MERN Chat App with Socket.IO

This presentation will guide you through the features of a real-time chat app built using the MERN stack and Socket.IO. We'll explore key features like logging in, sending live messages, sharing images, and showing whether users are online or offline.

HK **by Nisha**

REAL TIME CHAT APPLICATION
SOCKET.IO REACTJS NODEJS
Jane Doe
Agent (Online)
Hey there! Any question?
Hello!
20m ago
Hey! Would you like to talk sales, support, or anyone?
Lorem Ipsum is simply dummy text of the printing and typesetting industry.
Just now. Not seen yet
Send a message
We make it simple and seamless for businesses and people to talk to each other. Ask us anything
Where can I get some?
The standard chuck...
There are many variations of passages of Lorem Ipsum

Introduction to the Project

Project Goals

This project was all about building a chat app where users can talk in real-time, send private messages, and share pictures. The main goals were to make it simple, easy to use, and secure. We focused on creating a user-friendly interface that is intuitive to navigate and provides a seamless chat experience.

Key Features

The app lets users log in, send private messages, share images, and store data securely with MongoDB. The app provides a secure login mechanism. Users can enter their credentials and access their private chat spaces. The system encrypts user data, ensuring the safety of their personal information. It uses Socket.IO for smooth, real-time communication and reliable message delivery.

Technologies Used

1 MongoDB

A NoSQL database is used to store user information, messages, and images. Its flexible structure makes it perfect for managing dynamic data easily.

3 React

A JavaScript library used to build user interfaces. It helps create dynamic and interactive front-end features for the chat app.

2 Express.js

A powerful Node.js framework for creating web apps. It helps with routing, middleware, and managing server-side tasks efficiently.

4 Node.js

A JavaScript runtime that lets us run JavaScript code outside a web browser. It's the backbone of the server-side parts of the app.

5 Socket.io

Real-time Updates

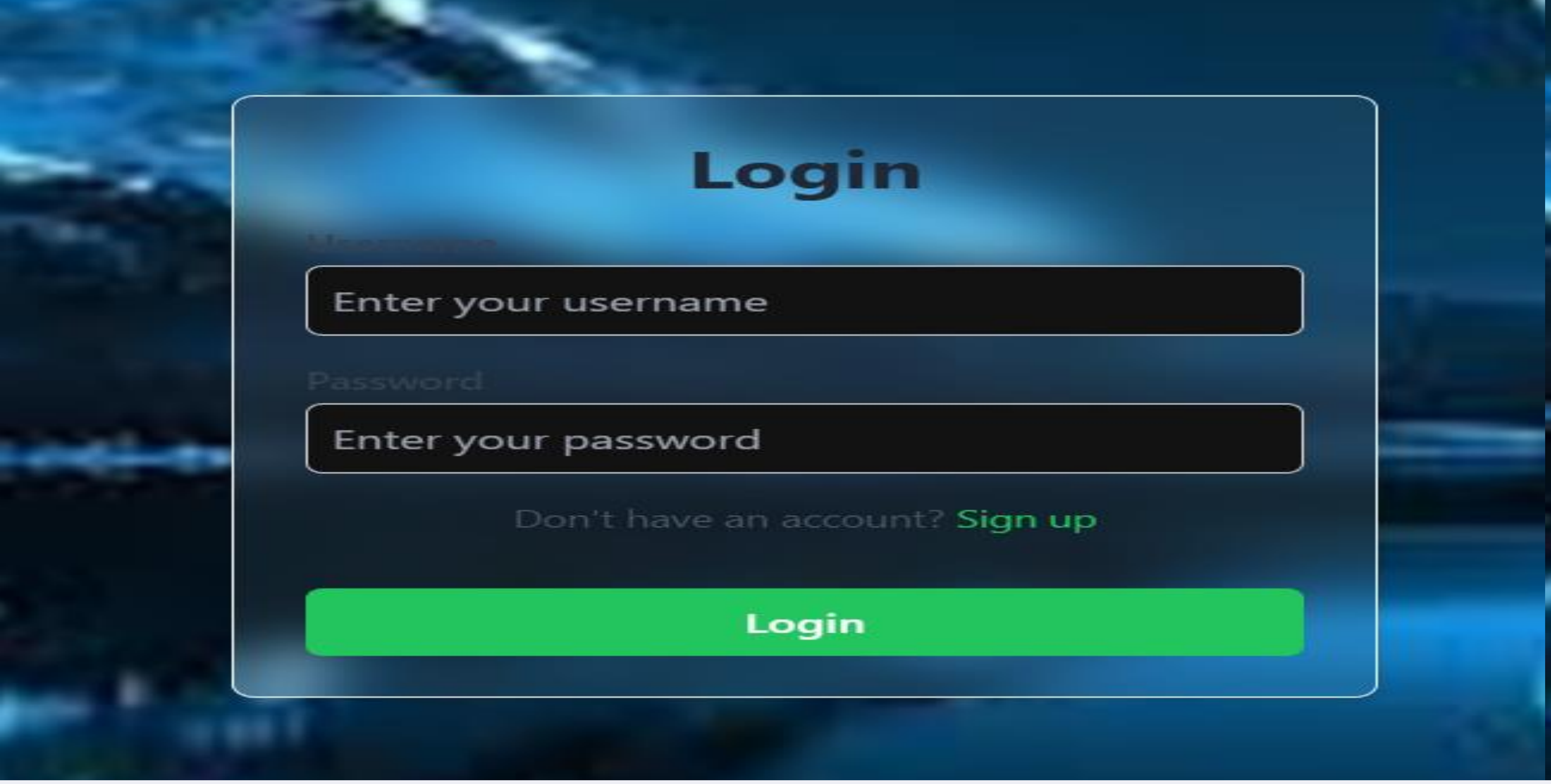
Socket.IO allows the server and client to communicate in real-time, making messages and updates happen instantly.

