**Project Presentation of Full Stack Engineering Project (FSE) (22CS037)**

On

# Real Time Chat App

Saikat Hazra (2210992216)
Sachidanand Gupta (2210992202)
Sachin Kumar (2210992204)
Sarbraj Singh(2210992258)

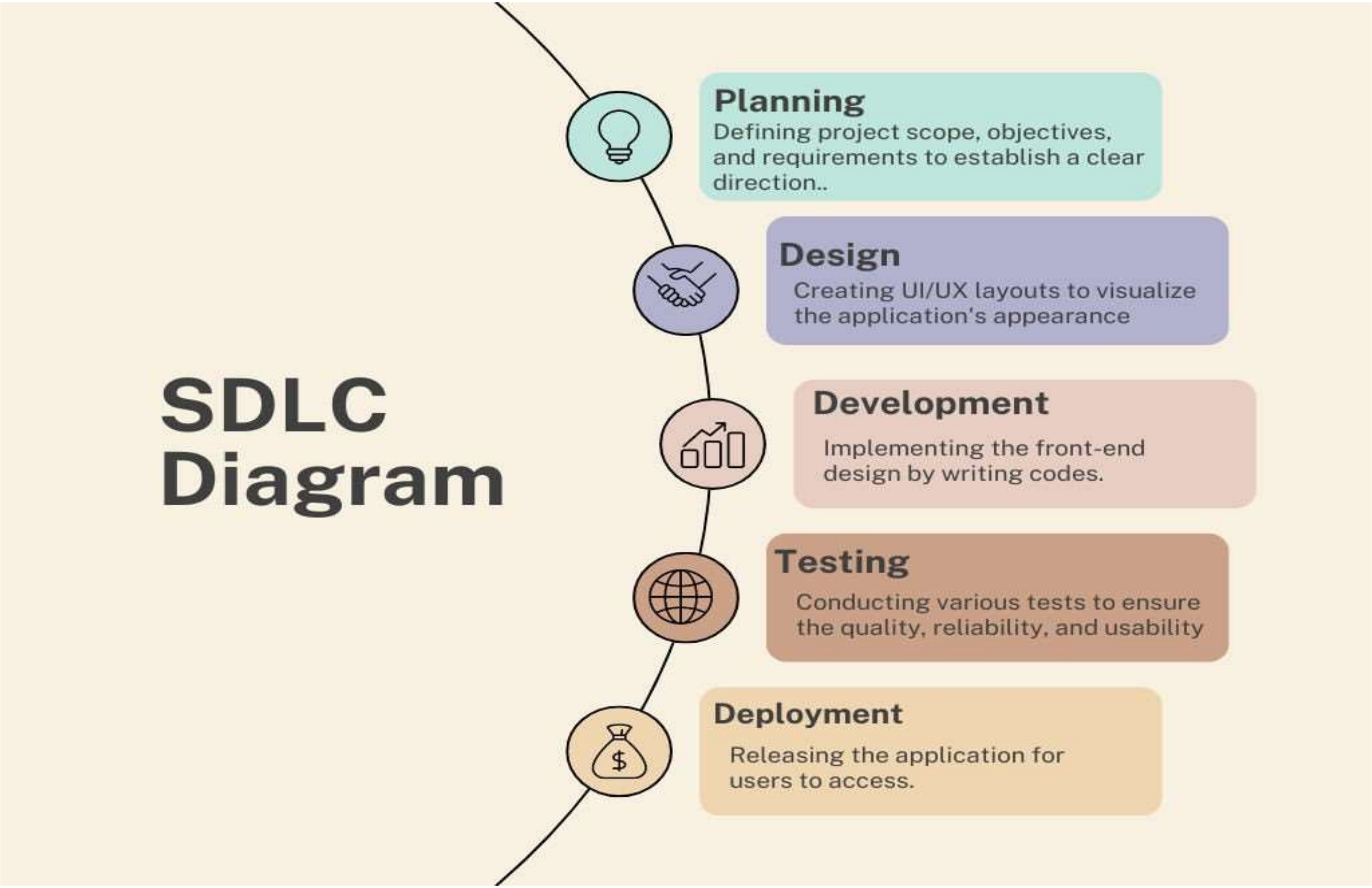
Supervised By
Mr. Rahul

Department of Computer Science and Engineering,
Chitkara University, Punjab

# Introduction

A **Real-Time Chat App** using the **MERN stack (MongoDB, Express.js, React, Node.js)** enables seamless, instant messaging between users. It leverages **WebSockets** (via **Socket.io**) to establish real-time communication. The backend, powered by **Node.js** and **Express**, handles authentication, message storage in **MongoDB**, and WebSocket connections.

