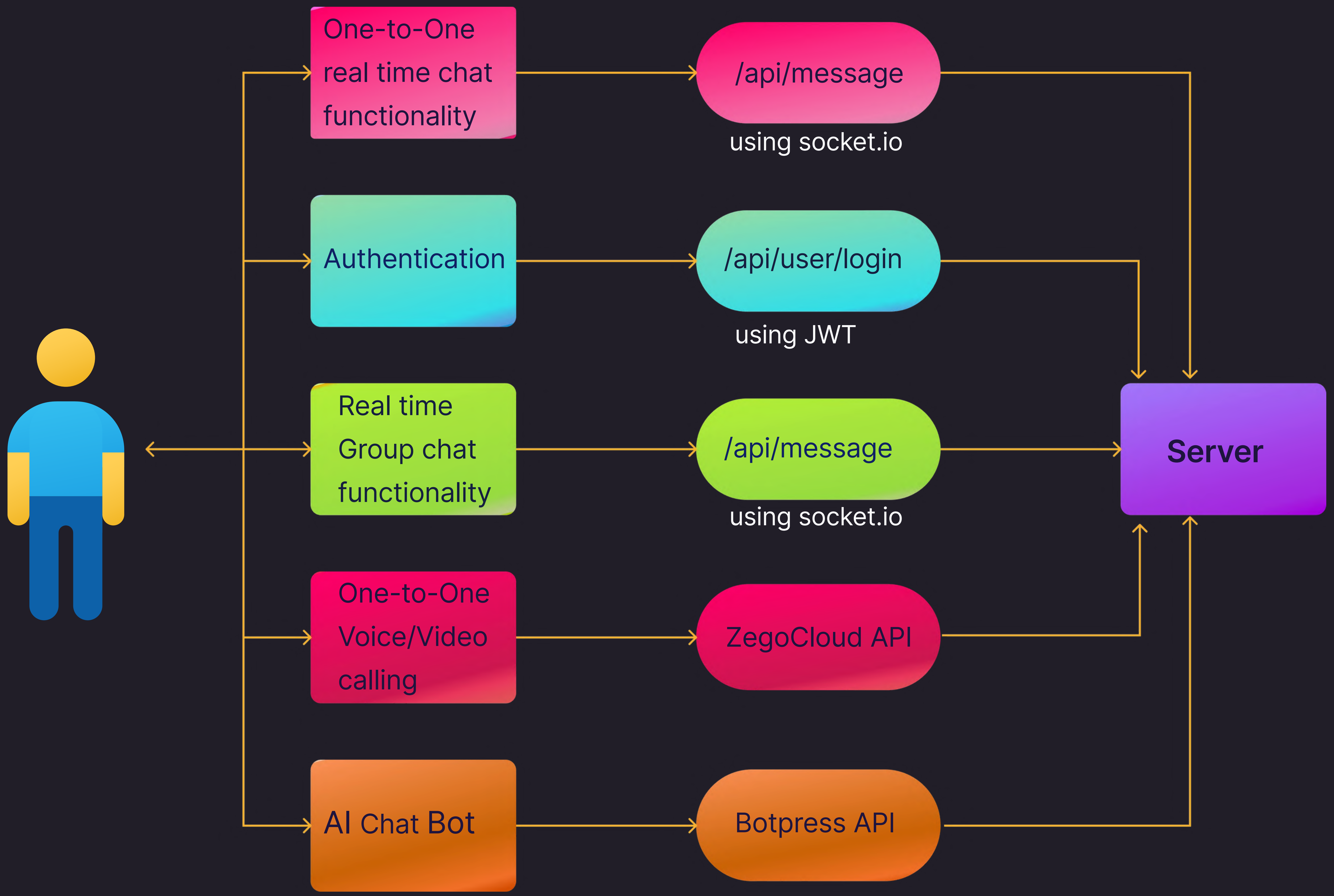
System Design Report: Chats Web Application

Introduction :-

1. The Chat Messaging Web Application is designed to provide users with real-time one-to-one and group chat functionality.
2. The application is equipped with user authentication and authorization using JSON Web Tokens (JWT) for secure access.
3. In addition, it features notification functionality to keep users updated, and audio and video communication capabilities via Zegocloud.
4. To enhance user experience, an AI chatbot is integrated using the Botpress library.

Try Pitch

Use Case Diagram



Try Pitch

Class Diagram

Chat

/api/chat
/api/chat/group
/api/chat/remove
/api/chat/groupname
/api/chat/groupadd

→Group or one-to-one chat
→Update group name
→Remove group member
→Add group member