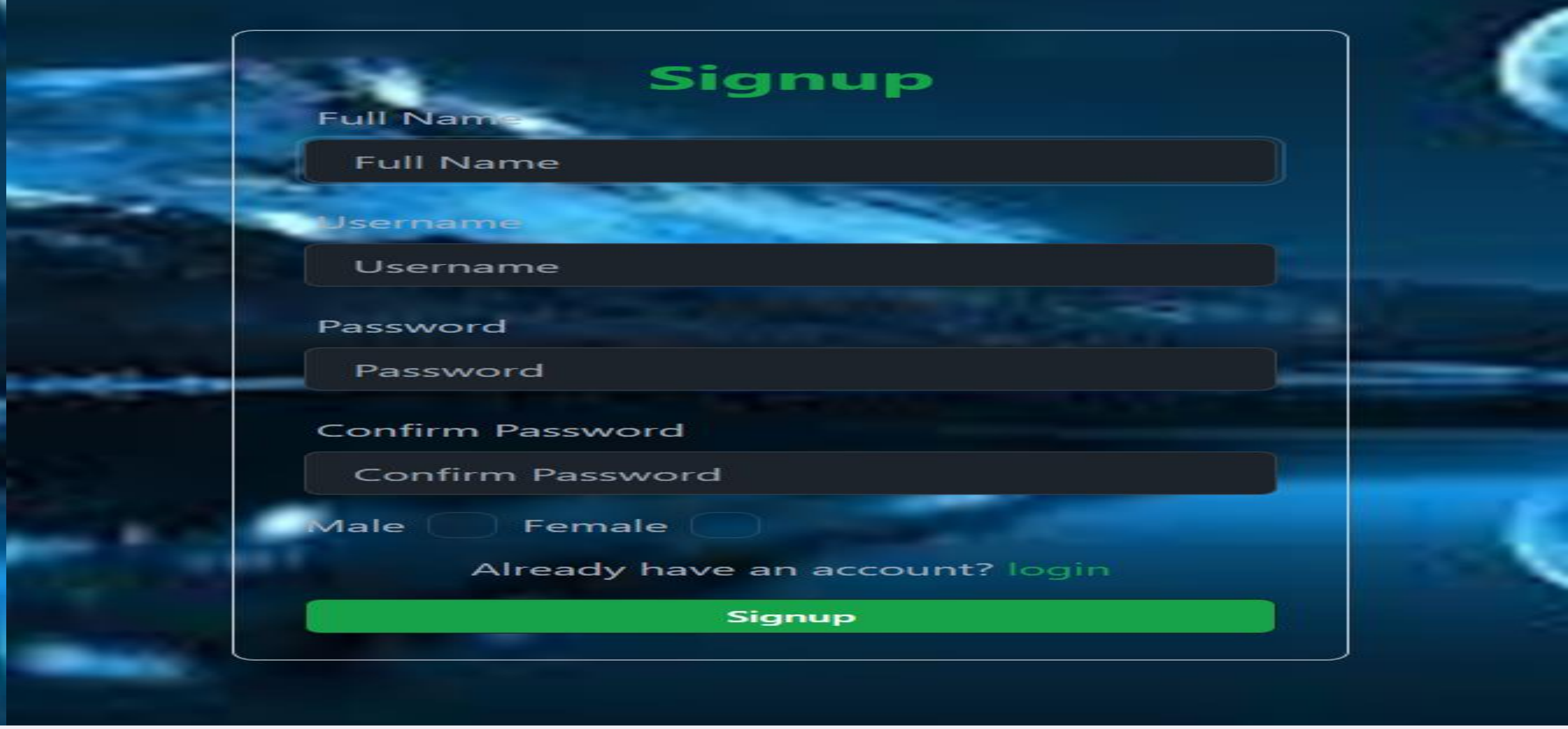
Event-driven Model

It uses an event-based system, where the server and client send and listen for events, allowing them to interact smoothly.

