CONVO CONNECT CHAT APPLICATION

Mini Project - Report

Submitted by

AVANEESH SHARMA	**AVINASH MONTEIRO**
(4NM21CS038)	(4NM21CS040)

6th Semester B.E.

Under the Guidance of

Dr. SARIKA HEGDE

Professor

Ms. JAYAPADMINI KANCHAN

Assistant Professor Gd-II

In partial fulfillment of the requirements for the award of
the Degree of

Bachelor of Engineering in Computer Science and Engineering

from

Visvesvaraya Technological University, Belagavi

Department of Computer Science and Engineering
NMAM Institute of Technology, Nitte - 574110
(An Autonomous Institution affiliated to VTU, Belagavi)

MAY 2024

NITTE
EDUCATION TRUST

N.M.A.M. INSTITUTE OF TECHNOLOGY
(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)
Nitte – 574 110, Karnataka, India

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

CERTIFICATE

Certified that the Mini project work entitled

"Convo Connect Real Time Chat App"

is a bonafide work carried out by

AVANEESH SHARMA (4NM21CS038)

AVINASH MONTEIRO (4NM21CS040)

of 6th Semester B.E. in partial fulfilment of the requirements for the award of ***Bachelor of Engineering Degree in Computer Science and Engineering*** *prescribed by* ***Visvesvaraya Technological University, Belagavi***

during the year ***2023-2024.***

____________________ ____________________

Signature of the Guide ***Signature of the HOD***

Viva Voce Examination

Name of the Examiners **Signature with Date**

1. ____________________________ ____________________________

2. ____________________________ ____________________________

Abstract

This project focuses on the development of a real-time chat web application utilizing the MERN stack—comprising MongoDB, Express.js, React.js, and Node.js—tailored for efficient and interactive communication. React.js enhances user interface design, offering dynamic and adaptive front-end components for an engaging user experience. MongoDB, a flexible NoSQL database, excels in the rapid storage and retrieval of large and diverse data sets, crucial for the real-time aspect of the chat application. The server-side logic is orchestrated by Node.js, with Express.js facilitating the efficient handling of APIs and backend processes.A standout feature of this application is its advanced media handling capabilities, powered by Cloudinary. This integration allows users to seamlessly send and receive messages, images, audio, video, and other file types in real-time, greatly enriching the communication experience. Cloudinary's robust cloud-based platform supports the efficient upload, optimization, and management of media assets, ensuring high performance and scalability. Together, these technologies create a sophisticated and versatile platform that supports a wide range of communication needs in real-time.

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to all those who have contributed to the development and success of our real-time chat web application.

First and foremost, I wish to express my deep feelings of gratitude to our institution**, Nitte Mahalinga Adyanthaya Memorial Institute of Technology, Nitte** for providing me the opportunity for completing my project successfully

I express my kind thanks to **Dr. Niranjan N Chiplunkar**, Principal and **Dr. I Ramesh Mithanthaya**, Vice Principal, NMAMIT, Nitte for their able support and encouragement.

I express my sincere gratitude to **Dr. Jyothi Shetty**, Head of the Dept., Department of Computer Science and Engineering, NMAMIT, Nitte for their support and guidance.

I take this opportunity to express my gratitude and deep regards to my mini project guides **Dr. Sarika Hegde**, Professor, Department of Computer Science and Engineering and **Ms. Jayapadmini Kanchan**, Assistant professor, Department of Computer Science and Engineering for their exemplary guidance, monitoring, and constant encouragement.

AVANNESH SHARMA
4NM21CS038

AVINASH MONTEIRO
4NM21CS040

Table of Contents

CHAPTER 1

INTRODUCTION

1.1 BRIEF OVERVIEW OF THE PROJECT

Our project is dedicated to developing an innovative real-time chat web application that aims to revolutionize online communication using cutting-edge web technologies. Our primary objective is to create a seamless and efficient platform for instant messaging, while also incorporating additional features to enhance the overall user experience. In addition to basic messaging functionality, users will enjoy features such as profile customization with images, secure user authentication, voice messaging, and the ability to search for contacts and messages.

To achieve this ambitious goal, we are leveraging a combination of modern web technologies. React.js is instrumental in frontend development, ensuring a dynamic and responsive user interface that adapts seamlessly to various devices. MongoDb serves as the foundation for database management, facilitating efficient storage and retrieval of data. For backend development, we rely on Node.js/Express to handle server-side logic and requests. Where MongoDB Studio streamlines database management tasks with its user-friendly interface, enabling easy schema visualization and data manipulation. Additionally, Cloudinary enriches our application with all the features like file sharing image sharing audio sharing and video sharing which we can download in our own system

This deployed web application allows users to log in using their Gmail accounts, enabling easy access for a wide audience. Once logged in, users can engage in personal chats with others who are also using the web app. This feature enhances the user experience by facilitating direct and personalized communication, making it convenient for users to connect with each other in realtime.

In our quest to deliver a comprehensive chat experience, we prioritize user accessibility by ensuring compatibility with mobile devices. Users can enjoy the convenience of instant communication anytime, anywhere. Moreover, features such as searching for contacts and messages enhance usability, allowing users to navigate and manage their conversations effortlessly.

1.2 OBJECTIVES OF THE PROJECT

- Facilitate Instant Communication: Our primary objective is to develop a chat application that enables users to converse in real-time, providing a platform for quick and seamless interaction.
- Leverage Advanced Technologies: Utilizing cutting-edge web technologies such as MongoDb, Express, React.js, Node.js, and Cloudinary, we aim to build a robust and efficient application infrastructure. These technologies empower us to create a smooth and responsive user experience.
- Enhance User Engagement: In addition to basic messaging functionality, our goal is to enrich the user experience with features such as profile customization, secure authentication, file sharing, and audio video sending. These enhancements aim to make the chat experience more enjoyable and engaging for users.
- Ensure Universal Accessibility: We prioritize ensuring compatibility across various devices and platforms, ensuring that users can access the application seamlessly from anywhere. Additionally, implementing secure login mechanisms, such as Gmail authentication, ensures user safety and convenience.
- Successful Deployment: Our ultimate objective is to deploy the application effectively, making it accessible to a wide audience. Through meticulous deployment strategies and ongoing monitoring, we aim to ensure a reliable and user-friendly experience for all users**.**

1.3 IMPORTANCE OF THE PROJECT

The project's significance lies in its ability to transform online communication. By creating a real-time chat platform with advanced features and compatibility across devices, we're offering users a convenient way to connect instantly. This addresses the need for effective communication tools in today's fast-paced world, enhancing user engagement and accessibility. Through the project, we aim to set new standards in online communication, making chatting easy and enjoyable for everyone.

CHAPTER 2

LITERTATURE SURVEY

2.1 BRIEF OVERVIEW OF LITERTATURE SURVEY

1. Introduction to Real-Time Chat Applications:

Real-time chat apps, like WhatsApp and Facebook Messenger, let us talk instantly with friends and colleagues, no matter where they are. They've become a big part of how we communicate every day, both personally and professionally.

2. Technologies and Frameworks:

These apps are made using different tools and technologies. There are frameworks for building the parts you see (like React or Angular), and others for handling the behind-the-scenes stuff (like Node.js or Django). Special messaging protocols (like WebSocket) help messages move quickly between devices.

3. User Experience and Interface Design:

Making chat apps easy to use is super important. Good design means clear layouts, easy navigation, and fun features like emojis and picture sharing. It's also important that the app works well on different devices, like phones and computers.