User

/api/user
/api/user/login

→User exit or not
→User login, signup

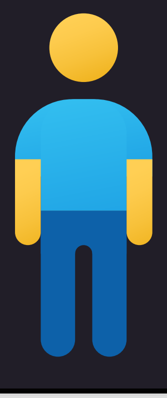
Message

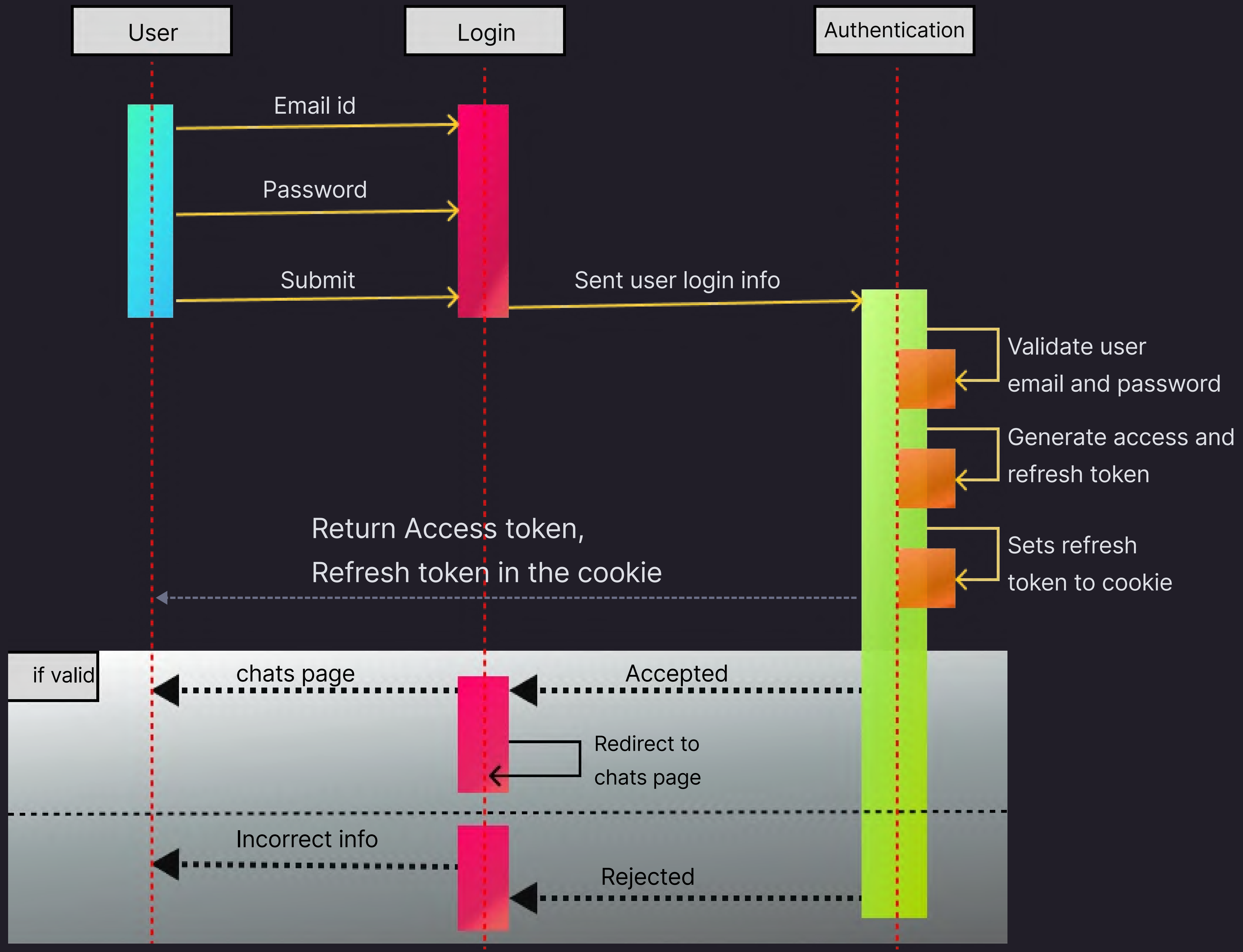
/api/message
/api/message/:chatId

→Send message
→Receive message
→Fetch all message

Try Pitch

Login Sequence Diagram





User

Login

Authentication

Email id

Password

Submit

Sent user login info

Validate user
email and password

Generate access and
refresh token

Sets refresh
token to cookie

Return Access token,
Refresh token in the cookie

