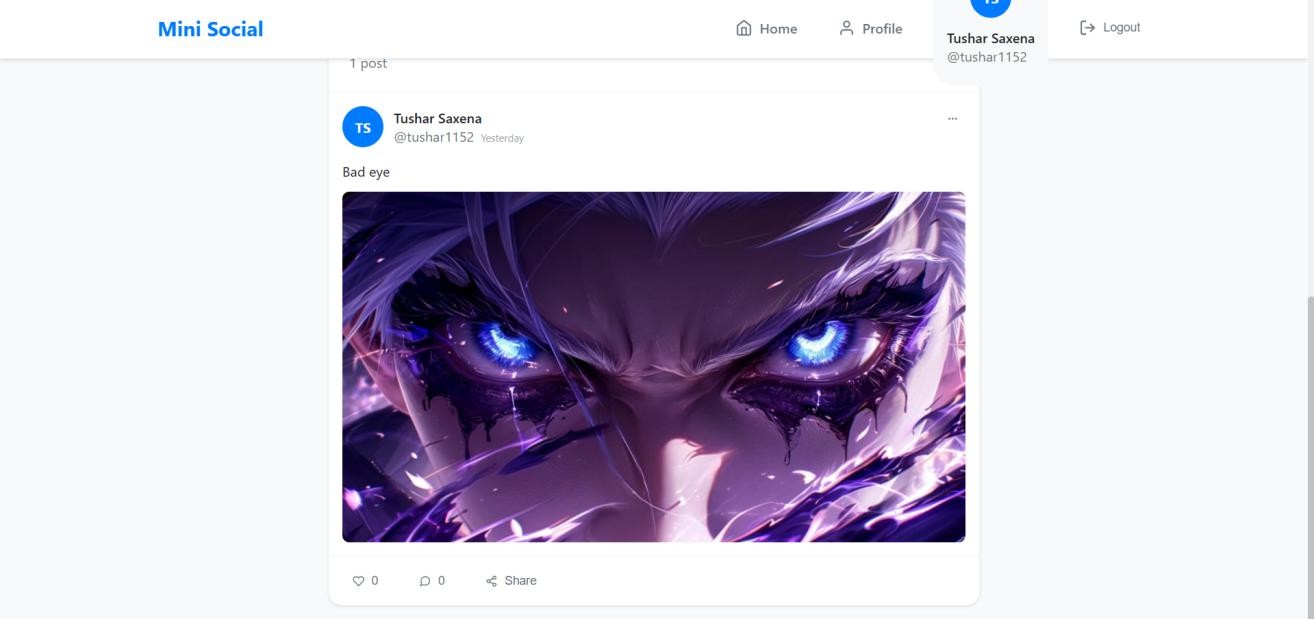
 **Real-Time Communication: Socket.IO for notifications and updates.**

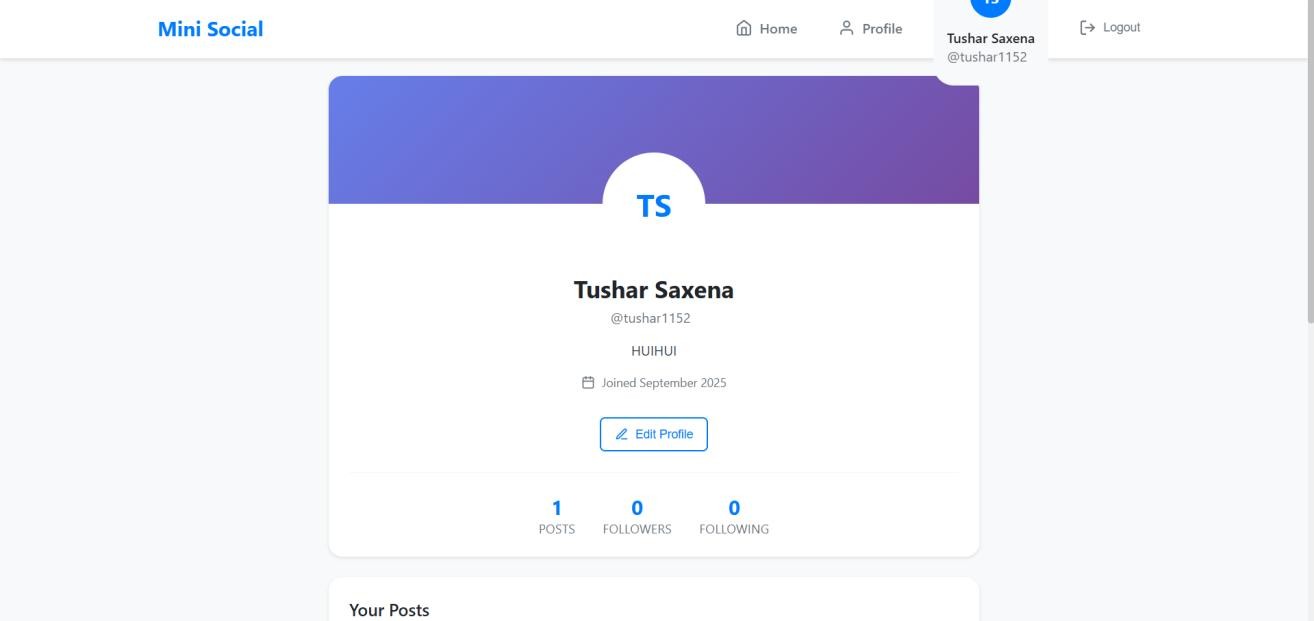
 **Admin Tools: Moderation system for safe community management.**

# Results









**References:**

**Official Documentation**: Documentation for libraries, frameworks, and tools used in the project, as well as APIs or services integrated.

**Tutorials and Guides**: Online tutorials, guides, blog posts, and educational videos that provided assistance or insights during development.

**Code Repositories**: GitHub repositories or other code repositories where code snippets, examples, or inspiration were found.

**Forums and Communities**: Online forums, such as Stack Overflow or Reddit, and developer communities where questions were asked, advice was sought, or discussions were participated in.

**Personal Communication**: Mentors, peers who provided guidance, feedback, or support during development.