4. Security and Privacy:

Keeping our messages safe is a big deal. Chat apps use things like encryption to protect our conversations from prying eyes. They also have to follow rules about how they handle our data, like where they store it and who can access it.

5. Social and Psychological Implications:

Using chat apps can change how we interact with each other online. They can make it easier to connect with friends and meet new people. But they can also affect things like our privacy and how we behave online.

6. Comparative Analysis:

People have different preferences when it comes to chat apps. Some like ones with lots of features, while others prefer simpler ones. By comparing different apps, we can see what works well and what could be improved.

CHAPTER 3

PROBLEM STATEMENT

"Design and implement a real-time chat web application enabling users to exchange both text and files sharing includes audio and video sending. The application should support instant text messaging for quick communication, alongside the innovative feature file sharings for a more personalized and expressive interaction. The objective is to enhance user engagement and provide a comprehensive chatting experience tailored to modern communication preferences."

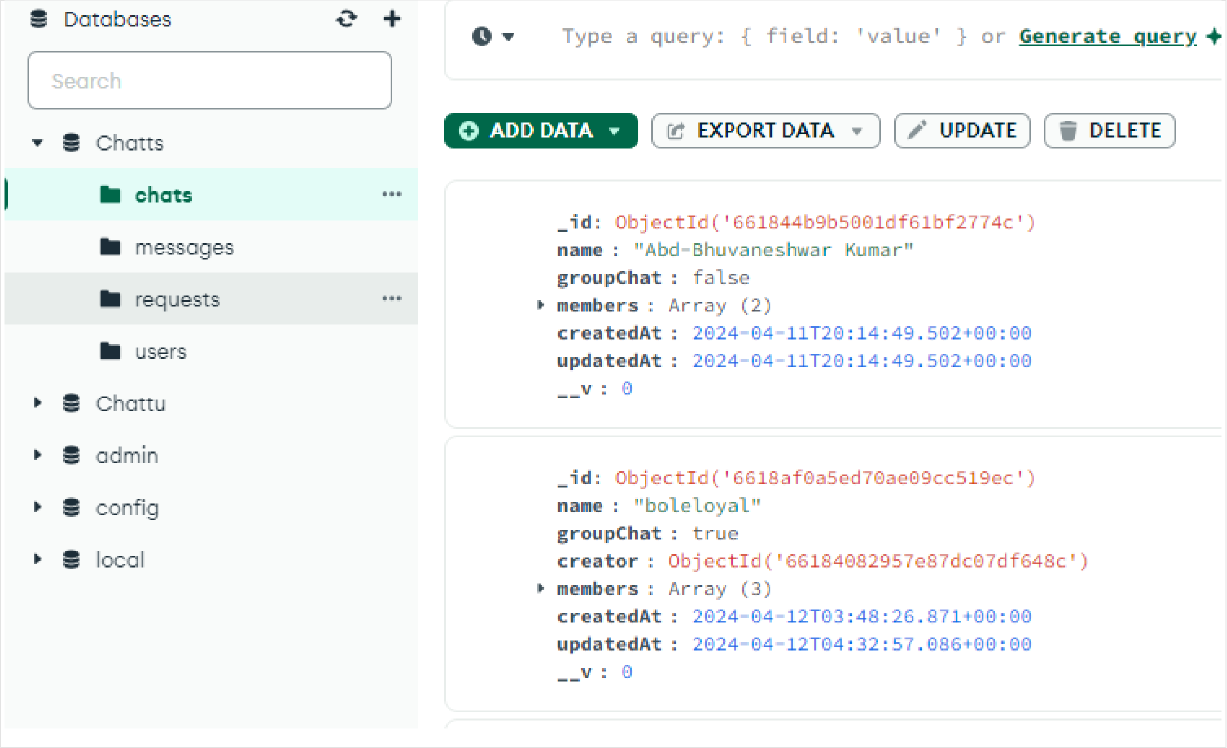
In today's digital age, effective communication is crucial. While text messaging has been a staple, sharing of files which is crucial, allowing users to express themselves more fully. Our project aims to develop a chat app that combines both text and voice messaging features. By offering users the choice between typing or speaking, we aim to make chatting more interactive, catering to different communication styles and preferences.

CHAPTER 4

METHODOLOGY

The development process for the real-time chat application using MongoDb , Express, Node.js, React.js, and Socket.IO begins with a comprehensive analysis of project requirements and objectives. This initial stage involves identifying essential features such as real-time chat functionality, user authentication, profile management, and file sharing. Once the requirements are established, the appropriate technologies are selected to fulfill these needs. MongoDb is chosen as the database management system for storing chat messages and user data, while React.js with Tailwind CSS is utilized for frontend development. Express and Node.js serve as the foundation for backend development The integration of Cloudinary for the file sharing here the file sharing may include files which you can download as a user and send it to tother ..Enabling the audio and video sharing which supports the mp3 and mp4 files and Socket.IO is employed to implement real-time chat functionality, ensuring instant messaging without delays and providing a smooth communication experience for users.

MongoDB: As the database of choice in the MERN stack, MongoDB is a NoSQL database that excels in handling large volumes of unstructured data. It is utilized to store user data, chat messages, and multimedia content, ensuring flexibility and scalability in data management.



```
_id: ObjectId('661844b9b5001df61bf2774c')
name : "Abd-Bhuvaneshwar Kumar"
groupChat : false
members : Array (2)
createdAt : 2024-04-11T20:14:49.502+00:00
updatedAt : 2024-04-11T20:14:49.502+00:00
__v : 0
```

```
_id: ObjectId('6618af0a5ed70ae09cc519ec')
name : "boleloyal"
groupChat : true
creator : ObjectId('66184082957e87dc07df648c')
members : Array (3)
createdAt : 2024-04-12T03:48:26.871+00:00
updatedAt : 2024-04-12T04:32:57.086+00:00
__v : 0
```

Express.js and Node.js: Together, these technologies form the backend of our application. Node.js is a powerful JavaScript runtime built on Chrome's V8 JavaScript engine, and Express.js is a fast, unopinionated, minimalist web framework for Node.js. This combination provides a robust server-side platform that manages API requests and orchestrates the application's server-side logic.

React.js: React.js is a declarative, efficient, and flexible JavaScript library for building user interfaces. It enables the construction of reusable UI components that enhance the interactive elements of the application, providing a dynamic user experience.

Socket.IO: This JavaScript library facilitates real-time, bidirectional event-based communication. In our application, Socket.IO is crucial for implementing real-time chat features, allowing messages to be sent and received instantly without refreshing the web page.

Cloudinary: Integrated for media management, Cloudinary offers an efficient solution for uploading, storing, and delivering images, videos, and other files. It supports real-time transformations and optimizations, enhancing the application's capability to handle media-rich content.

Tailwind CSS: A utility-first CSS framework used for styling our application. Tailwind CSS provides low-level utility classes that help in building custom designs without leaving your HTML. It's instrumental in creating a responsive and aesthetically pleasing user interface, facilitating rapid UI development that is consistent across different screen sizes.

CHAPTER 5

IMPLEMENTATION

5.1 WEEKLY ANALYSIS:

Week 1: Project Initiation and Frontend Design The project kicked off with a clear definition of requirements and a structured planning phase. The team started by familiarizing themselves with React and Next.js to craft the frontend. The initial task was designing the login and sign-up pages, focusing on user interface elements using HTML, CSS, and React components. Special attention was given to robust input validation for passwords, ensuring they included a capital letter, a special character, and at least one digit to enhance security.