if valid

chats page

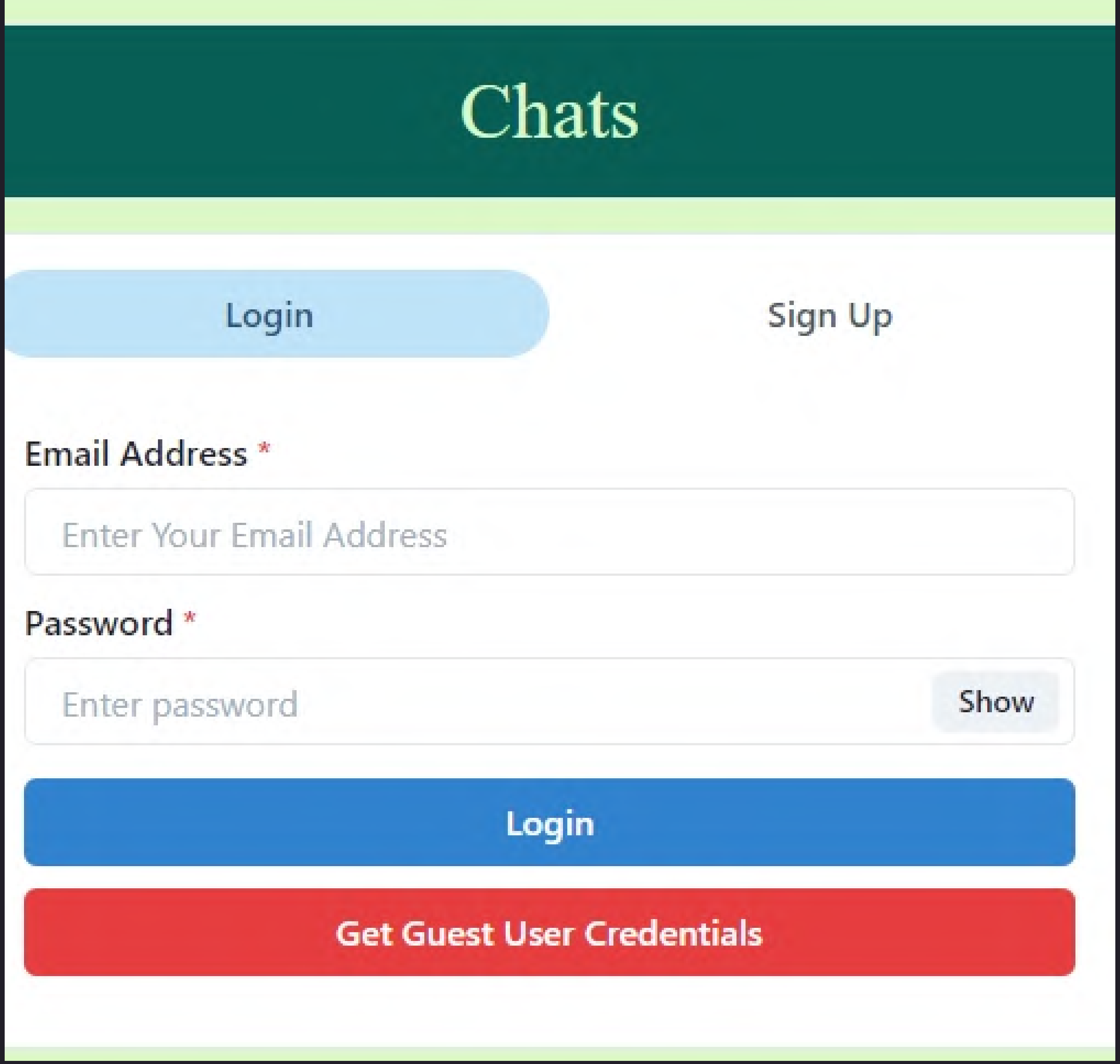
Accepted

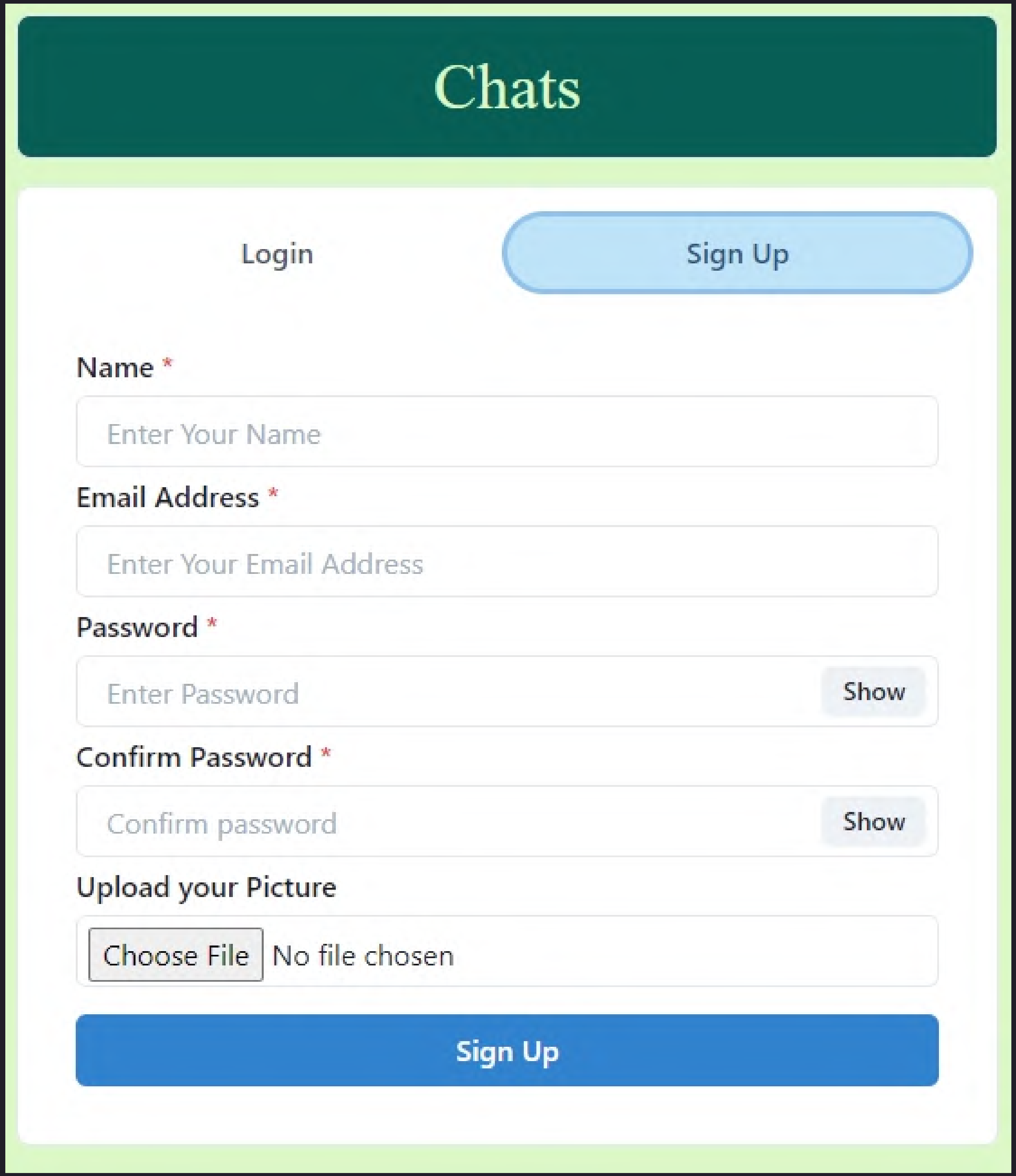
Redirect to
chats page

Incorrect info

Rejected

Try Pitch


Chats
Login
Sign Up
Email Address *
Enter Your Email Address
Password *
Enter password
Show
Login
Get Guest User Credentials


Chats
Login
Sign Up
Name *
Enter Your Name
Email Address *
Enter Your Email Address
Password *
Enter Password
Show
Confirm Password *
Confirm password
Show
Upload your Picture
Choose File
No file chosen
Sign Up