# SDLC MODEL



## SDLC Diagram

**Planning**
Defining project scope, objectives, and requirements to establish a clear direction..

**Design**
Creating UI/UX layouts to visualize the application's appearance

**Development**
Implementing the front-end design by writing codes.

**Testing**
Conducting various tests to ensure the quality, reliability, and usability

**Deployment**
Releasing the application for users to access.

# TECHNICAL DETAILS

## HTML:

- Markup language for creating the structure of web pages.
- Utilizes elements like <div>, <p>, <img> to organize content.
- Forms for user input, with various input types like text, password, etc.
- Semantics for accessibility and SEO.

## TAILWIND CSS:

- A utility-first CSS framework for styling HTML elements.
- Simplifies layout design with classes for the box model: content, padding, border, and margin
- Provides utility classes for positioning, display, flexbox, and grid layouts.
- Enables responsive design through media query utilities.

## REACTJS:

- Allows dynamic rendering of HTML content based on real-time data.
- Facilitates reusable components like headers, footers, and forms.
- Supports conditional rendering to tailor content based on user roles or data.
- Seamlessly integrates form handling for actions like login and data submissions.
- Efficiently loops through arrays to render dynamic content such as notices, events, or appointments.

# TECHNICAL DETAILS

CHITKARA UNIVERSITY

## Backend

### MongoDB:

- NoSQL database designed for handling large volumes of data.
  Uses flexible, JSON-like documents to store data.
  Supports dynamic schemas, allowing easy updates to data structure.
- Provides horizontal scaling for managing distributed data.
- Facilitates fast queries and indexing for efficient data retrieval.

### Express:

- Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.
- Express is a lightweight and flexible routing framework with minimal core features meant to be augmented through the use of Express middleware modules.

### Postman:

- Postman is a global software company that offers an API platform for developers to design, build, test, and collaborate on APIs.

# Features

**User Authentication** – Secure login/signup using JWT and bcrypt.
✅ **Real-Time Messaging** – Instant chat using **Socket.io** for real-time communication.
✅ **One-on-One & Group Chats** – Support for personal and group conversations.
✅ **Online/Offline Status** – Show users' availability in real-time.
✅ **Message Read Receipts** – Indicate when messages are delivered and read.

✅ **Typing Indicators** – Display when a user is typing.
✅ **Media & File Sharing** – Send images, videos, and documents.
✅ **Notifications** – Real-time alerts for new messages.
✅ **Search & Filters** – Find users or messages easily.
✅ **Chat History & Storage** – Store messages securely in MongoDB.

# Target Audience

## •Who is this for?

**1. General Users**

•Individuals looking for instant messaging solutions.

**2. Businesses & Enterprises**

•Teams needing real-time collaboration (e.g., Slack alternative).

**3. Developers & Tech Enthusiasts**

•MERN stack developers learning WebSockets & real-time features.

**4. Educational Platforms**

•Online learning portals requiring live discussions.

# User-Friendly Interface

**Drag-and-Drop File Sharing**
Users can easily send images, videos, and documents by dragging and dropping them into the chat.
**Rich Messaging Experience**
Built-in support for emojis, GIFs, reactions, and formatted text to enhance communication.
**Intuitive Navigation**
A simple and well-structured UI with easy access to contacts, recent chats, and settings.
Would you like me to add more features or refine the content further?