Week 2: Styling with Tailwind CSS and DaisyUI During the second week, the team explored the popular Tailwind CSS framework, integrating it with DaisyUI to enhance the application's aesthetics and responsiveness. This integration significantly improved the interactivity of the user interface. The onboarding page was developed, connecting seamlessly with the login system to ensure a smooth user transition post-authentication.

Week 3: Database Setup with MongoDB In week three, the focus shifted to backend development, particularly around setting up MongoDB for data management. The team developed and tested various models to handle user data effectively, ensuring that the database design was robust and scalable to support the application’s needs.

Week 4: Real-Time Chat Functionality The fourth week was pivotal as the team implemented real-time chat functionality using Socket.IO. This included setting up the core framework for users to send and receive messages instantly. Additionally, the contact search functionality was integrated, allowing users to search and send friend requests, enhancing the interactive experience.

Week 5: Contact Management and Notifications This week saw the refinement of contact management and the introduction of a notification system. Whenever a user received a new friend request, a notification would prompt them to accept or reject it. Group chat functionalities were also implemented, enabling users to form and interact within groups.

Week 6: Group Management Enhancements Further enhancements were made to the group chat functionality by empowering group administrators with the ability to add or remove members. This added a layer of dynamic interaction within the application, making the group management more flexible.

Week 7: File Sharing Integration The integration of Cloudinary marked a significant enhancement in week seven, enabling users to share files, images, audio, and videos. This feature was crucial for enriching the communication experience, allowing users to share a variety of media effortlessly within chats.

Week 8: Admin Dashboard Development The development of an admin dashboard was the main focus of week eight. Accessible via a super key, this dashboard provided administrators comprehensive insights into user activities, chat metrics, and group dynamics, offering tools to analyze daily traffic and monitor the overall health of the application.

Week 9: Deployment and Learning The final week was dedicated to learning about deployment platforms such as Vercel and Render. The team deployed the frontend of the application to Vercel and set up backend services on Render, ensuring the application was robust, scalable, and accessible to a global audience.

CHAPTER 6 RESULTS

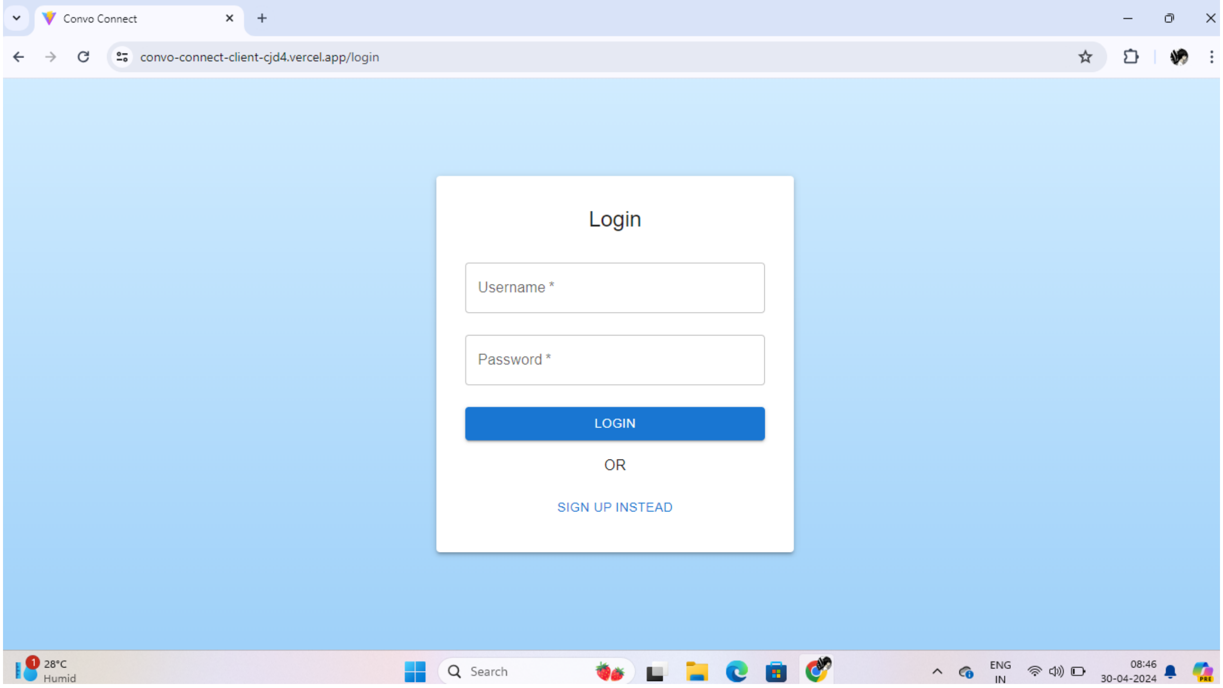
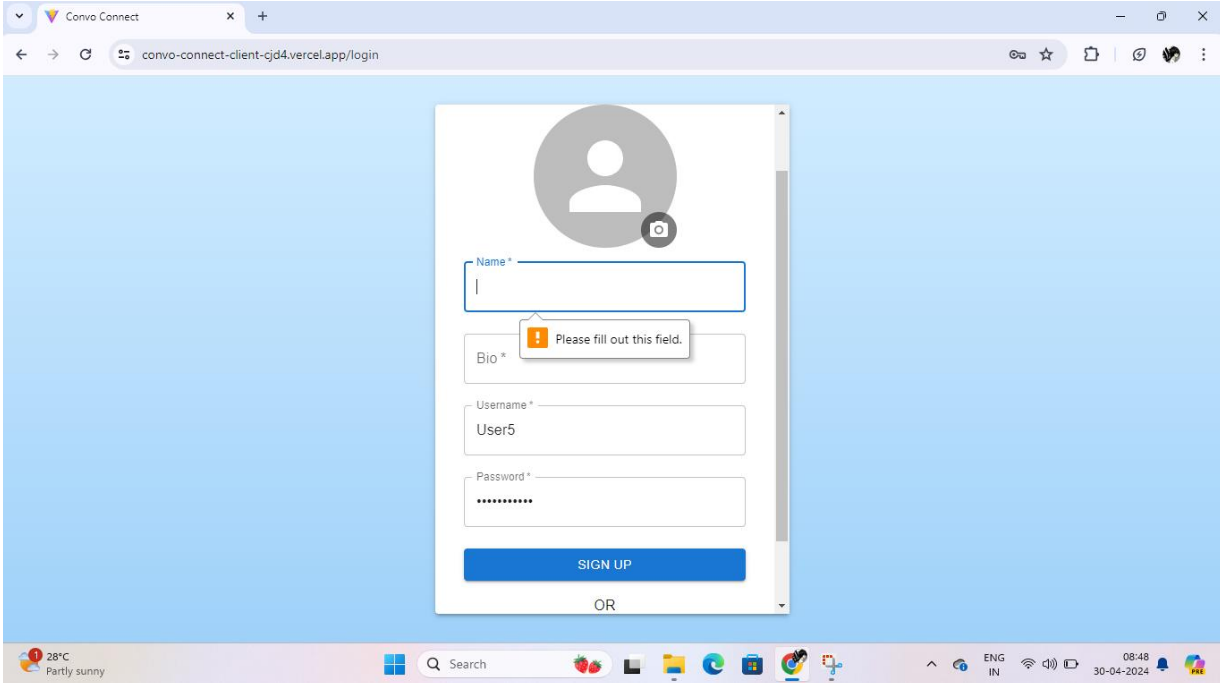


