A

PROJECT

REPORT

ON

Chatting Application

BCA SEM – 6

Developed by

Harsh Tilala

Submitted To

Geetanjali College of Computer Science and Commerce (BCA),
Indian Red Cross Building, Opp. Shastra Maidan, Suchak Road,
Rajkot – 360002.

Saurashtra University Rajkot

Year:2024-2025

Project Guide

ABC

Acknowledgement

- First of all we are sincerely thanked to Saurashtra University for giving a project as a particular subject.
- I am also greatly grateful to our Geetanjali Group of colleges.
- I am also greatly grateful to our college. We would like to thanks Prof. Brijesh Shah (HOD) for providing us opportunity to prepare this project.
- A special thanks to online learning platforms, documentation resources, and open-source communities that helped us understand the required technologies.
- This project has strengthened our understanding of database management, real-time communication
- We acknowledge the role of Flask, Socket.IO, and MySQL in making real-time communication and data storage efficient for this project.

Your Faith Fully,

Harsh Tilala

Perface

In today's digital age, instant messaging applications are crucial for seamless communication. This report highlights the development of a Chat Application that enables real-time messaging with secure user authentication and efficient data management. Built using Flask, Flask-SocketIO, SQLAlchemy, and MySQL, the application ensures smooth communication and message security.

We are proud to present this report on the project titled "Chatting Application developed at the Geetanjali Group of Colleges, Saurashtra University, as part of our 6th-semester curriculum. The project addresses the growing need for efficient cybercrime reporting and management using modern web technologies.

I sincerely thank my faculty members, project guide, and peers for their continuous support and guidance. Their encouragement has been instrumental in achieving the project's goals.

This report serves as a valuable reference for those interested in real-time chat applications and web-based messaging systems. I hope it inspires further exploration in communication technology.

We have tried our best of the practical study and presentation of this report.

Index

Project Profile

- **Project Title** : Chatting Application
- **Development Software** : Visual Studio
- **Front End** : HTML/CSS/JavaScript
- **Back End** : Python (Flask Framework)
- **Database** : MySQL
- **Academic Year** : 2025
- **Developed By** : Harsh Tilala
- **Submitted To** : Geetanjali Groups of College
- **Operating System** : Windows 7, 10, 11

Technology Requirement

Minimum Hardware Requirement

- **Processor** : 2core, 1.2 GHz or higher.
- **RAM** : 1 GB
- **Hard Disk** : 2 GB

Minimum Software Requirement

- **Editing Too** : Visual Studio
- **Browser** : Google Chrome
- **Operating System** : Windows 7, 10, 11
- **Viewer** : Chrome
- **Local Server** : Xampp, Laragon

OverView

- This project is a real-time chat application designed to enable users to communicate instantly.
- It provides a user-friendly interface where users can send and receive messages without delays.
- The system ensures secure authentication, real-time message delivery, and efficient data storage.
- Technologies such as Flask, MySQL, and Socket.IO are used to handle backend processing and real-time communication.
- The application includes features like user login, friend requests, chat history, and online/offline status tracking**.**

Objective

- To Develop a real-time chat system using Flask and Socket.IO for instant messaging.
- To implement a simple yet effective user authentication system.
- Enable friend requests and user connections for private messaging.
- Store and manage chat history efficiently using MySQL
- To ensure an easy-to-use and visually appealing user interface with responsive design.
- Provide AJAX-based real-time updates for smooth navigation without page reloads.

About the Tool (Tools & Technology used)

- **VS Code :** VS code is a lightweight, cross-platform code editor by Microsoft, offering features like IntelliSense, Git integration, debugging, and extensive customization through a vast extension marketplace, making it a powerful tool for developers across programming languages and platforms.
- **Python** - Used for backend development with **Flask**, handling user authentication, session management, database interactions, and server-side logic. It also integrates **Flask-Socket.IO** for real-time messaging.
- **JavaScript** - Enables **real-time chat functionality** using **Socket.IO**, handling message sending, receiving, and updating user statuses dynamically. It also manages AJAX calls for seamless data fetching and UI updates without reloading the page.
- **HTML** - Structures the web pages, including login, registration, chat dashboard, and user profiles. It defines elements like buttons, forms, and message display areas, ensuring a proper user interface layout.

- **CSS** - Styles the application using **Bootstrap** for responsiveness and custom animations for a modern UI. It enhances the visual appeal with gradient effects, dynamic elements, and interactive components for a smooth user experience.
- **SQL (MySQL)** - Manages the database using **Flask-SQLAlchemy ORM**, storing user data, messages, and friendships. It supports user authentication, chat history retrieval, and real-time updates, ensuring efficient data management
- **Laragon** : Laragon is a lightweight, portable local development environment for building web applications with support for multiple languages and databases. It offers speed, isolation, and easy setup, making it ideal for PHP, Node.js, Python and more.
- **Socket.IO** : Socket.IO is a JavaScript library enabling real-time, bidirectional communication between clients and servers using WebSockets to use without refreshing whole page.