Try Pitch

Chat Sequence Diagram

Functional Requirements

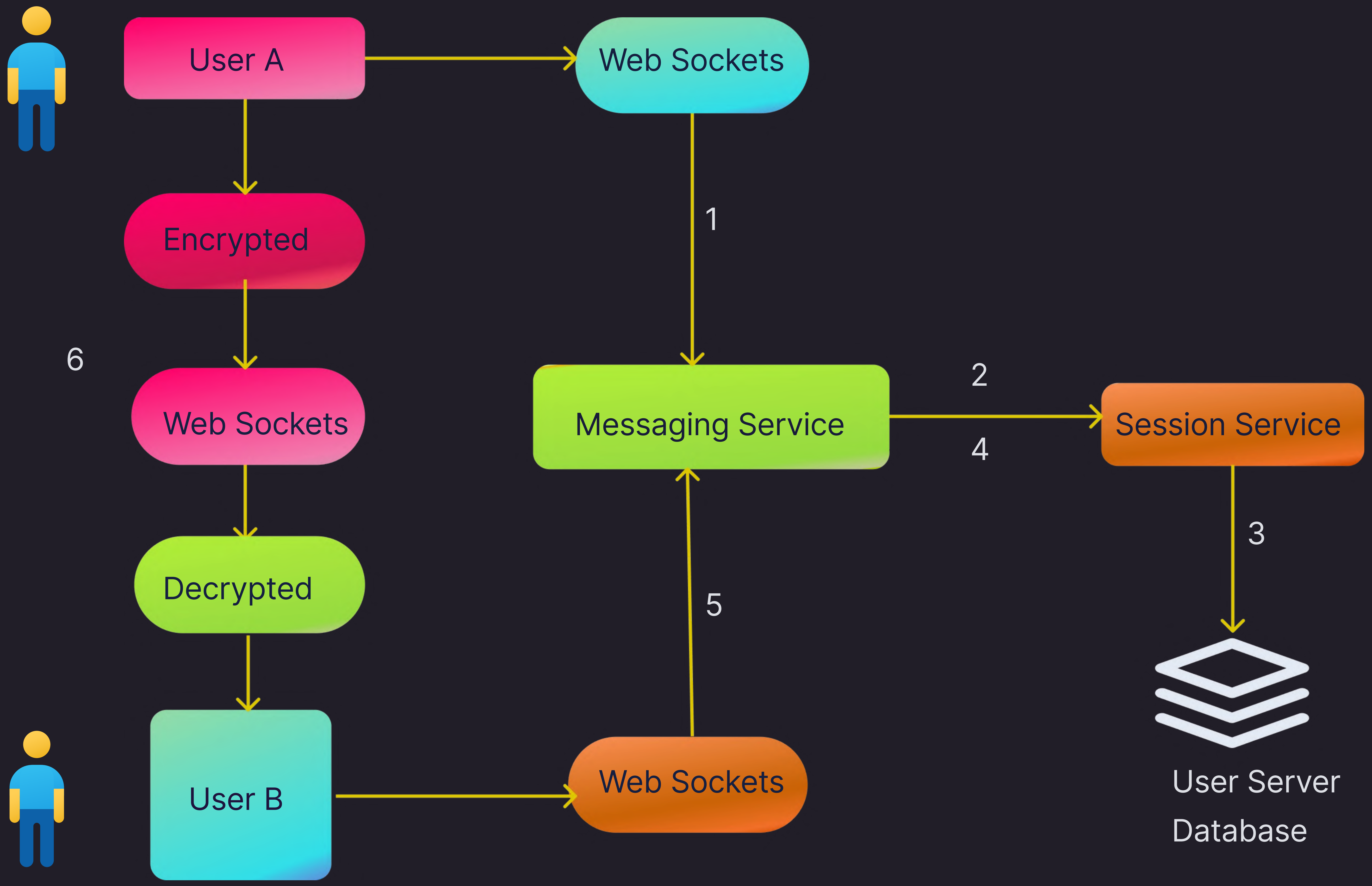
1. One-to-one chat
2. Group chat
3. Notifications

System Requirements

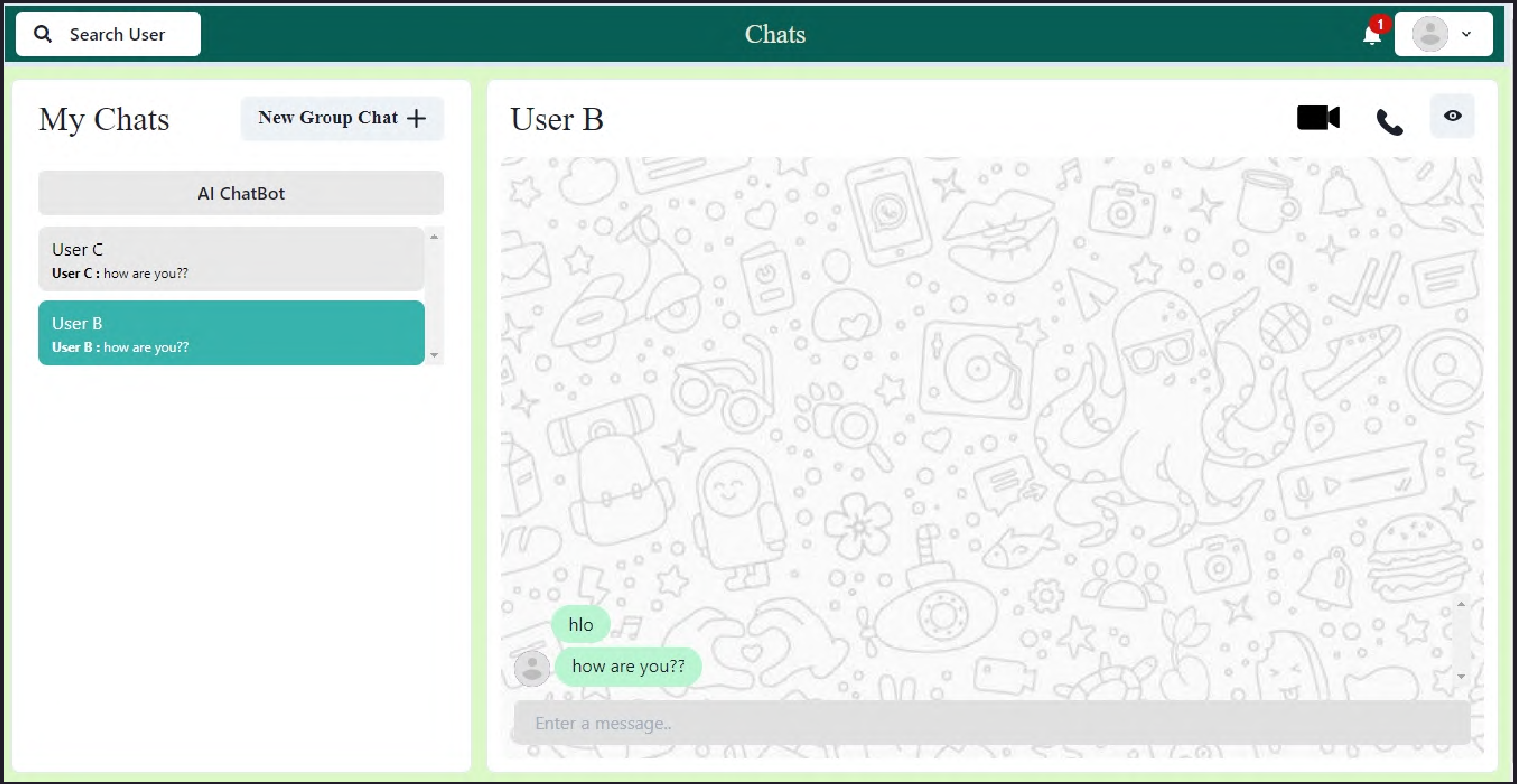
1. Low latency
2. High reliability
3. High availability
4. Mobile & desktop
5. Chat History
6. ELE Encryption
7. Web Sockets

Try Pitch

One-to-one Chat Sequence Diagram



Try Pitch



User C : how are you??