fig.1.Basic/Login Page

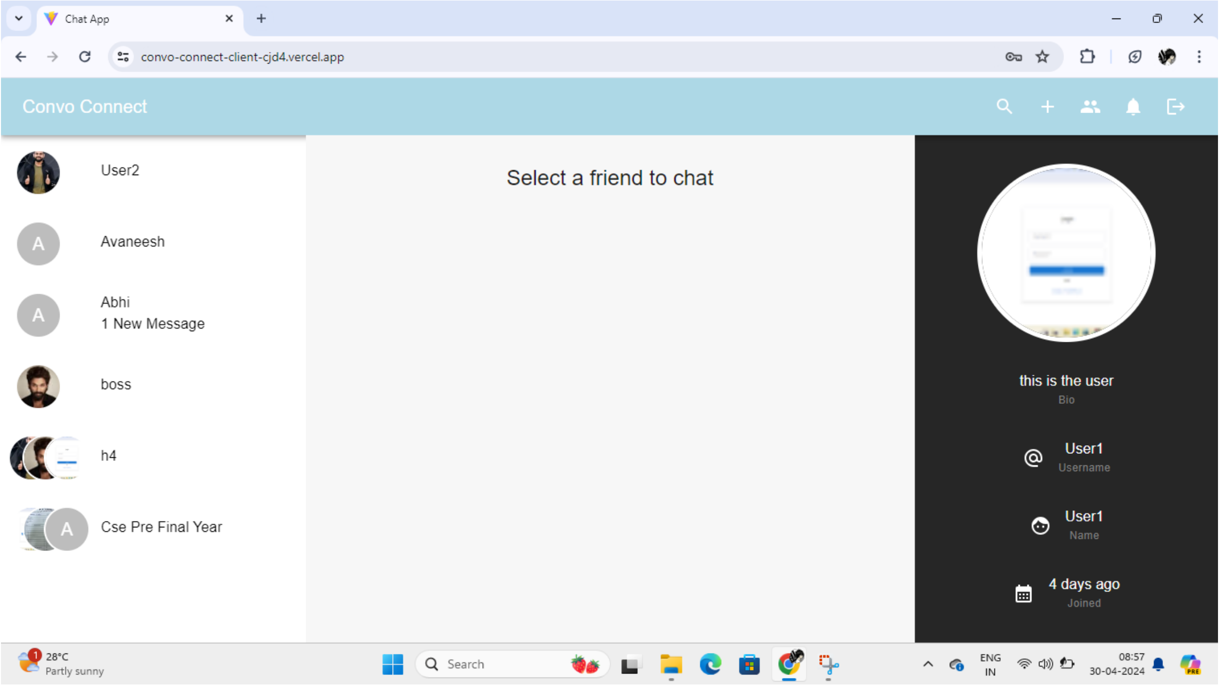


Please fill out this field.