Features

- **User Registration and Login**: Secure authentication system with password encryption.
- **Friend Requests**: Users can send, accept, or decline friend requests to connect with others.
- **Real-Time Messaging**: Instant text communication using WebSockets and Socket.IO.
- **Chat History**: Messages are stored in a database for future reference.
- **Online/Offline Status**: Users can see the availability of their friends in real time.
- **Secure Data Storage**: MySQL database ensures structured and efficient data management.
- **Responsive UI**: Designed using Bootstrap and CSS for better user experience

Database Structure

User:

User Table – Stores user information like id, username, password, status, and last_seen to manage authentication and activity tracking.

No	Field Name	Data Type	Size
1	id	INT	11
2	username	VARCHAR	50
3	password	VARCHAR	255
4	status	VARCHAR	20
5	last_seen	DATETIME	-

Message :

Message Table – Saves chat messages with id, from_user, to_user, message, timestamp, and is_read for messaging between users.

1	**id**	**INT**	**11**
2	from_user	INT	11
3	to_user	INT	11
4	message	TEXT	-
5	timestamp	DATETIME	-
6	is_read	BOOLEAN	-

Friendship

Friendship Table – Maintains accepted friendships with id, user1_id, user2_id, and timestamp to track user relationships in the system.

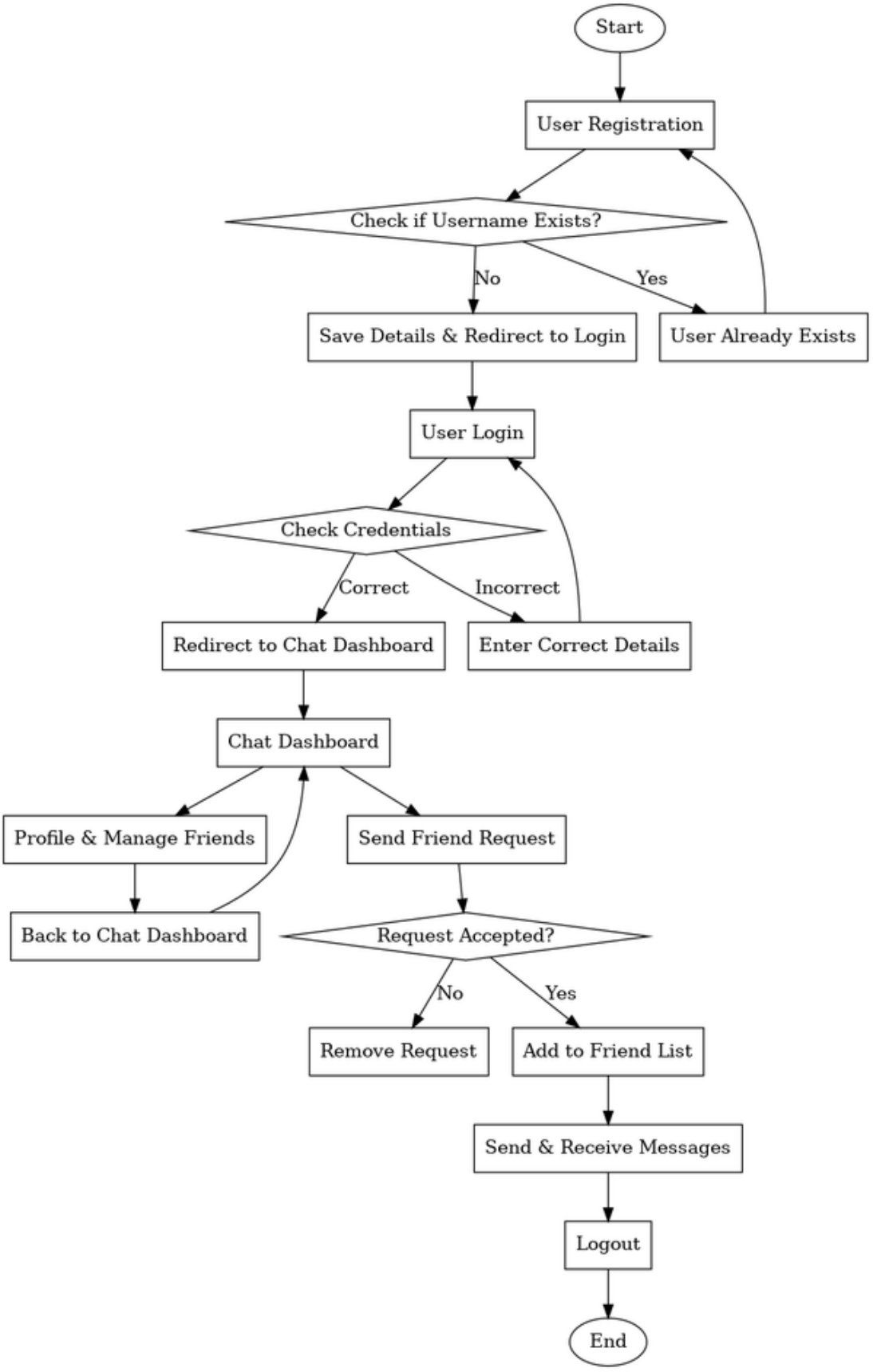
1	id	INT	11
2	user1_id	INT	11
3	user2_id	INT	11
4	timestamp	DATETIME	-