Persistent Connections

Socket.IO keeps a constant connection between the client and server, so they can keep communicating even after the page has loaded.





User Authentication and Logout

Registration

Users can create an account by filling out a simple form with their name and password. After signing up, they get a confirmation notification on top

Authentication

Existing users can log in with their username and password, making sure only authorized people can access the app.

User Logout

The app allows users to securely log out of their accounts, ending their active session. Once logged out, users can be confident their data is safe.

Features and Functionality

Message and real time chat

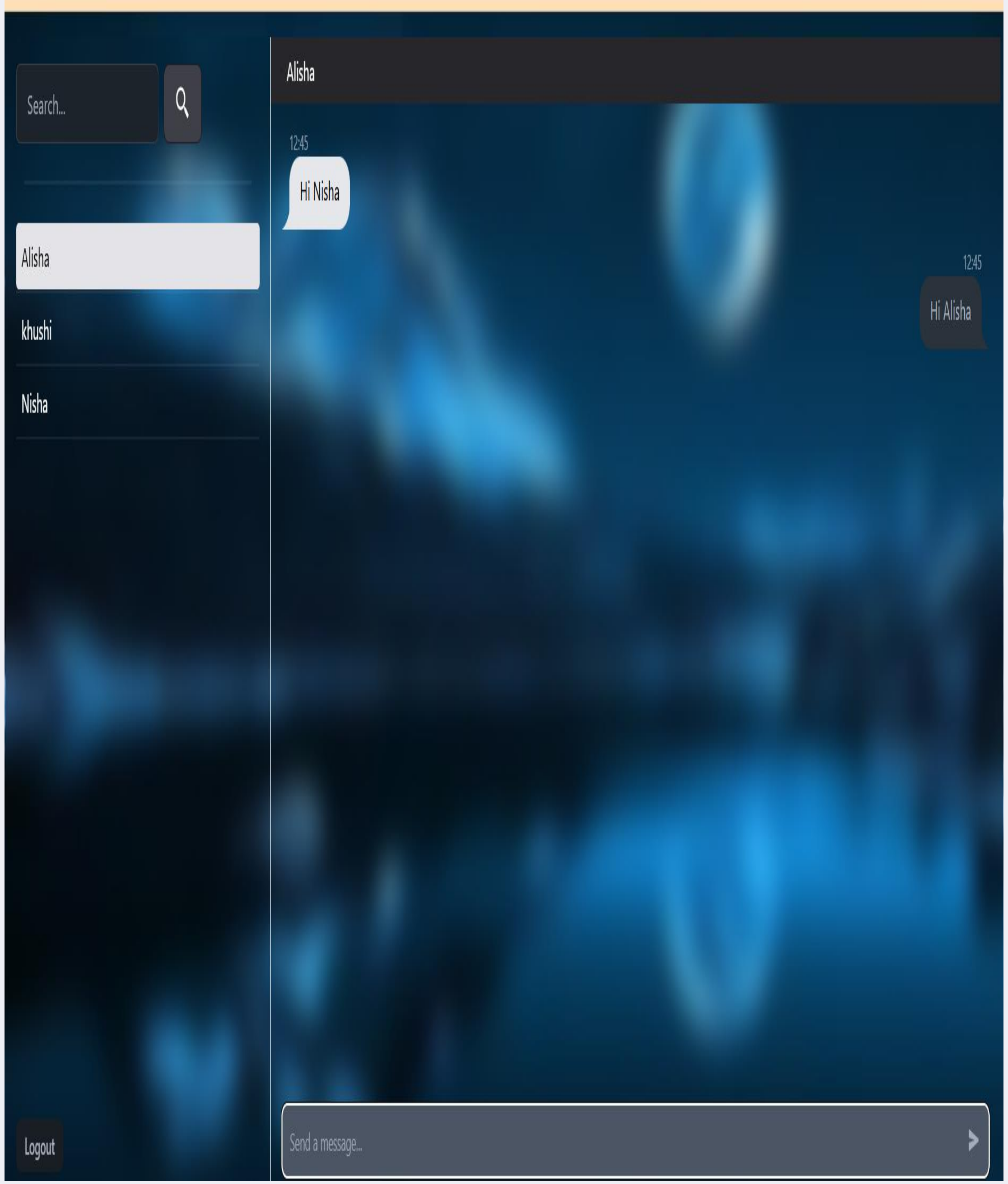
Users can send and receive messages instantly, making it feel like a real conversation. This keeps communication fast and seamless.

