**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.

**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

# Index

|  |  |
| --- | --- |
| **NO** | **NAME** |
| 1 | Project Profile |
| 2 | Project Requirement |
| 3 | Technology Requirement |
| 4 | Data Flow Diagram |
| 5 | E-R Diagram |
| 6 | Data Dictionary |
| 7 | Screenshots |

**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 11

# Technology Requirement

**Minimum Hardware Requirement**

* **Processor :** 4core, 2.2 GHz or higher.
* **RAM :** 8 GB
* **Hard Disk** : 256 GB

**Minimum Software Requirement**

* + **Editing Too :** Visual Studio
* **Browser :** Google Chrome
* **Operating System :** Windows 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 an interactive and secure chat application for real-time communication.
* To implement a simple yet effective user authentication system.
* To enable instant message delivery using Web Sockets and real-time database updates.
* To provide a structured and scalable database for storing user information and chat history.

# To ensure an easy-to-use and visually appealing user interface with responsive design.

# 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:

|  |  |  |  |
| --- | --- | --- | --- |
| 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 :**

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | id | INT | 11 |
| 2 | from\_user | INT | 11 |
| 3 | to\_user | INT | 11 |
| 4 | message | TEXT | - |
| 5 | timestamp | DATETIME | - |
| 6 | is\_read | BOOLEAN | - |

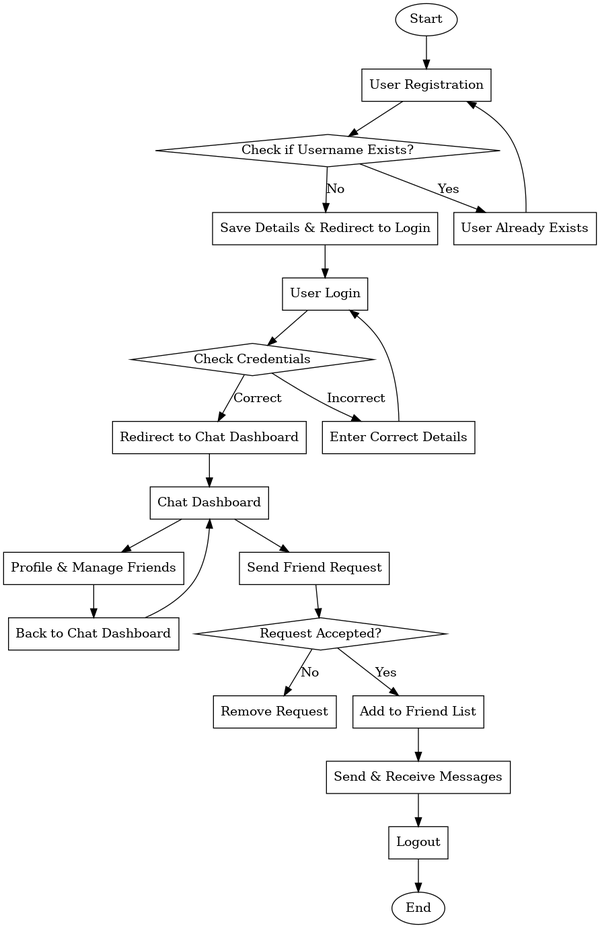
**Follows**

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | id | INT | 11 |
| 2 | user1\_id | INT | 11 |
| 3 | user2\_id | INT | 11 |
| 4 | timestamp | DATETIME | - |

**Message Requests**

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | id | INT | 11 |
| 2 | sender\_id | INT | 11 |
| 3 | receiver\_id | INT | 11 |
| 4 | status | ENUM | - |
| 5 | timestamp | DATETIME | - |

**Flow Chat**



**E-R Diagram**

**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.

# Flow chart

# Data Flow Diagram

DFD Level ( 0 )

Login

Login Update / Delete

Or

Signup

User Register

Admin Register

Admin

Add

Delete

Update

DFD Level ( 1 )

Real-time Chat Application

Registered

User

Login

Chat

Signup

# E-R Diagram

An Entity Relationship Diagram (ERD) is a data modeling technique that graphically illustrates an information system’s entities and the relationship between those entities . An ERD is a conceptual and representation model of data used to represent the entity from work infrastructure .





Kitchenware

Admin

Add

Delete

update

Registered

user

User name

User name

User name

View user list

View Chat

View Chat

View Chat

Chat with Other user

# Data Dictionary

Database Tables:

* User
* Message

**User :**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Null** |
| user\_id | int | primary |
| unique\_id | int(200) | No |
| fname | varchar(255) | No |
| lname | varchar(255) | No |
| email | varchar(255) | No |
| password | varchar(255) | No |
| img | varchar(255) | No |
| status | varchar(400) | No |

**Message :**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Null** |
| msg\_id | int(11) | primary |
| incoming\_msg\_id | int(255) | No |
| outgoing\_msg\_id | int(255) | No |
| msg | varchar(1000) | No |