Friend Request

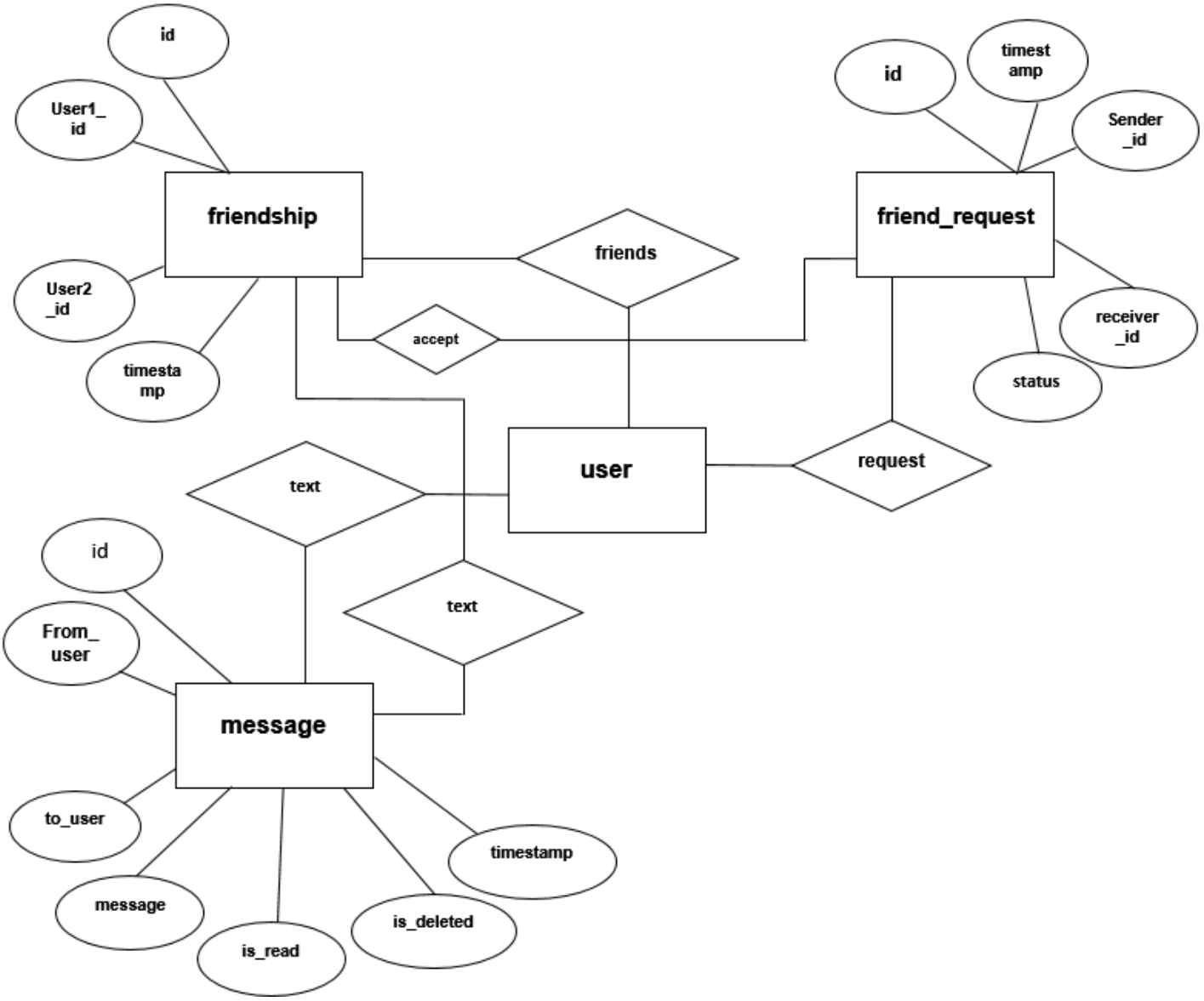
Friend Request Table – Handles friend connections with id, sender_id, receiver_id, status, and timestamp to manage pending and accepted requests.