Time and Date Stamps

Each message includes a timestamp, indicating the time and date it was sent. It easier to follow the flow of conversations, especially in long chat threads.

Message privately and Online status

Users can start private chats with specific people for secure and personal conversations. This ensures their messages stay private and only visible to the intended recipient. The app displays the online/offline status of users in real-time.



12:45
Hi Nisha

12:45
Hi Alisha

User Interaction

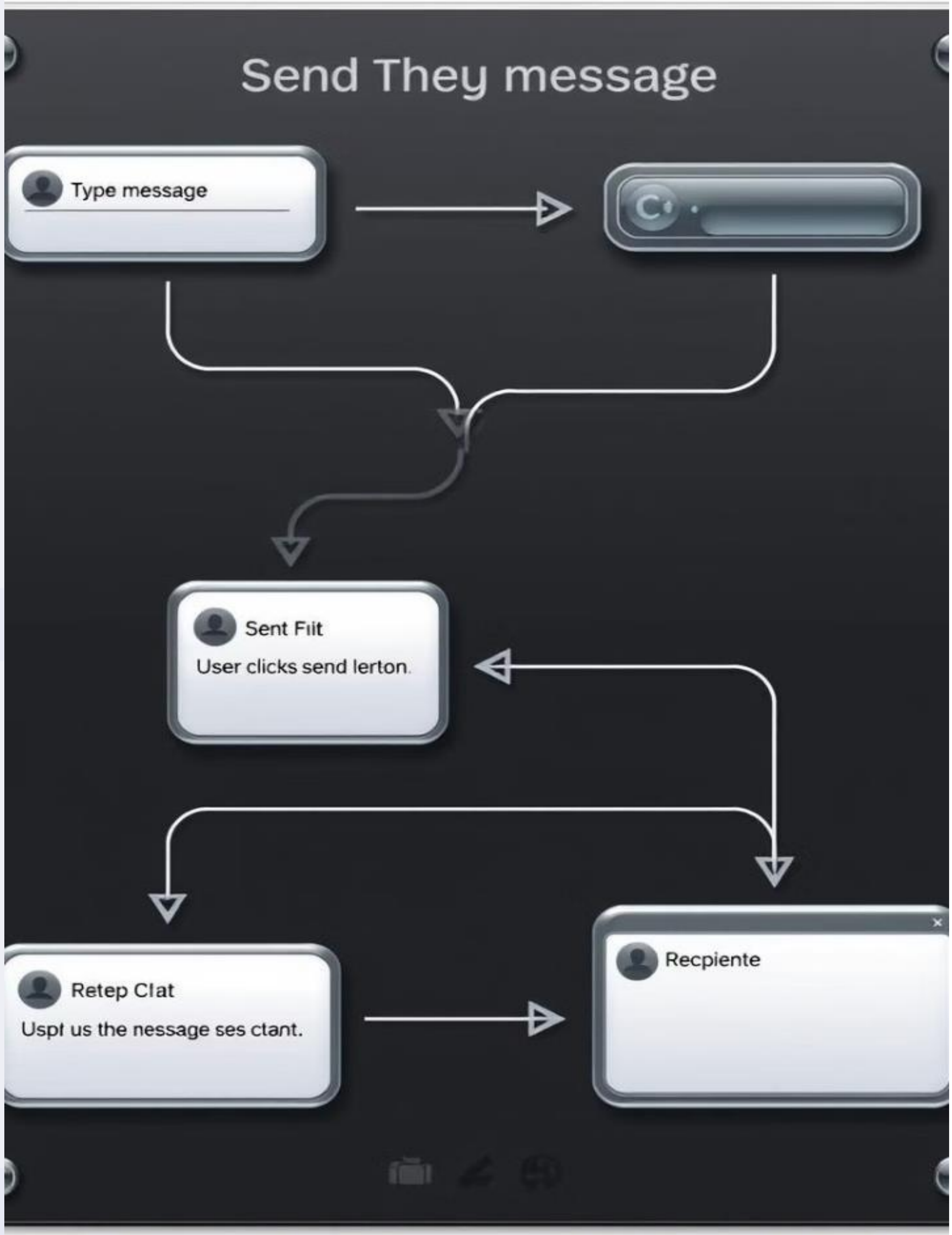
1 User sign-up and login with their credentials(username and password.