Username * User5

Password * •••••••••••

fig.2.Sign in Page with input validations



User2

Avaneesh

Abhi
1 New Message

boss

h4

Cse Pre Final Year

Select a friend to chat

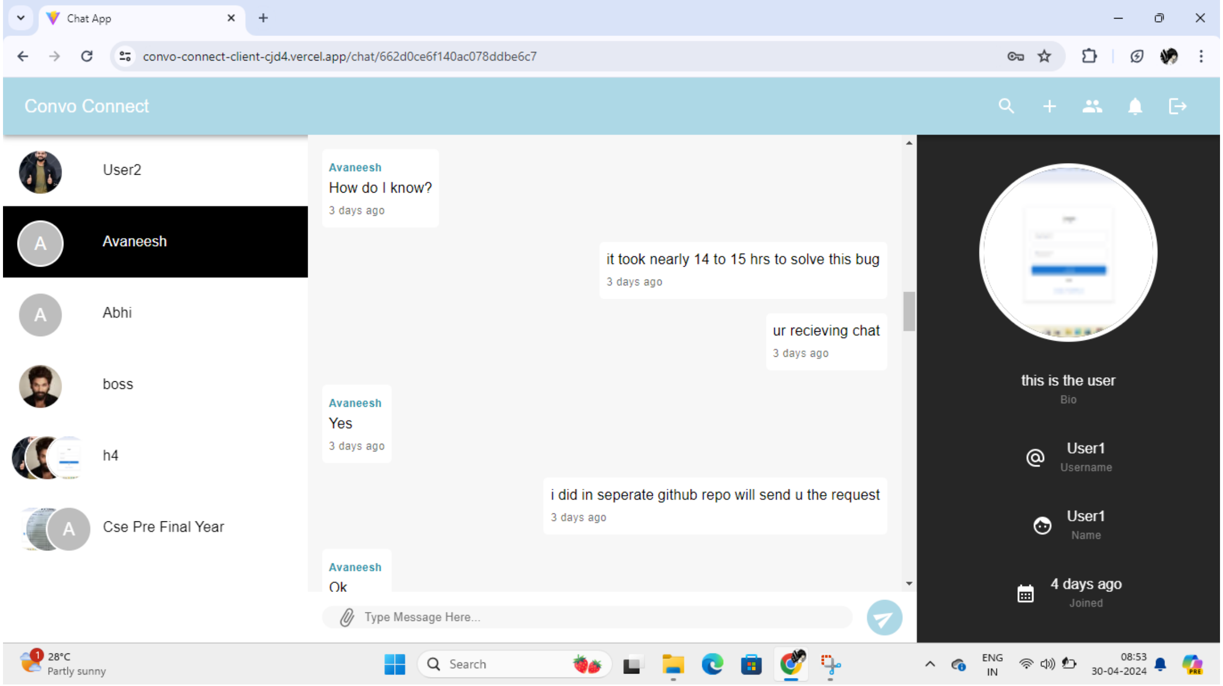
this is the user
Bio

User1
Username

User1
Name

4 days ago
Joined

fig.3.Chatt interface with select friend to chat



User2

Avaneesh

Abhi

boss

h4

Cse Pre Final Year

Avaneesh
How do I know?
3 days ago

it took nearly 14 to 15 hrs to solve this bug
3 days ago

ur recieving chat
3 days ago

Avaneesh
Yes
3 days ago

i did in seperate github repo will send u the request
3 days ago

Avaneesh
Ok

