

Project Report

**CommunityBoard: A Social Platform**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Only for course Teacher** | | | | | | |
|  | | **Needs Improvement** | **Developing** | **Sufficient** | **Above Average** | **Total Mark** |
| **Allocate mark & Percentage** | | **25%** | **50%** | **75%** | **100%** | **25** |
| **Understanding/Analysis** | **7** |  |  |  |  |  |
| **Implementation** | **8** |  |  |  |  |  |
| **Report Writing** | **10** |  |  |  |  |  |
| **Total obtained mark** | | | | | |  |
| **Comments** |  | | | | | |

**Semester:** Spring 2025

**Name:** Masuk A Alahi  **ID:** 221-35-989

**Name:** Md Redwan  **ID:** 221-35-813

**Name:** Md. Kawsar Uddin  **ID:** 221-35-987

**Name:** Md. Asif Khan  **ID:** 221-35-860

**Batch:** 37th  **Section:** D1  **Course Code:** SE331

**Course Name:** Software Engineering Design Capstone Project

**Course Teacher Name:** Foysal Khandakar Joy

**Designation:** Lecturer, Department of Software Engineering

**Submission Date:** 17/04/2025

**Index**

**Chapter 1: Introduction ........................................................................... 1  
1.1 Introduction ....................................................................................... 1  
1.2 Motivation ......................................................................................... 1  
1.3 Objectives .......................................................................................... 1  
1.4 Key Features ...................................................................................... 1  
1.5 Feasibility Study ............................................................................... 2  
1.6 Gap Analysis ..................................................................................... 2  
1.7 Project Outcome ............................................................................... 3**

**Chapter 2: Proposed Methodology/Architecture .................................. 3  
2.1 Requirement Analysis & Design Specification ................................ 3  
 2.1.1 Overview ................................................................................. 3  
 2.1.2 Proposed Methodology/System Design ................................ 3  
2.2 UI Design .......................................................................................... 3  
2.3 Overall Project Plan .......................................................................... 4**

**Chapter 3: Implementation and Results ............................................... 4  
3.1 Implementation ................................................................................ 4  
3.2 User & Profile API Endpoints .......................................................... 4  
3.3 Performance Analysis ...................................................................... 6  
3.4 Results and Discussion ................................................................... 6  
3.5 Flowchart ........................................................................................ 7**

**Chapter 4: Engineering Standards and Mapping ................................ 7  
4.1 Impact on Society, Environment and Sustainability ....................... 8  
 4.1.1 Impact on Life ...................................................................... 8  
 4.1.2 Impact on Society & Environment ....................................... 8  
 4.1.3 Ethical Aspects ..................................................................... 8  
 4.1.4 Sustainability Plan ............................................................... 8  
4.2 Project Management and Team Work ............................................ 8  
4.3 Complex Engineering Problem ...................................................... 8  
 4.3.1 Mapping of Program Outcome ........................................... 8  
 4.3.2 Complex Problem Solving ................................................... 8  
 4.3.3 Engineering Activities ......................................................... 8**

**Chapter 5: Conclusion .......................................................................... 9  
5.1 Summary ........................................................................................ 9  
5.2 Limitation ....................................................................................... 9  
5.3 Future Work ................................................................................... 9**

**Chapter 1: Introduction**

**1.1 Introduction**

This project aims to design and implement a web-based platform that acts as a community notice board while also serving as a social space for users to share images, thoughts, and engage with peers. The platform, named **CommunityBoard**, combines functionality and social interaction to foster communication and connectivity within communities.

**1.2 Motivation**

In the digital age, communities often lack a centralized platform to disseminate information and build social engagement simultaneously. Traditional notice boards are static and limited in reach, while most social platforms are too broad for localized community interaction. This project addresses this gap by providing a dedicated, easy-to-use digital hub.

**1.3 Objectives**

* Centralize community updates and announcements.
* Foster peer-to-peer social interactions.
* Allow users to post and share multimedia content.
* Provide secure authentication and privacy controls.
* Ensure accessibility, scalability, and ease of use.