1	id	INT	11
2	sender_id	INT	11
3	receiver_id	INT	11
4	status	ENUM	-
5	timestamp	DATETIME	-

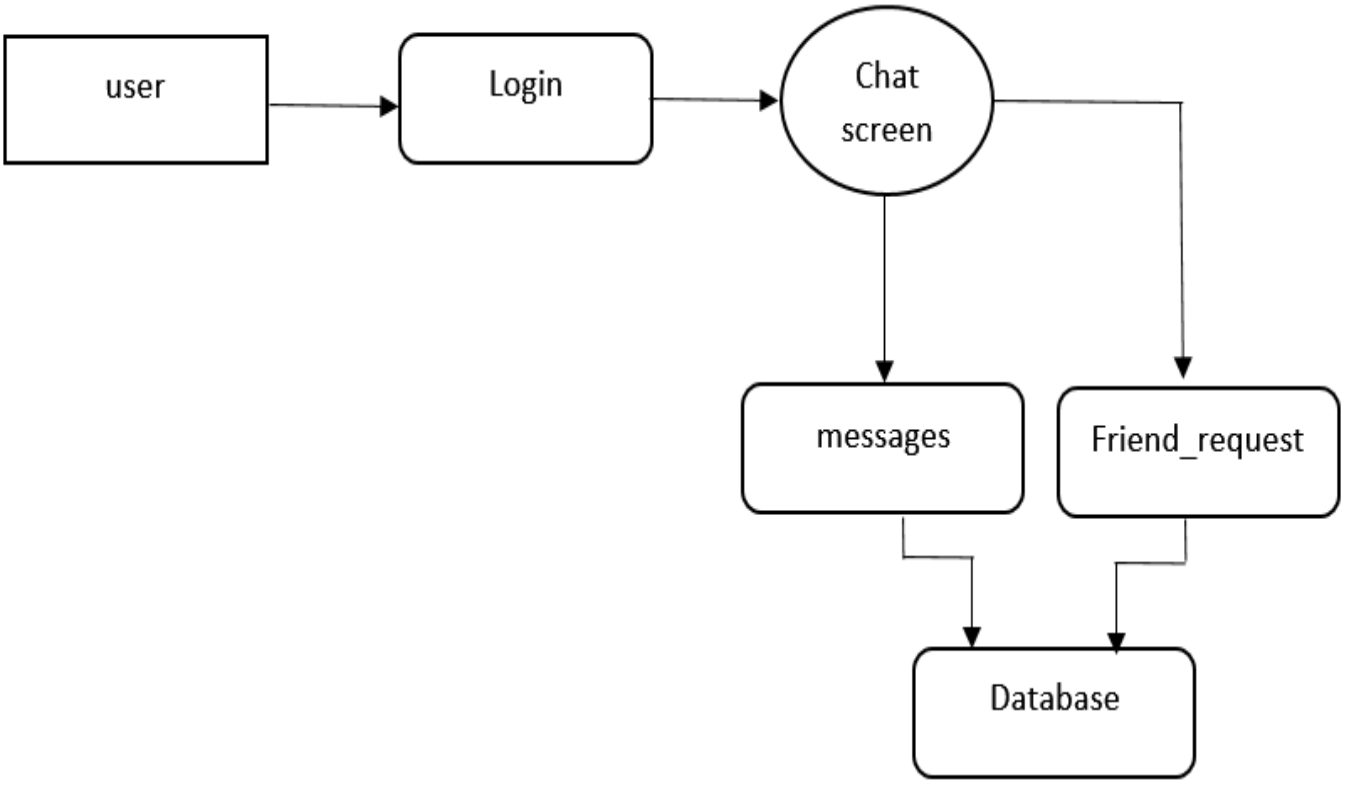
Flow Chart


Start
User Registration
Check if Username Exists?
No
Yes
Save Details & Redirect to Login
User Already Exists
User Login
Check Credentials
Correct
Incorrect
Redirect to Chat Dashboard
Enter Correct Details
Chat Dashboard
Profile & Manage Friends
Send Friend Request
Back to Chat Dashboard
Request Accepted?
No
Yes
Remove Request
Add to Friend List
Send & Receive Messages
Logout
End