this is the user
Bio

User1
Username

User1
Name

4 days ago
Joined

fig.4.Chat with Text Messsages

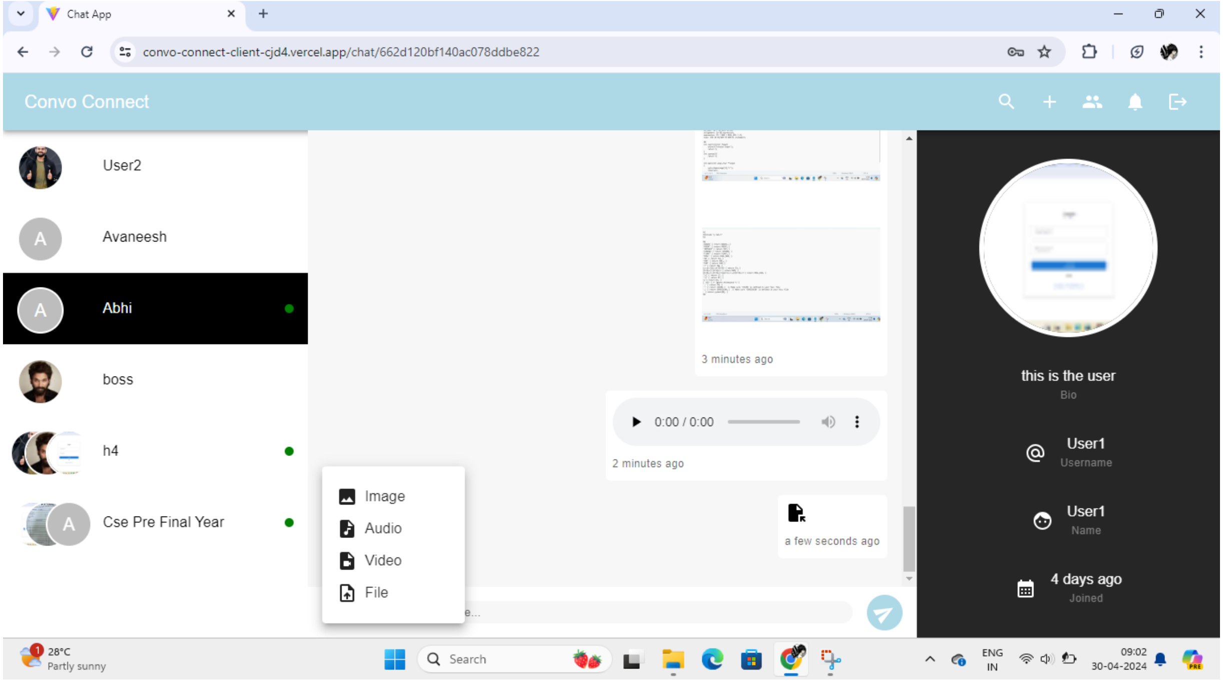


fig.5.Chat with Attachements such as Files Audio Video and File

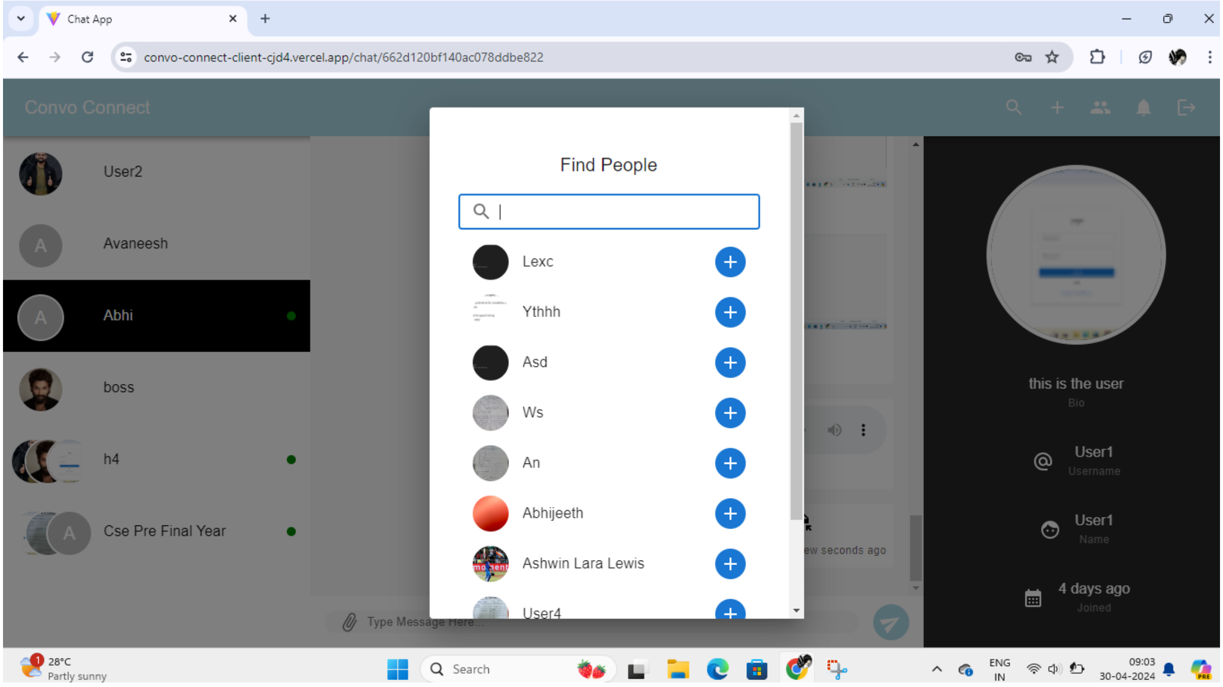


fig.6.Chat with searching friends and sending the request

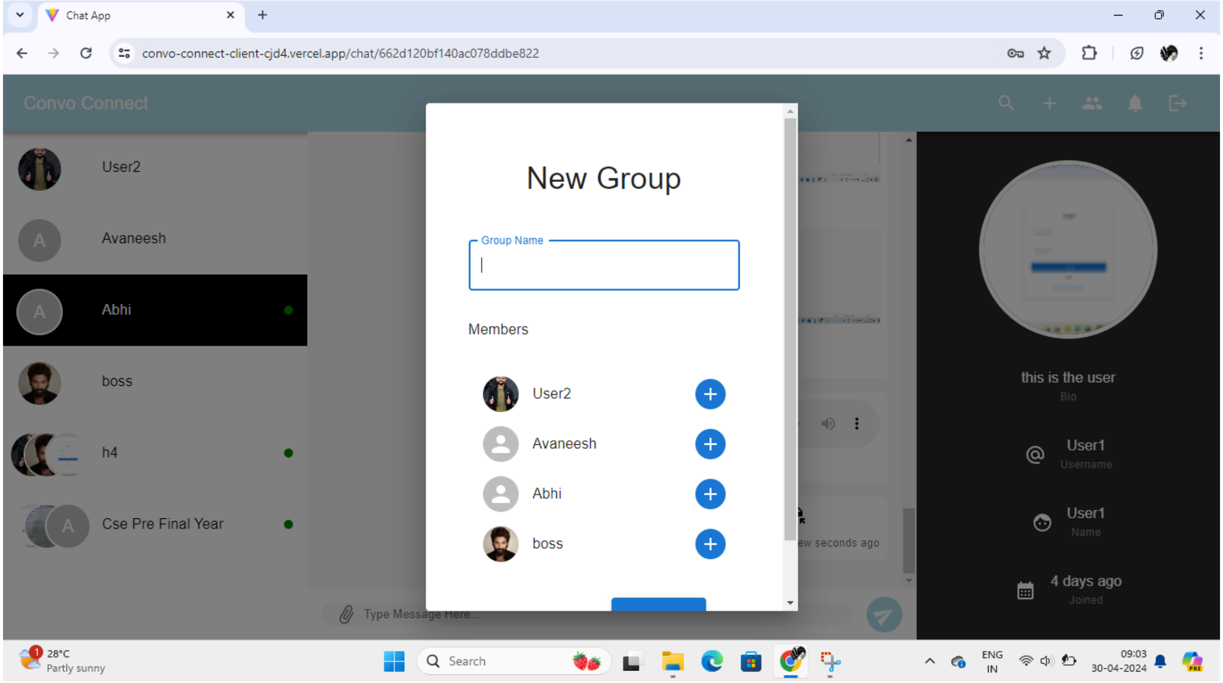


fig.7.Making the group giving the name to it and add member

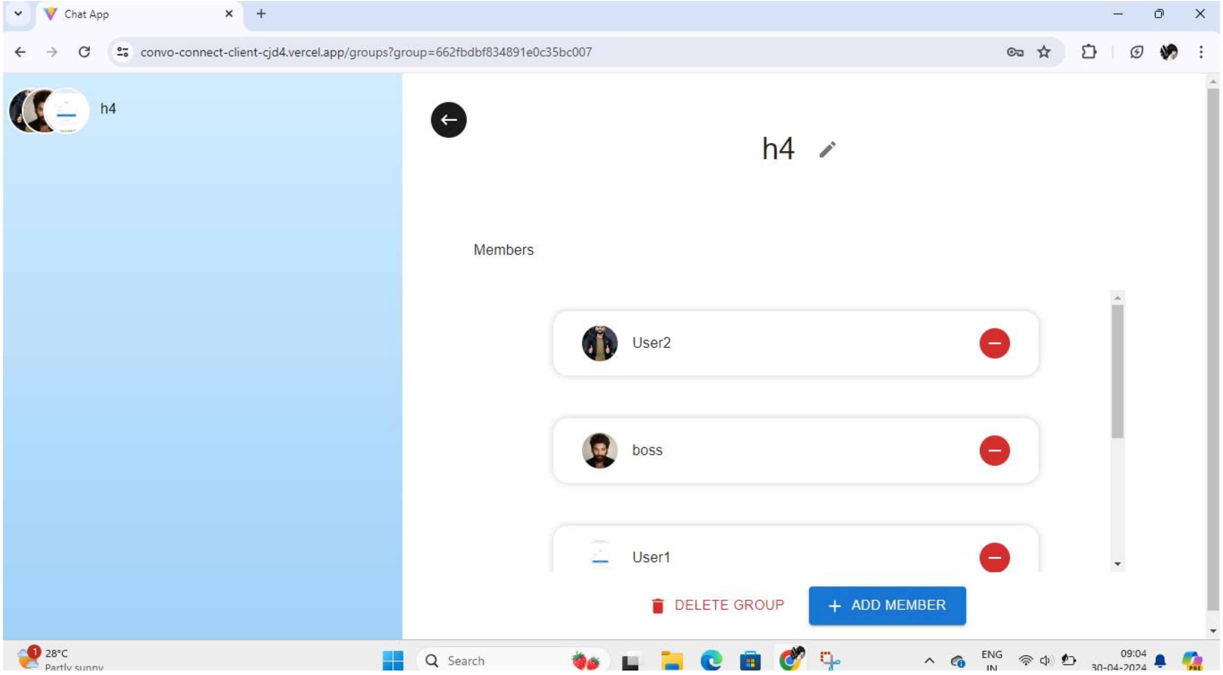
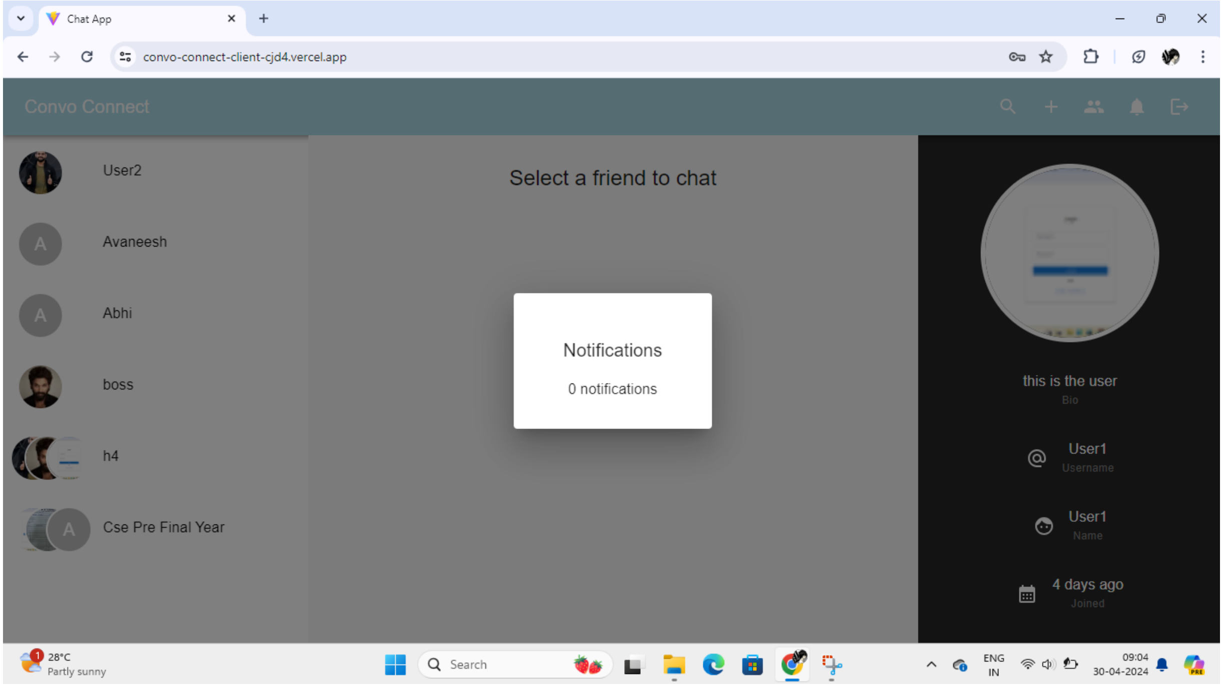

Chat App
convo-connect-client-cjd4.vercel.app/groups?group=662fbdbf834891e0c35bc007
h4
h4
Members
User2
boss
User1
DELETE GROUP
ADD MEMBER