**1.4 Key Features**

* **Notice Board Module**: Users can create, view, and categorize notices for public communication.
* **Social Feed**: A timeline for community members to post updates, share thoughts, and engage with each other.
* **Media Upload**: Support for image and file uploads to enrich posts and announcements.
* **User Authentication**: Secure login options using JWT, OAuth2, and Email OTP.
* **Role Management**: Differentiated access and privileges for admins and regular users.
* **Responsive Design**: Tailored for use on both desktop and mobile devices.
* **Search and Filter Options**: Allows users to find specific posts or announcements quickly.
* **Direct Messaging**: Enables private communication between community members.

**1.5 Feasibility Study**

* **Technical Feasibility**: The selected tech stack (Tailwind, Node.js, Express, MongoDB) is lightweight and scalable.
* **Economic Feasibility**: The platform can be hosted on budget-friendly cloud solutions with open-source technologies reducing costs.
* **Operational Feasibility**: Designed to be user-friendly with low learning curve, and manageable with admin controls.

**1.6 Gap Analysis**

Current platforms lack the integration of formal communication tools (like notices and alerts) with informal community engagement features (such as image sharing and comments). This project bridges this gap by combining both in a single system.

**1.7 Project Outcome**

A fully functional web application where users can view public notices, interact socially, upload content, and maintain community dialogue in a secure and organized environment.

**Chapter 2: Proposed Methodology/Architecture**

**2.1 Requirement Analysis & Design Specification**

**2.1.1 Overview**

This system supports content creation, dissemination, and interaction. It features user authentication, categorized notices, a social feed, media galleries, and admin tools.

**2.1.2 Proposed Methodology/System Design**

* **Frontend**: Developed with Tailwind CSS for responsive design.
* **Backend**: Built on Express.js and Node.js for handling APIs and logic.
* **Database**: MongoDB used for flexible document storage.
* **Authentication**: JWT, OAuth2, and Email OTP implemented for secure access.

**2.2 UI Design**

The UI emphasizes clarity and ease of navigation, featuring a dashboard, side navigation bar, notice board feed, and post creation tools. Design includes mobile responsiveness.

**2.3 Overall Project Plan**

* **Phase 1**: Requirement gathering and design
* **Phase 2**: Development of backend and frontend
* **Phase 3**: Integration and testing
* **Phase 4**: Deployment and feedback collection

**Chapter 3: Implementation and Results**

**3.1 Implementation**

The project was implemented using:

* **Frontend**: Tailwind CSS
* **Backend**: Express.js (Node.js runtime)
* **Database**: MongoDB
* **Authentication**: JWT, OAuth2, Email OTP Features include role-based access, post management, notifications, and direct messaging.

**3.2 User & Profile API Endpoints**

1. **GET /**  
   **Job**: Renders the index view (home page).  
   **Description**: This is the landing page for the application, serving as the main entry point.
2. **GET /read**  
   **Job**: Retrieves and displays all users from the database.  
   **Description**: Fetches all users from the userModel and passes them to the read view for rendering.  
   **Functionality**:

* Retrieves the list of users using await userModel.find().
* Renders the read view with the fetched users.

1. **GET /delete/:id**  
   **Job**: Deletes a specific user based on the user ID.  
   **Description**: This route deletes a user from the database based on the ID passed in the URL.  
   **Functionality**:

* Finds and deletes the user with the matching ID using await userModel.findOneAndDelete({ \_id: req.params.id }).
* Redirects to /read to refresh and show the updated list of users.

1. **GET /edit/:userid**  
   **Job**: Retrieves and displays the current user information to be edited.  
   **Description**: This route fetches the user’s data for editing purposes and renders it in the edit view.  
   **Functionality**:

* Finds a single user based on the userid parameter using await userModel.findOne({ \_id: req.params.userid }).
* Renders the edit view with the user data.

1. **POST /update/:userid**  
   **Job**: Updates the specific user’s data in the database.  
   **Description**: This route allows the editing of a user's information. It updates the user data based on the userid parameter.  
   **Functionality**:

* Extracts the name, email, and image fields from the request body.
* Updates the user record in the database with the new data using await userModel.findOneAndUpdate({ \_id: req.params.userid }, { name, email, image }, { new: true }).
* Redirects to /read to view the updated list of users.