E-R Diagram

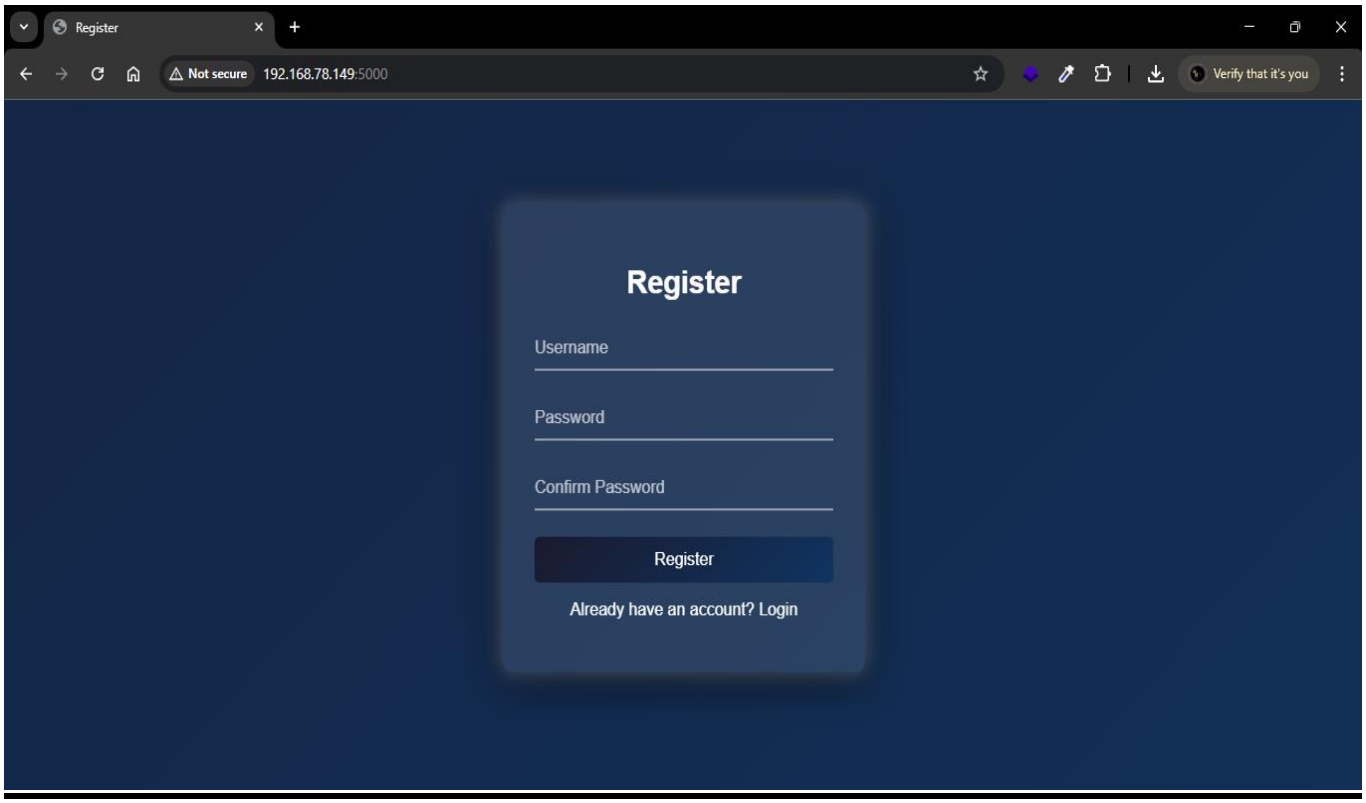

id
User1_
id
friendship
User2
_id
timesta
mp
friends
accept
id
timest
amp
Sender
_id
friend_request
receiver
_id
status
user
request
text
text
id
From_
user
message
to_user
message
is_read
is_deleted
timestamp