fig.8.Access is given to group he can add or delete and change the name of group

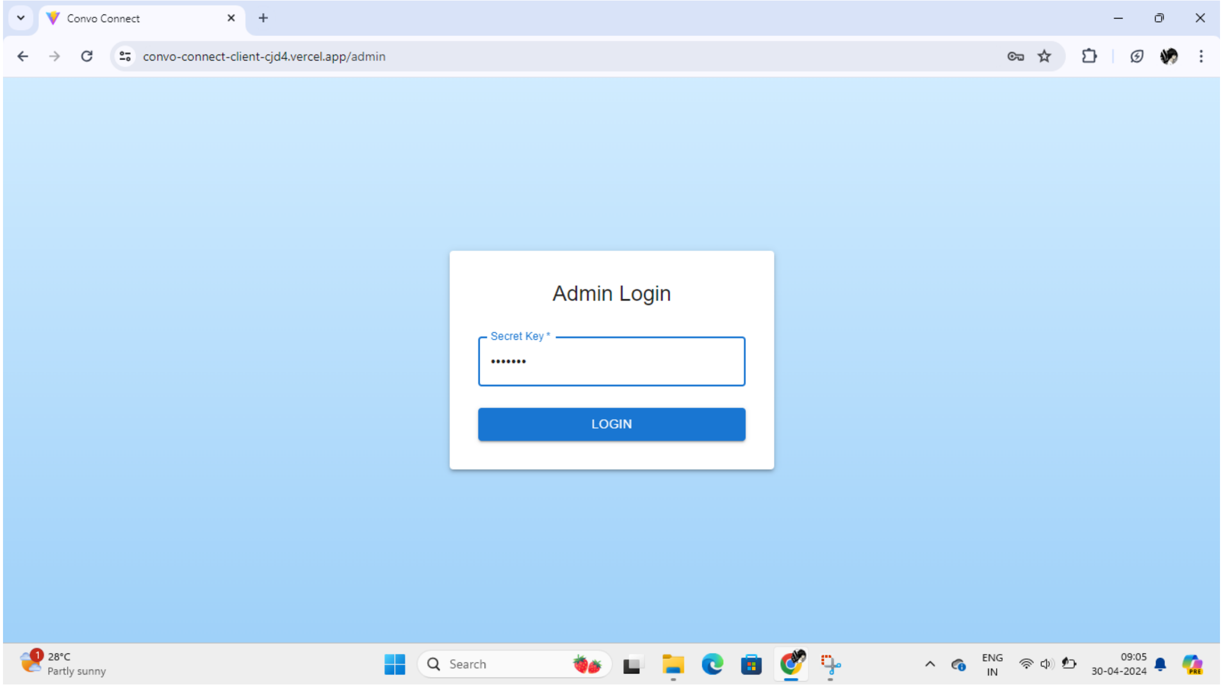


Select a friend to chat

Notifications

0 notifications

fig.9.Chat with Notifications section where user can accept or reject the user



Admin Login

Secret Key *

•••••••

LOGIN

fig.10.Admin Login which is accesed by secret key

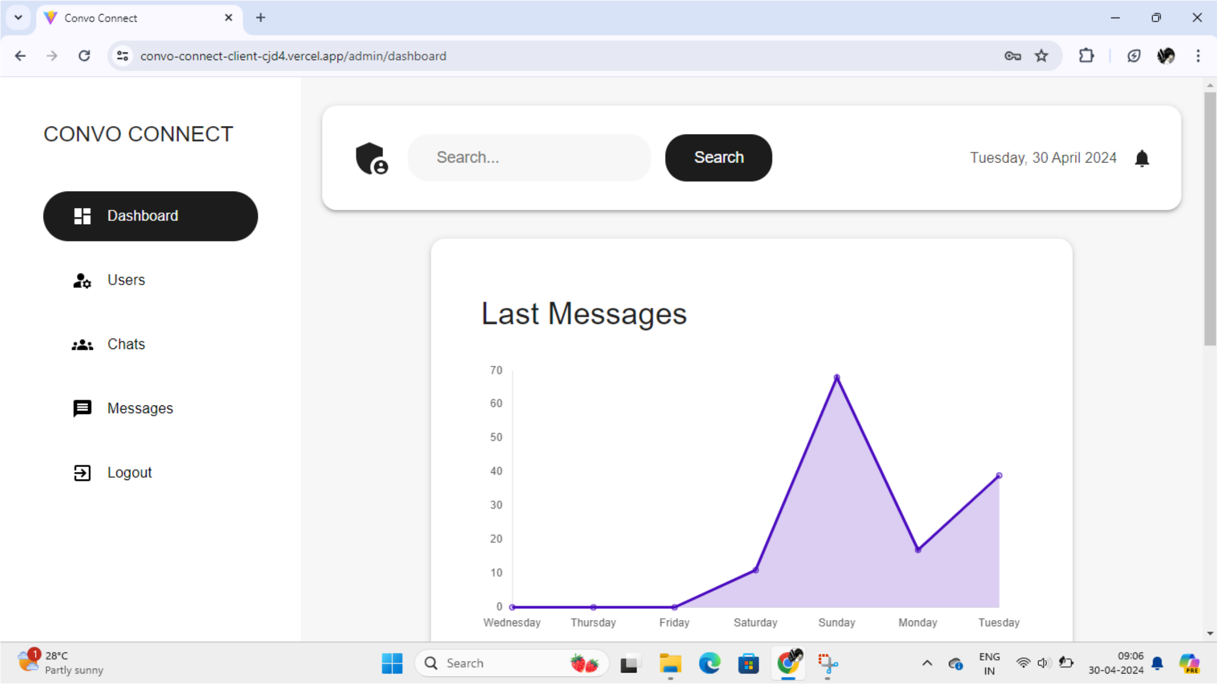


fig.11.Admin dashboard where he can see the traffic of the app

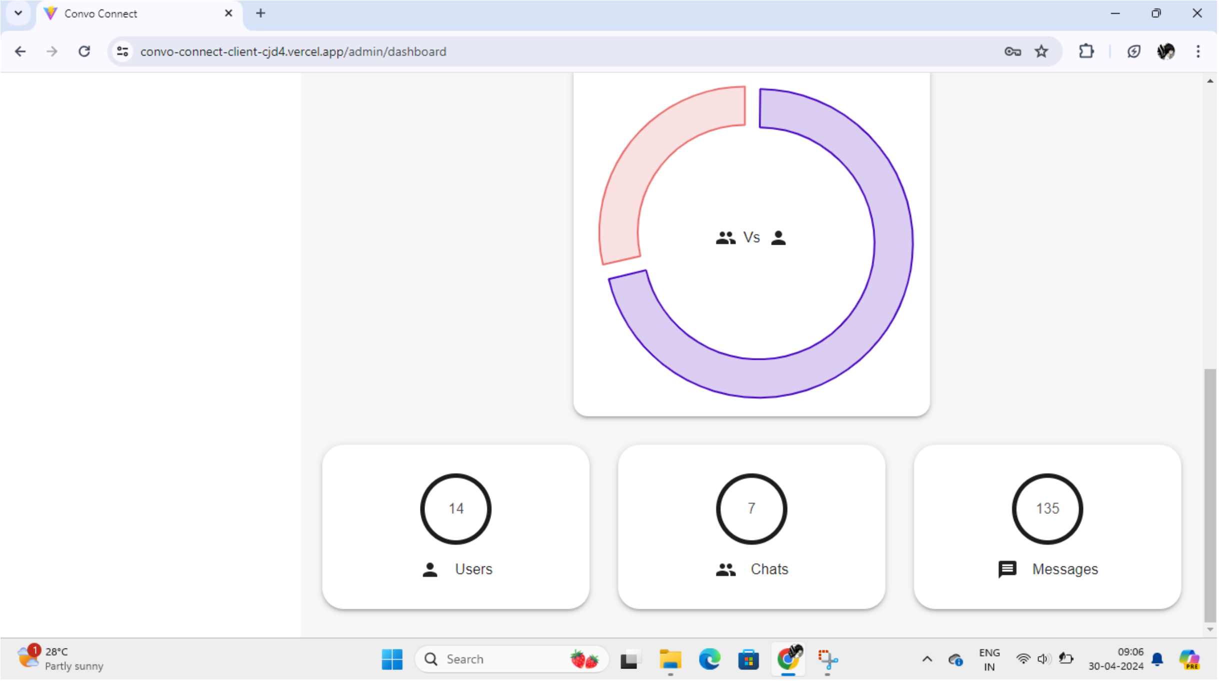
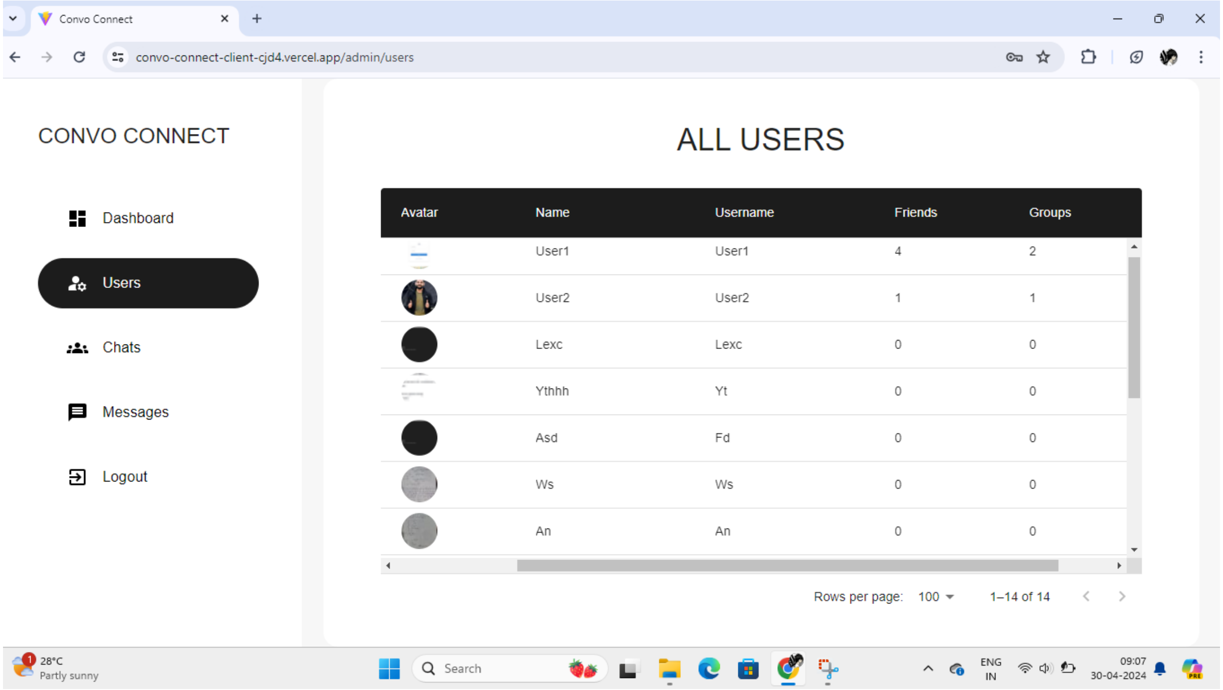