# Languages

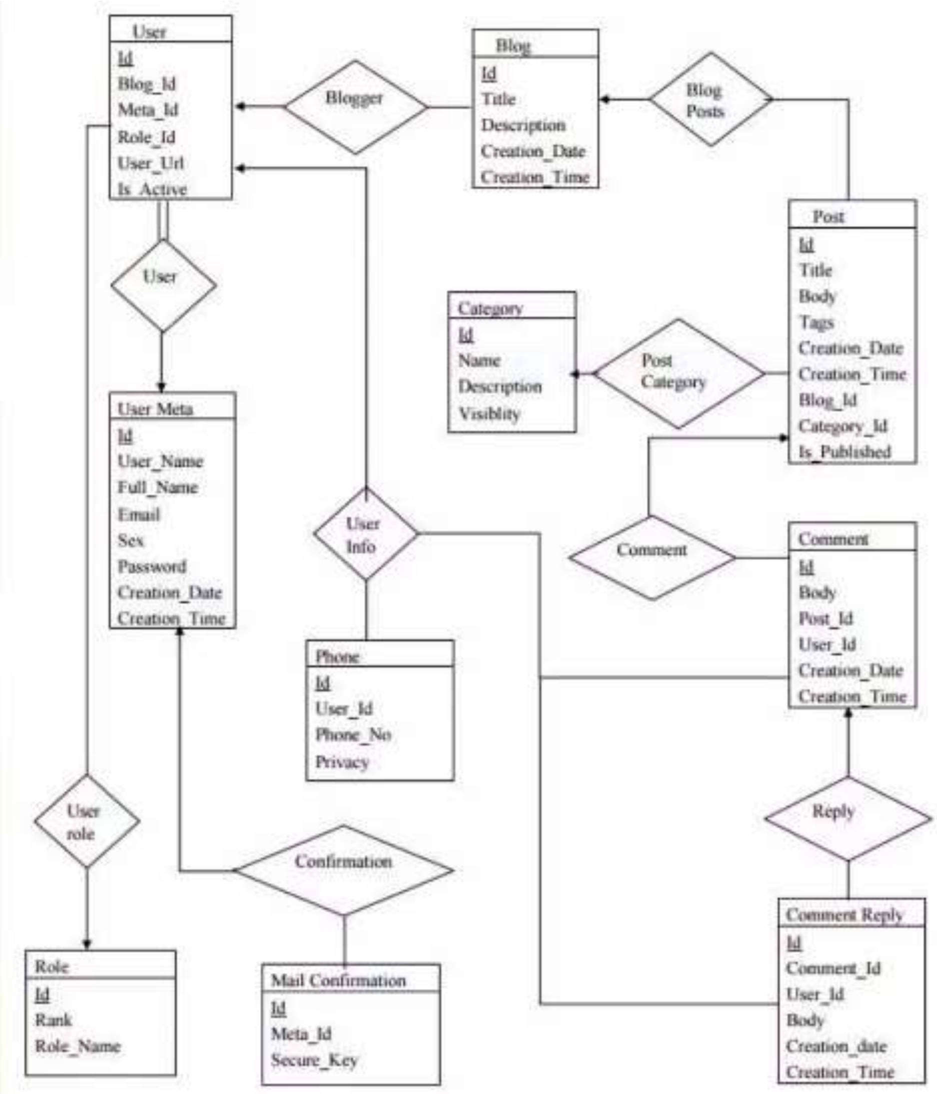
**MongoDB**
A NoSQL database that stores data in flexible, JSON-like documents, perfect for content-rich applications like blogs.

**Express.js**
A minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.

**React**
A powerful JavaScript library for building user interfaces, enabling the creation of dynamic, responsive, and interactive blog experiences.

**Node.js**
A JavaScript runtime built on Chrome's V8 JavaScript engine, allowing server-side scripting and efficient data handling.

# ER Diagram



User
Id
Blog_Id
Meta_Id
Role_Id
User_Url
Is_Active

Blog
Id
Title
Description
Creation_Date
Creation_Time

Post
Id
Title
Body
Tags
Creation_Date
Creation_Time
Blog_Id
Category_Id
Is_Published