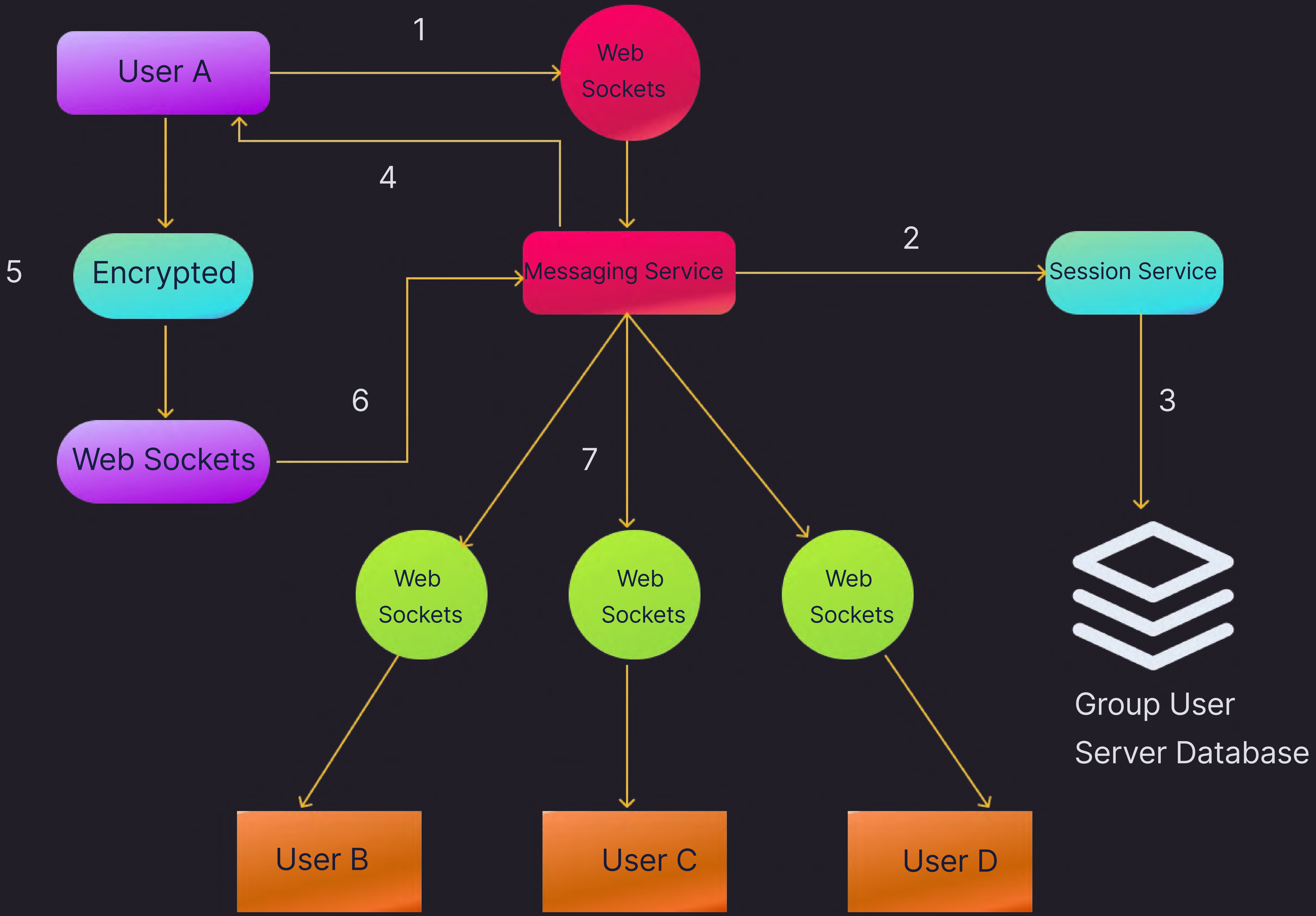
User B : how are you??