fig.12.Single vs group chat with no of users chats and messages in the app

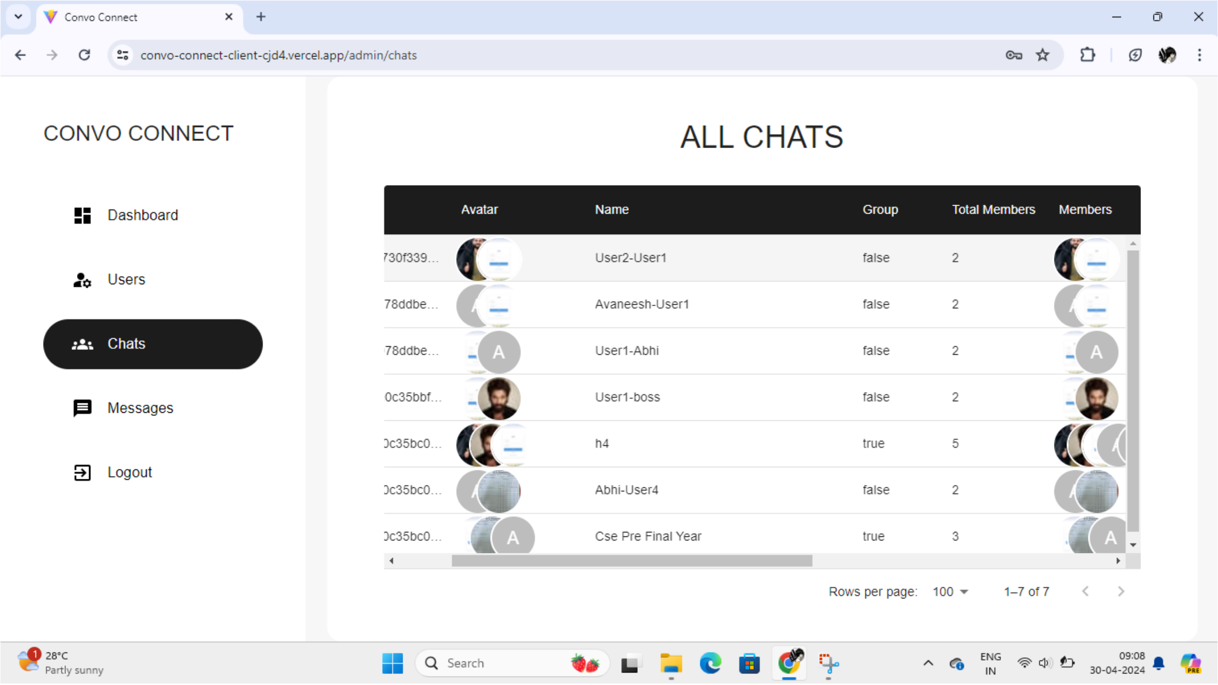


ALL USERS

Avatar	Name	Username	Friends	Groups
	User1	User1	4	2
	User2	User2	1	1
	Lexc	Lexc	0	0
	Ythhh	Yt	0	0
	Asd	Fd	0	0
	Ws	Ws	0	0
	An	An	0	0

Rows per page: 100 1–14 of 14

fig.13.Admin viewing the Users informations



ALL CHATS

	Avatar	Name	Group	Total Members	Members
730f339...		User2-User1	false	2	
78ddbe...		Avaneesh-User1	false	2	
78ddbe...	A	User1-Abhi	false	2	A
0c35bbf...		User1-boss	false	2	
0c35bc0...		h4	true	5	
0c35bc0...		Abhi-User4	false	2	
0c35bc0...	A	Cse Pre Final Year	true	3	A

Rows per page: 100 1–7 of 7

fig.14.Adming viewing the chat information with information like in which group user is how many members group has etc

CHAPTER 7 CONCLUSIONS

In conclusion, the development journey of our real-time chat web application has been both challenging and rewarding, culminating in a platform that redefines online communication. By harnessing the power of React.js,Tailwind css,MongoDb,Node/Express,Cloudinary and Socket.IO, we have created a dynamic and feature-rich environment for seamless interaction.

Throughout the project, we have prioritized user engagement and satisfaction, resulting in the successful integration of innovative features. From the initial login process facilitated by Firebase authentication to the creation of personalized profiles during onboarding, every step has been meticulously designed to enhance the user experience.

Moving forward, we envision further enriching the application with additional features. These may include:

Advanced message formatting options, such as emojis, stickers, and GIFs, to add fun and expressiveness to conversations.

Integration of video calling functionality, allowing users to connect face-to-face for more immersive interactions.

By continuously innovating and incorporating new features, we aim to maintain the application's relevance and appeal in an ever-evolving digital landscape. Our commitment to providing a cutting-edge and user-centric platform remains steadfast, and we are excited about the future possibilities for our real-time chat web application.

CHAPTER 8

REFERENCES

[1] https://www.youtube.com/watch?v=keYFkLycaDg

[2] https://reactjs.org/docs

[3] https://mongodb.com/docs

[4] https://daisyui.com

[5] https://cloudinary.com

[6] https://www.youtube.com/watch?v=HwCqsOis894