Category
Id
Name
Description
Visiblity

User Meta
Id
User_Name
Full_Name
Email
Sex
Password
Creation_Date
Creation_Time

Phone
Id
User_Id
Phone_No
Privacy

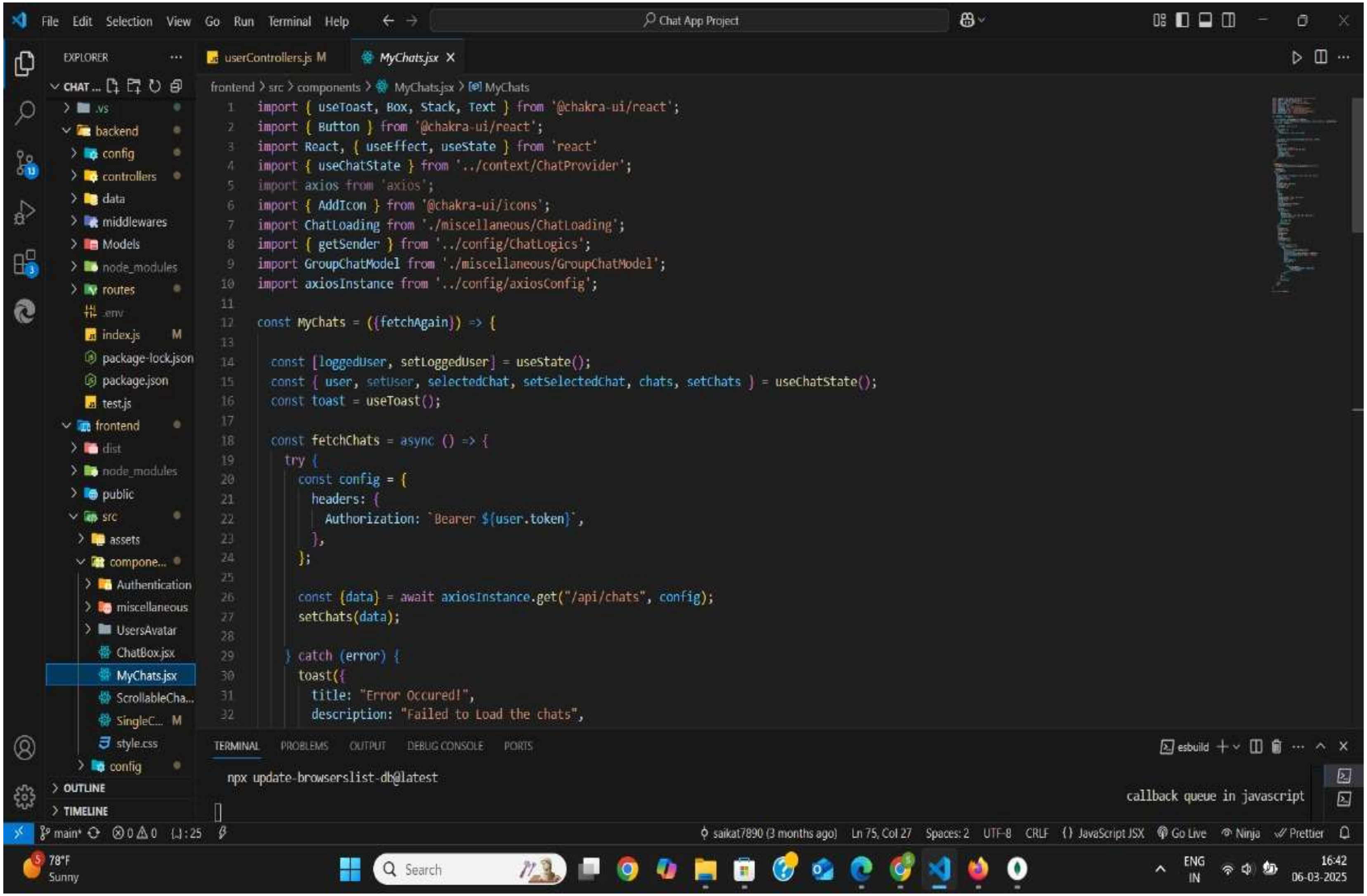
Comment
Id
Body
Post_Id
User_Id
Creation_Date
Creation_Time

Role
Id
Rank
Role_Name

Mail Confirmation
Id
Meta_Id
Secure_Key

Comment Reply
Id
Comment_Id
User_Id
Body
Creation_date
Creation_Time

# Code Snippets



```jsx
import { useToast, Box, Stack, Text } from '@chakra-ui/react';
import { Button } from '@chakra-ui/react';
import React, { useEffect, useState } from 'react'
import { useChatState } from '../context/ChatProvider';
import axios from 'axios';
import { AddIcon } from '@chakra-ui/icons';
import ChatLoading from './miscellaneous/ChatLoading';
import { getSender } from '../config/ChatLogics';
import GroupChatModel from './miscellaneous/GroupChatModel';
import axiosInstance from '../config/axiosConfig';

const MyChats = ({fetchAgain}) => {

  const [loggedUser, setLoggedUser] = useState();
  const { user, setUser, selectedChat, setSelectedChat, chats, setChats } = useChatState();
  const toast = useToast();

  const fetchChats = async () => {
    try {
      const config = {
        headers: {
          Authorization: `Bearer ${user.token}`,
        },
      };

      const {data} = await axiosInstance.get("/api/chats", config);
      setChats(data);

    } catch (error) {
      toast({
        title: "Error Occured!",
        description: "Failed to Load the chats",
```