hlo

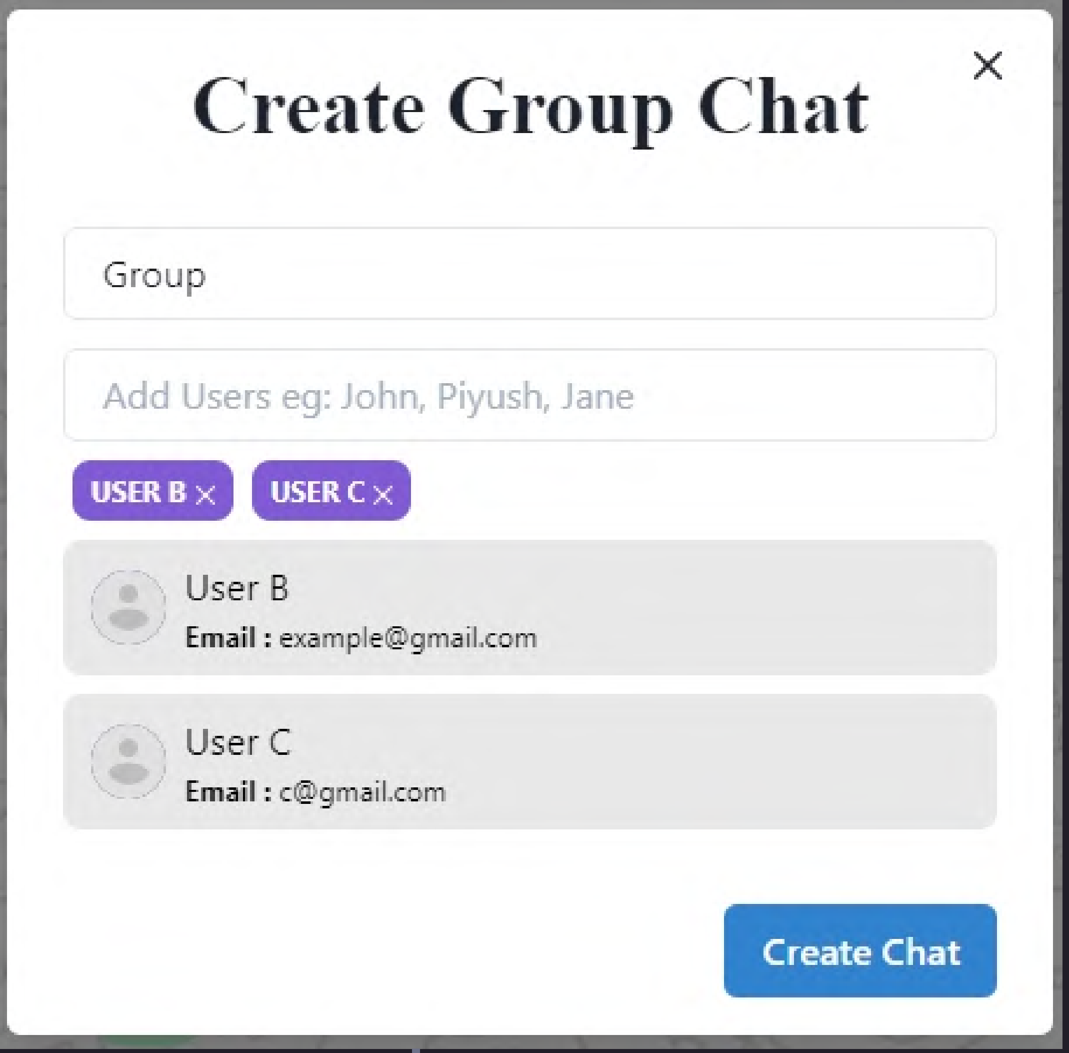
how are you??

Try Pitch

Group Chat Sequence Diagram



Try Pitch



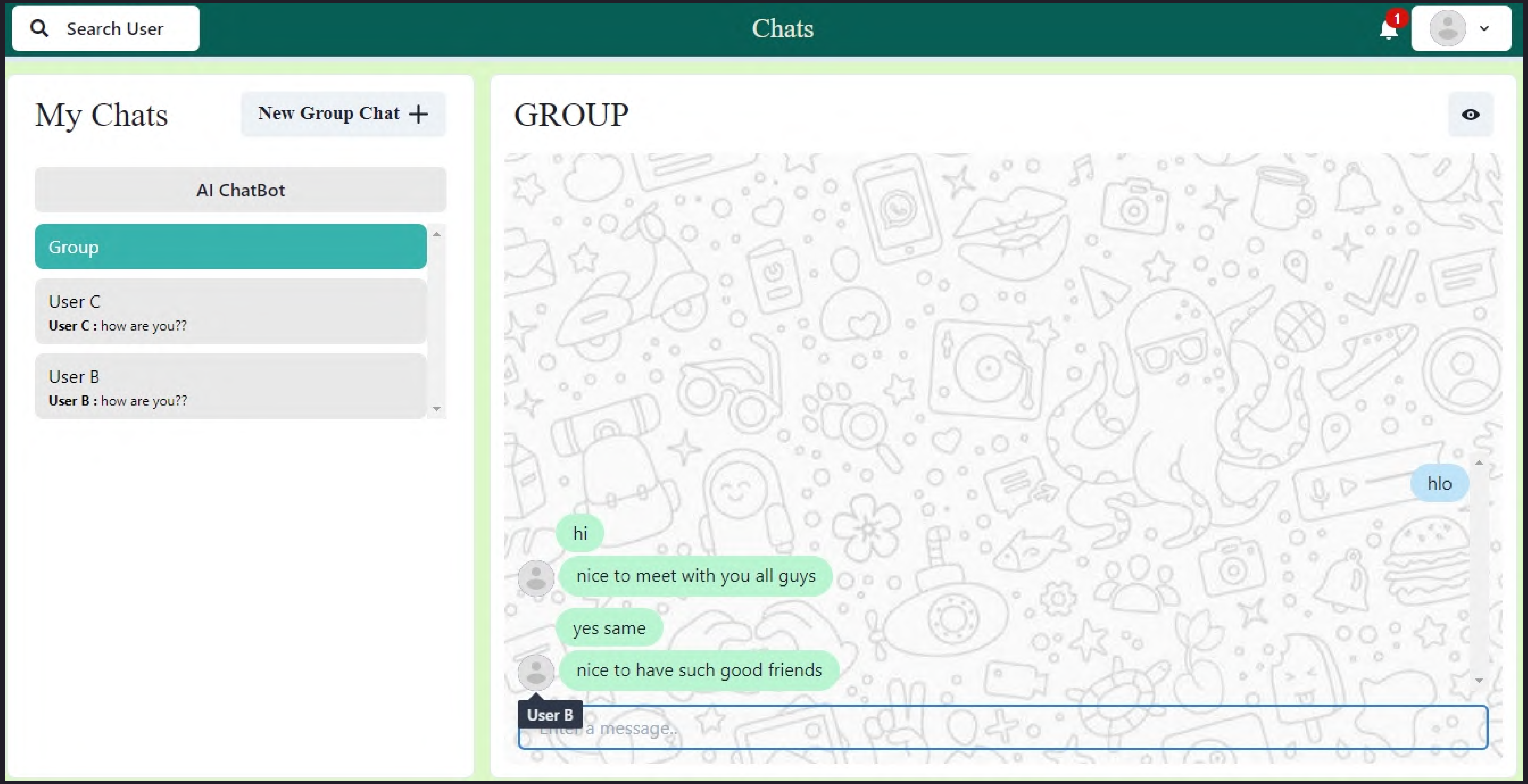
Create Group Chat

Group

Add Users eg: John, Piyush, Jane

User B
Email : example@gmail.com

User C
Email : c@gmail.com



Group

User C
User C : how are you??