DFD

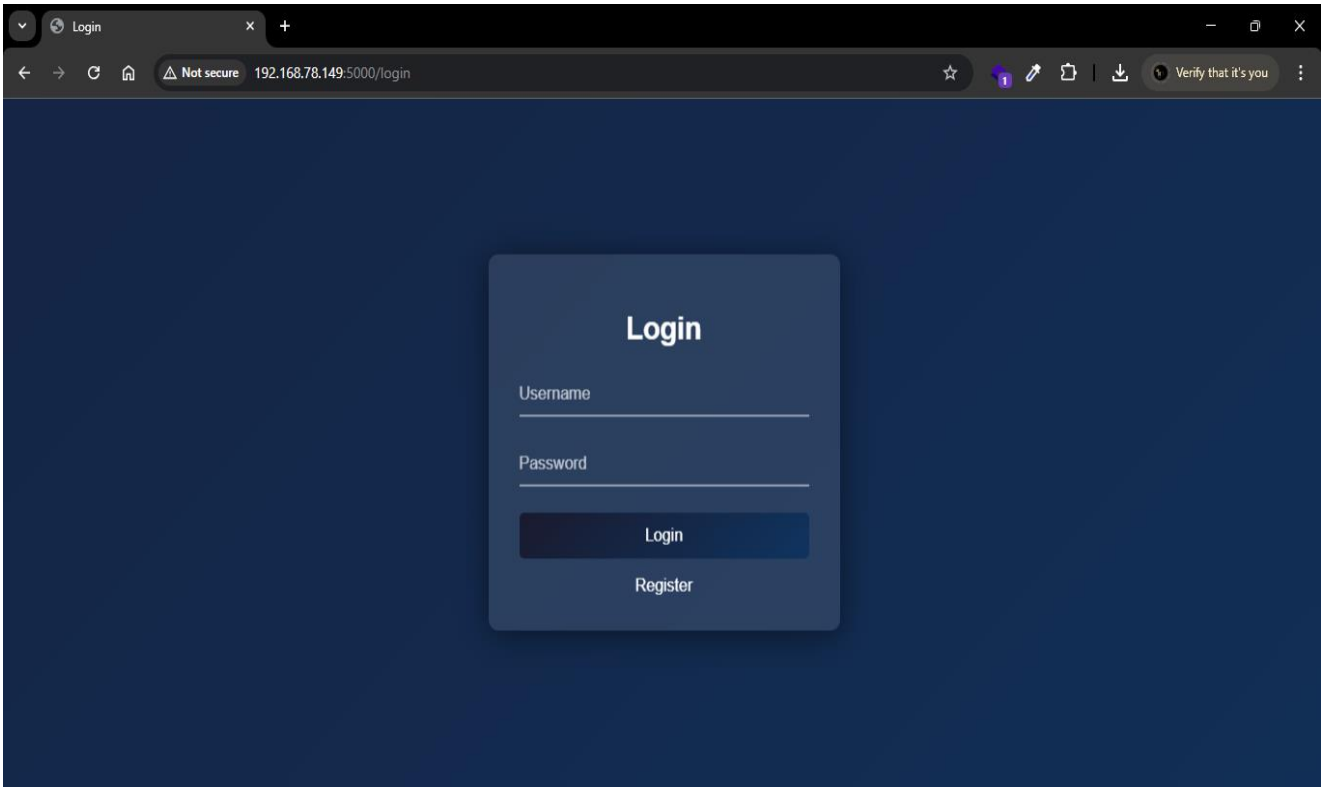


Screen Shots

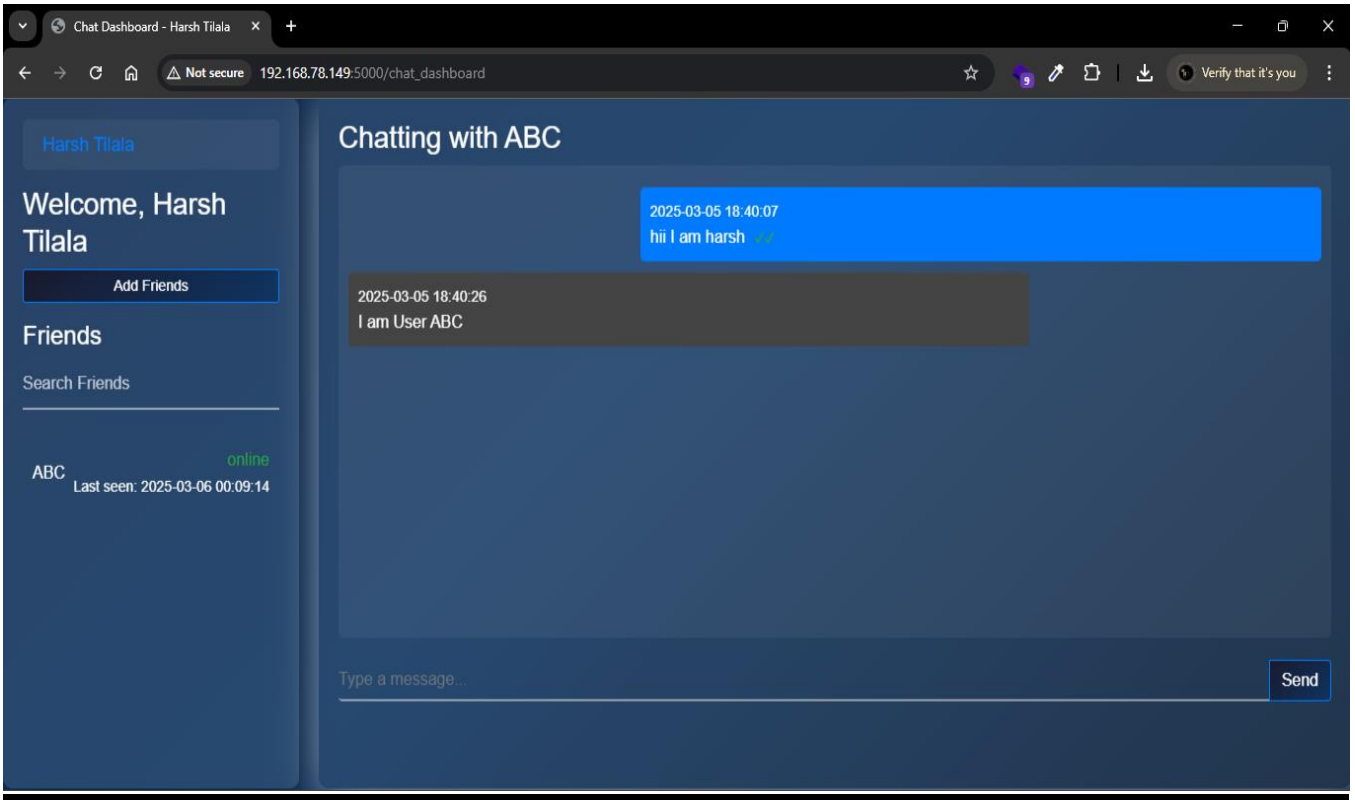
Register Page



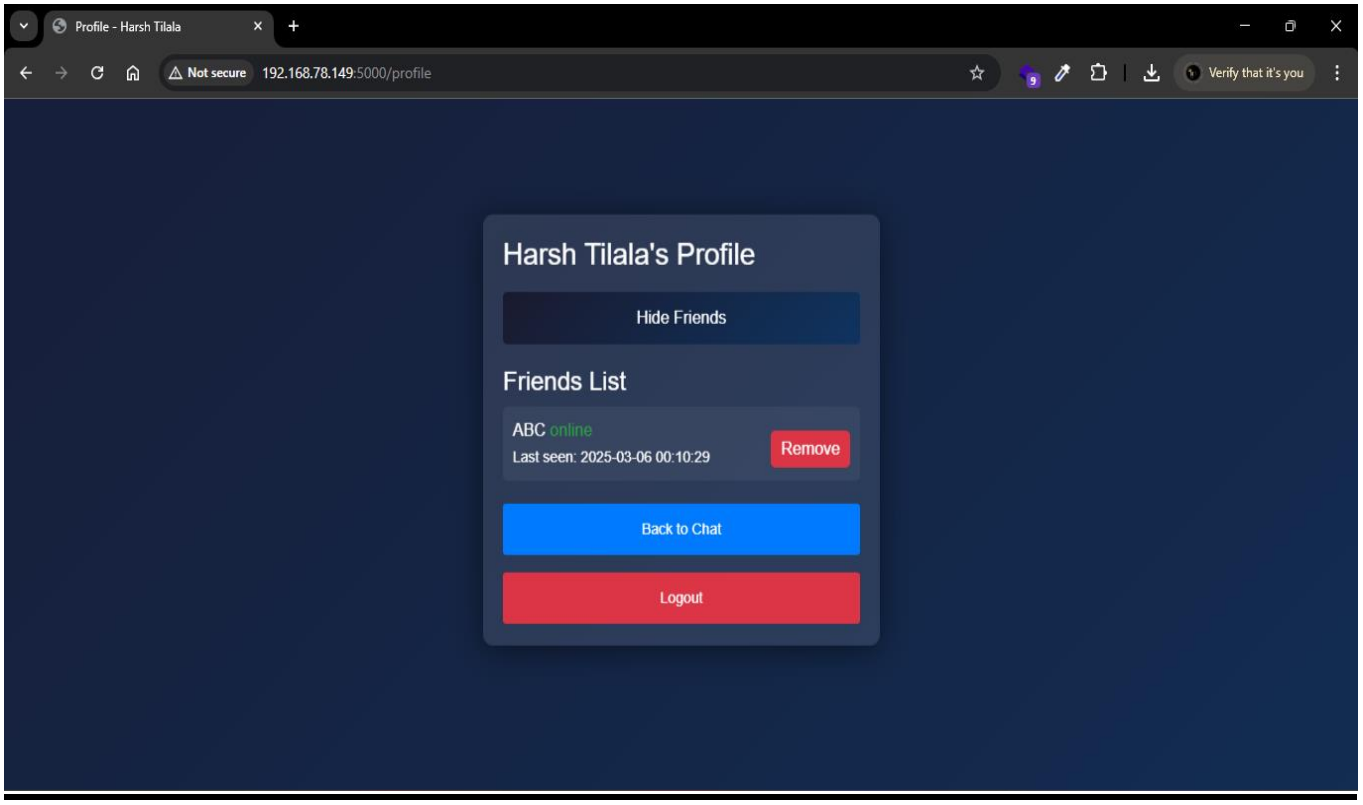
Login Page



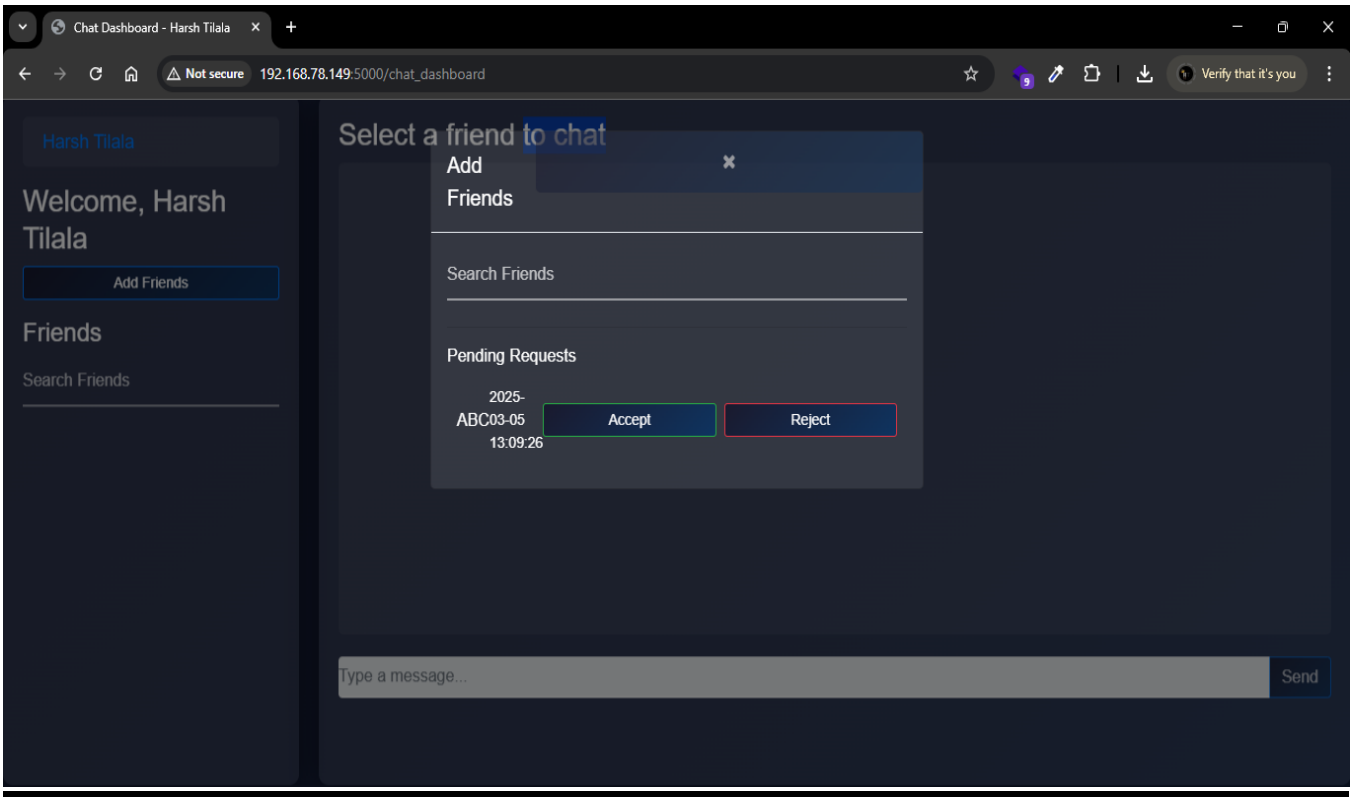
Dashboard Page



Profile Page



Add Friend Page



Conclusion

- This project provided a hands-on learning experience in developing a real-time chat application using Flask, MySQL, and Socket.IO.
- It enhanced our understanding of user authentication, session management, and secure password storage.
- Implementing real-time messaging helped us learn how Web Sockets work and how to optimize performance for live interactions.
- We gained practical experience in frontend development using HTML, CSS, Bootstrap, and JavaScript.
- The use of MySQL as a relational database improved our skills in designing and managing structured data.
- Security implementations, such as encrypted password storage and session handling, helped us understand how to protect user data.
- We learned how to integrate different technologies efficiently and manage dependencies in a full-stack web application.
- Overall, this project was a valuable learning experience that has strengthened our ability to develop scalable, secure, and efficient web applications.

Future Enhancements

- **Group Chat Support:** Allowing multiple users to chat in a group.
- **End-to-End Encryption:** Ensuring private and secure communication.
- **Multimedia Sharing:** Enabling users to send images, videos, and documents.
- **Voice & Video Calls:** Expanding communication options beyond text messages.

Limitations

- No Group Chat Support
- No Multimedia Sharing
- No Message Editing
- No Typing Indicator and Reaction

Bibliography

- www.google.com
- www.chatgpt.com
- www.geeksforgeeks.org