2 The dashboard display a list of available contacts, recent conversation and online/offline statues.

3 The user types a message and sends it. The message is sent to the backend server via a Socket.io event.

4 All messages and media are stored in MongoDB for future reference.

5 After logout the Socket.IO connection is terminated, ensuring no further are possible until they login again.



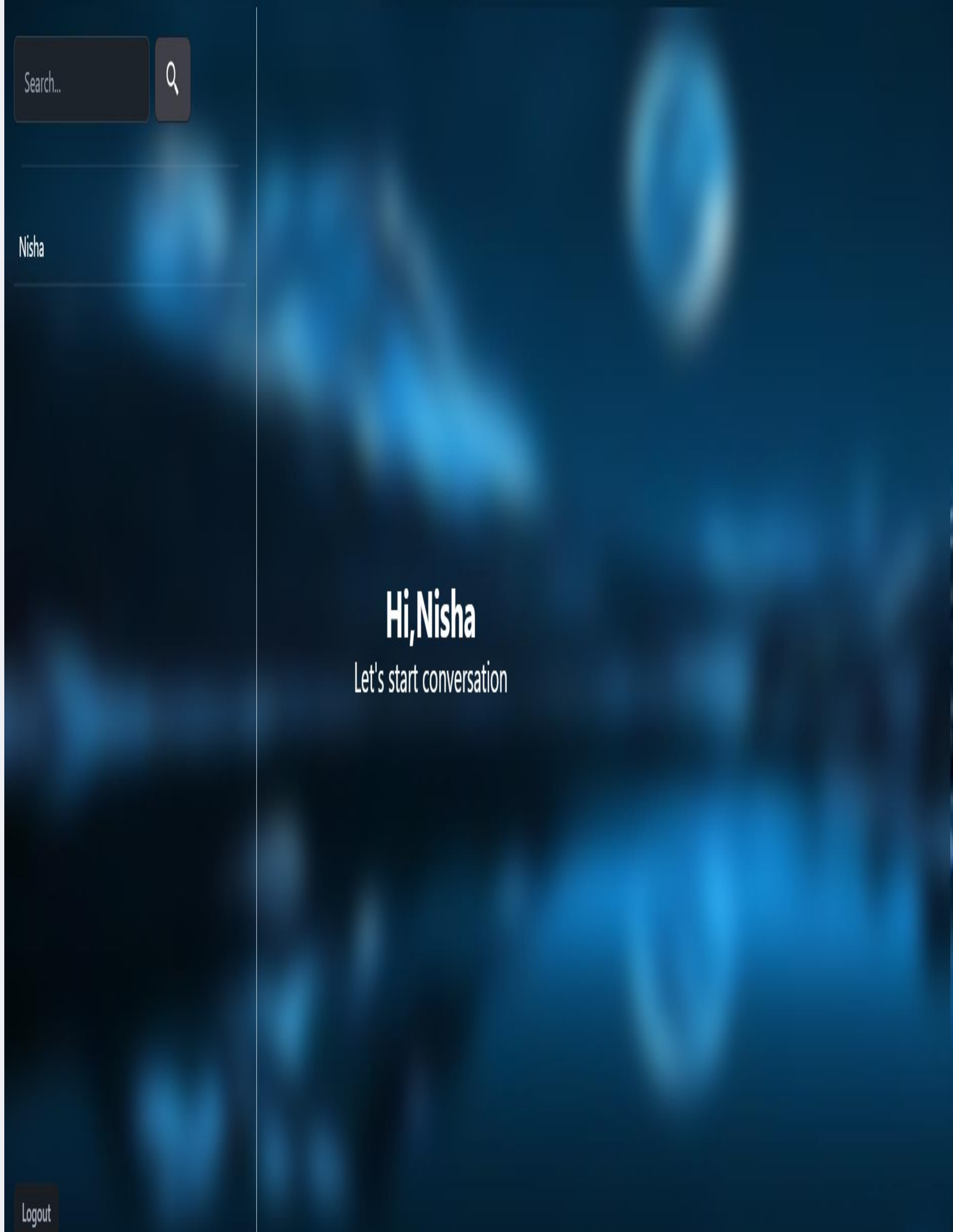
Future work

Group Creation

Implementing group chat functionality, allowing users to create and manage chat groups. This will enable multiple users to communicate in real-time within a shared conversation.

File and Image Sharing

Improve file and image sharing so user can send different file types easily. This will include clearer previews and better storage handling.



Hi,Nisha

Let's start conversation

thank you