User B
User B : how are you??

hlo

hi

nice to meet with you all guys

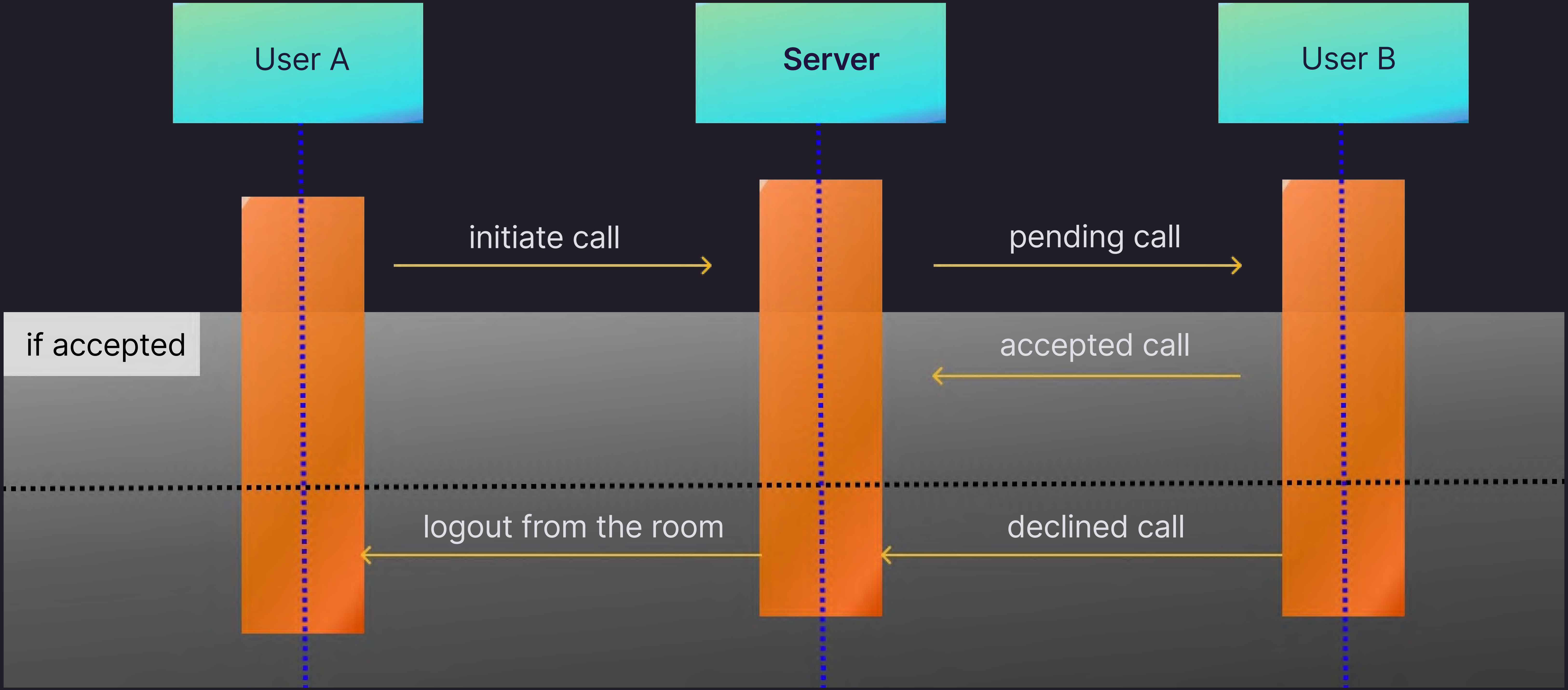
yes same

nice to have such good friends

After creating a group you can start chat with group members.