1. **POST /create**  
   **Job**: Creates a new user and saves it to the database.  
   **Description**: This route is for adding a new user to the database.  
   **Functionality**:

* Extracts name, email, and image from the request body.
* Creates a new user in the database using await userModel.create({ name, email, image }).
* Redirects to /read to show the newly added user in the list.

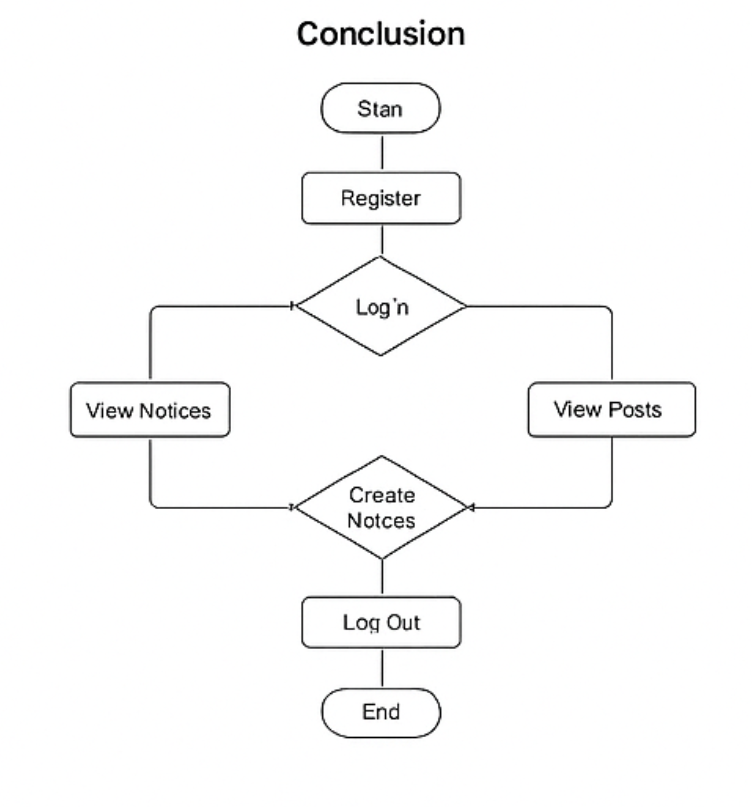
**3.3 Performance Analysis**

* **Load Testing**: Application handles concurrent users efficiently.
* **Response Time**: Average response time below 200ms for standard API calls.
* **Scalability**: MongoDB and Node.js allow horizontal scaling.

**3.4 Results and Discussion**

The system met functional requirements and user feedback indicated high usability and relevance. Some limitations were identified in real-time features and mobile optimization.

**3.5 Flowchart**

****

**Chapter 4: Engineering Standards and Mapping**

**4.1 Impact on Society, Environment and Sustainability**

**4.1.1 Impact on Life**

Encourages local participation, awareness, and expression, improving community cohesion.

**4.1.2 Impact on Society & Environment**

Reduces paper use and promotes eco-friendly communication. Builds a platform for digital inclusion.

**4.1.3 Ethical Aspects**

Ensures data privacy and user consent. Community guidelines implemented for respectful interaction.

**4.1.4 Sustainability Plan**

Codebase designed for long-term maintenance with modular components. Community-led moderation model supports ongoing use.

**4.2 Project Management and Team Work**

Work divided among frontend, backend, database, and testing teams. Agile methodology used for iterative development.

**4.3 Complex Engineering Problem**

**4.3.1 Mapping of Program Outcome**

Aligns with software engineering outcomes such as problem analysis, system design, and ethical responsibility.

**4.3.2 Complex Problem Solving**

Handled complexities in real-time updates, secure authentication, and database schema design.

**4.3.3 Engineering Activities**

Included system modeling, software development, user experience design, and performance evaluation.

**Chapter 5: Conclusion**

**5.1 Summary**

This project developed a platform that integrates structured announcements with community-driven content sharing. It is functional, scalable, and has meaningful social impact.

**5.2 Limitation**

* Limited mobile optimization
* Basic real-time capabilities (no WebSockets)
* Requires moderation policies to prevent misuse

**5.3 Future Work**

* Integrate push notifications and real-time updates
* Enhance mobile UI/UX
* Implement AI-based moderation tools and content tagging
* Offer analytics dashboard for community insights