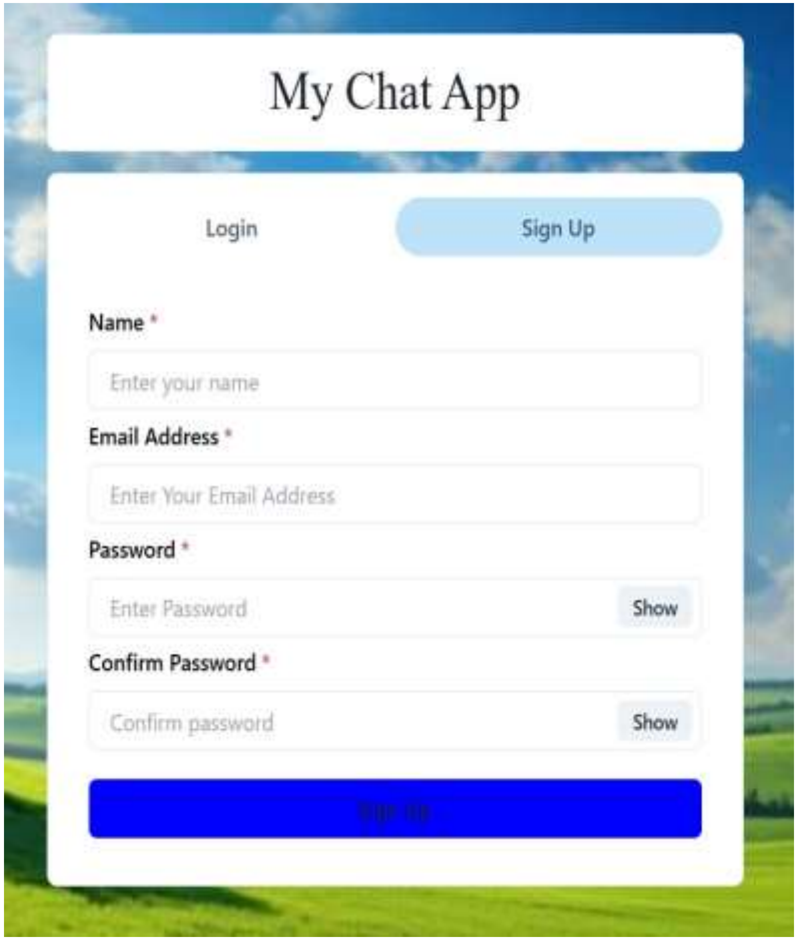
```
npx update-browserslist-db@latest
```

```
callback queue in javascript
```

# Output

CHITKARA UNIVERSITY

## Sign up



My Chat App

Login | Sign Up

Name *

Enter your name

Email Address *

Enter Your Email Address

Password *

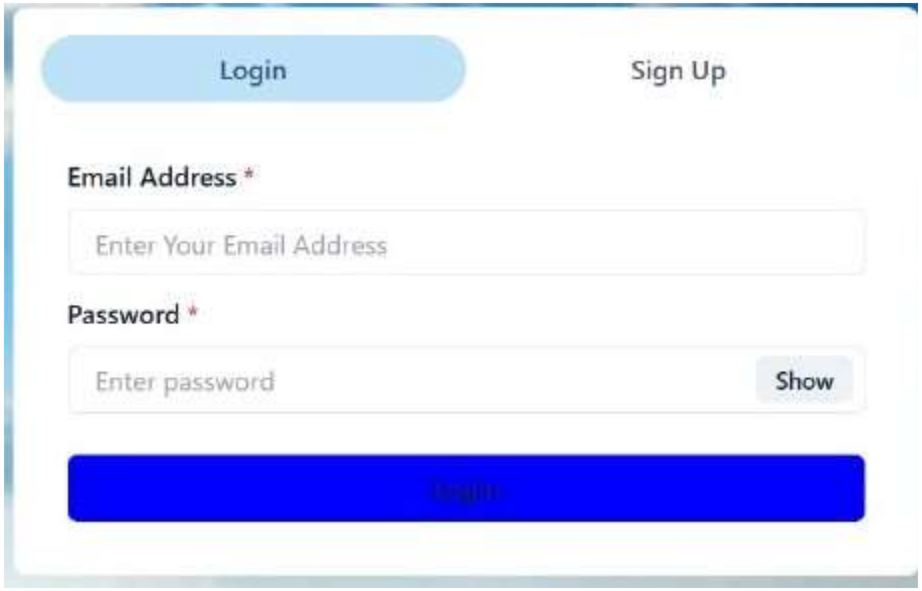
Enter Password Show

Confirm Password *

Confirm password Show

Sign Up

## Login



Login | Sign Up

Email Address *

Enter Your Email Address

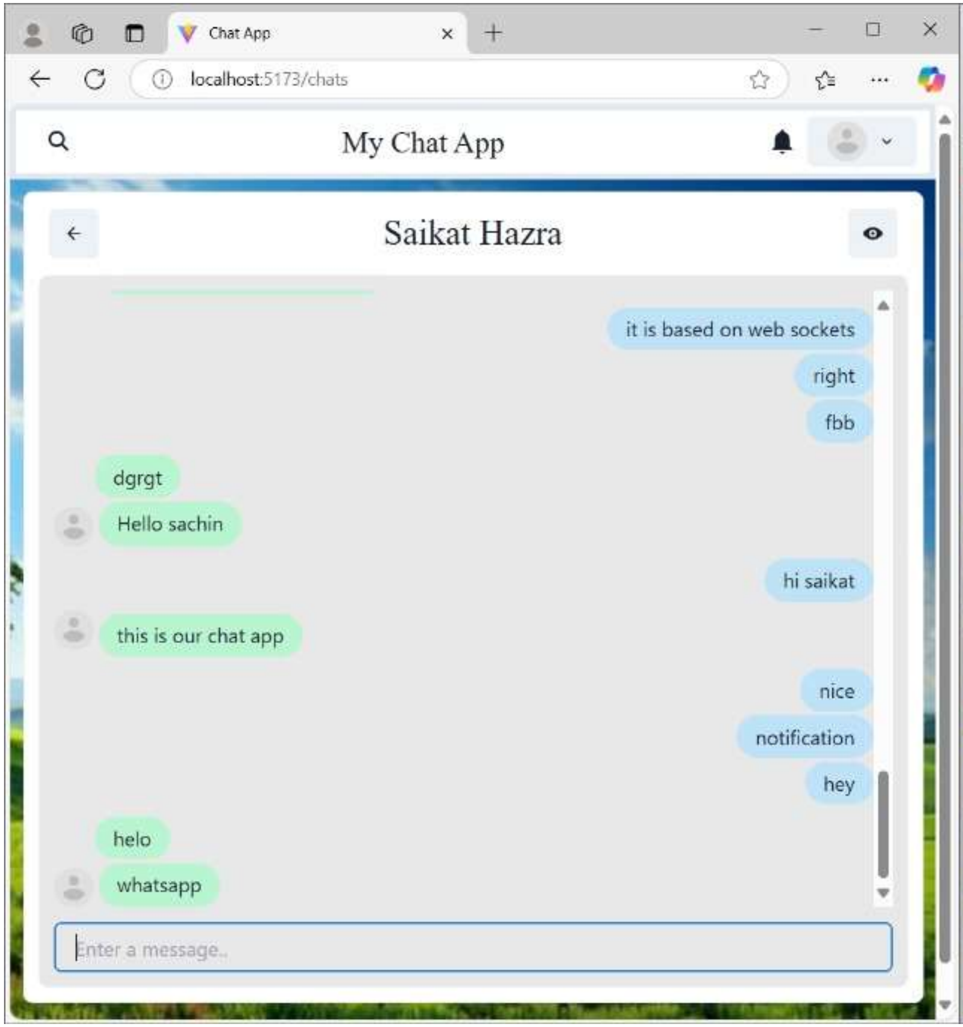
Password *

Enter password Show

Login

# Project Snippet

CHITKARA UNIVERSITY



My Chat App

Saikat Hazra

it is based on web sockets

right

fbb

dgrgt

Hello sachin

hi saikat

this is our chat app

nice

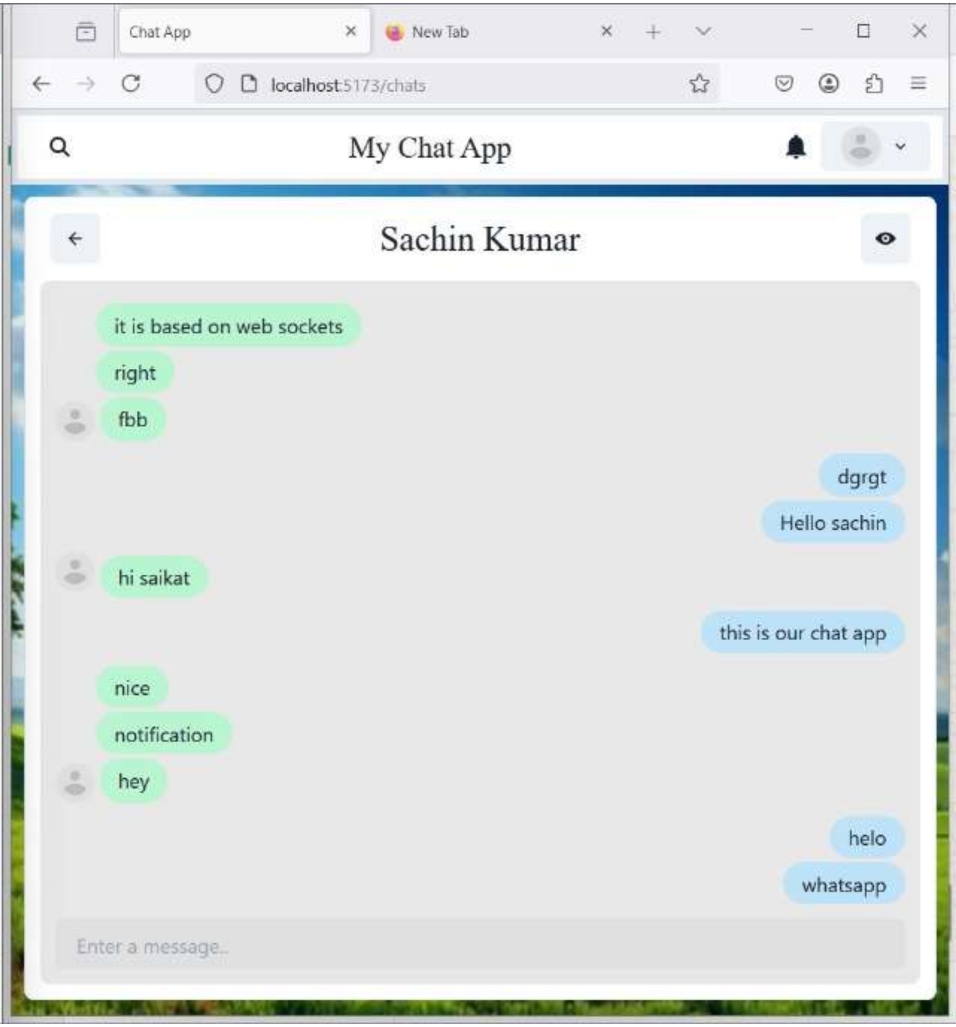
notification

hey

helo

whatsapp

Enter a message..



My Chat App

Sachin Kumar

it is based on web sockets

right

fbb

dgrgt

Hello sachin

hi saikat

this is our chat app

nice

notification

hey

helo

whatsapp

Enter a message..

# CONCLUSION

- The Real-Time Chat Application built using the MERN stack (MongoDB, Express.js, React.js, Node.js) provides a scalable, secure, and efficient communication platform. By integrating Socket.io for real-time messaging, JWT authentication for security, and cloud storage for media handling, the application ensures a smooth and seamless user experience.Following an Agile development methodology, the project is designed for continuous improvement, scalability, and adaptability to future enhancements like voice/video calls, AI chatbots, and multi-device synchronization.

Thank
you!