Try Pitch

Voice/Video Call Sequence Diagram



User A

Server

User B

initiate call

pending call

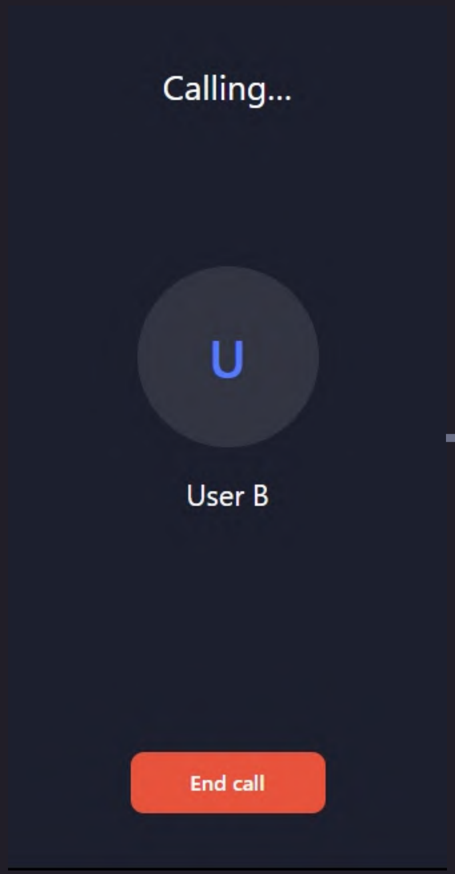
if accepted

accepted call

logout from the room

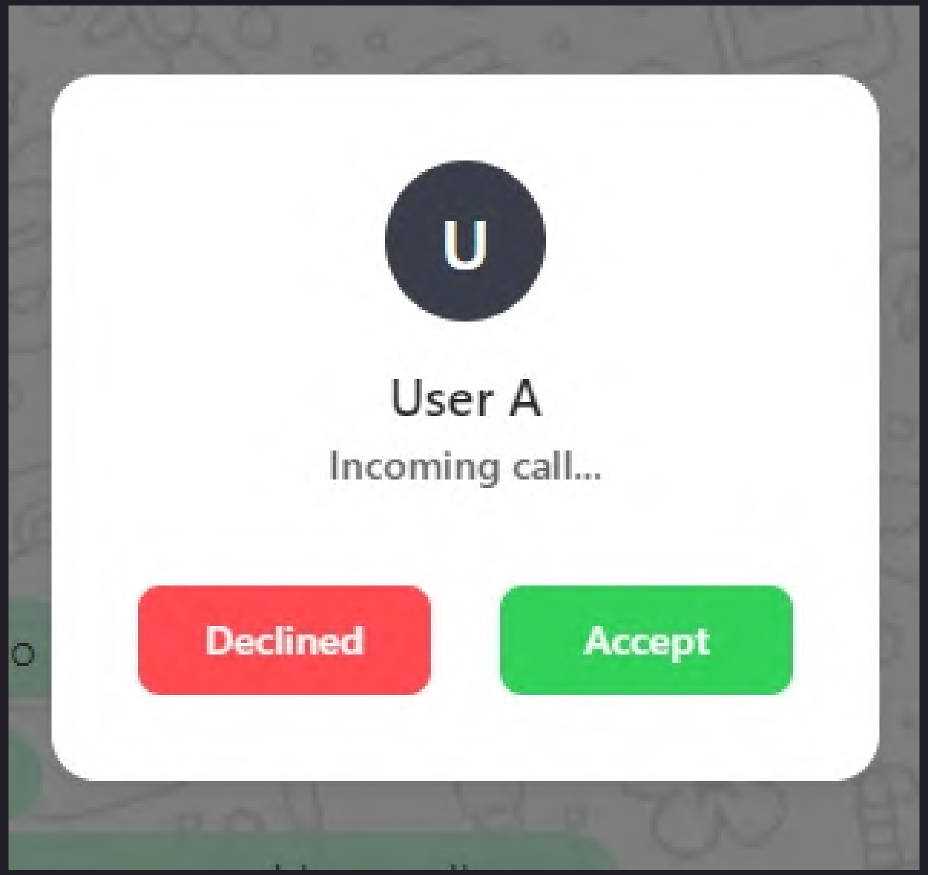
declined call

Try Pitch



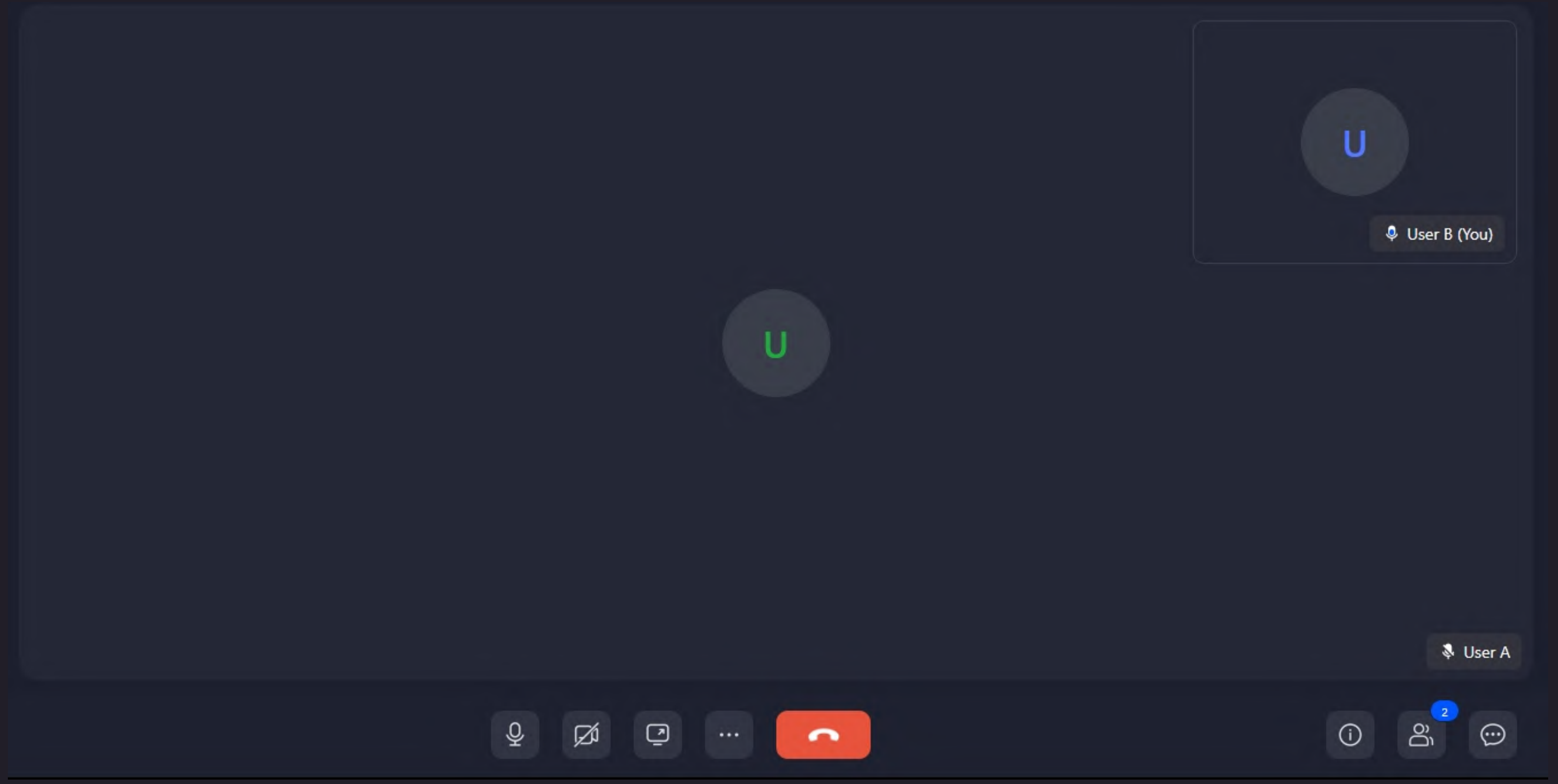
User A calling User B

Sending a call invitation to User B



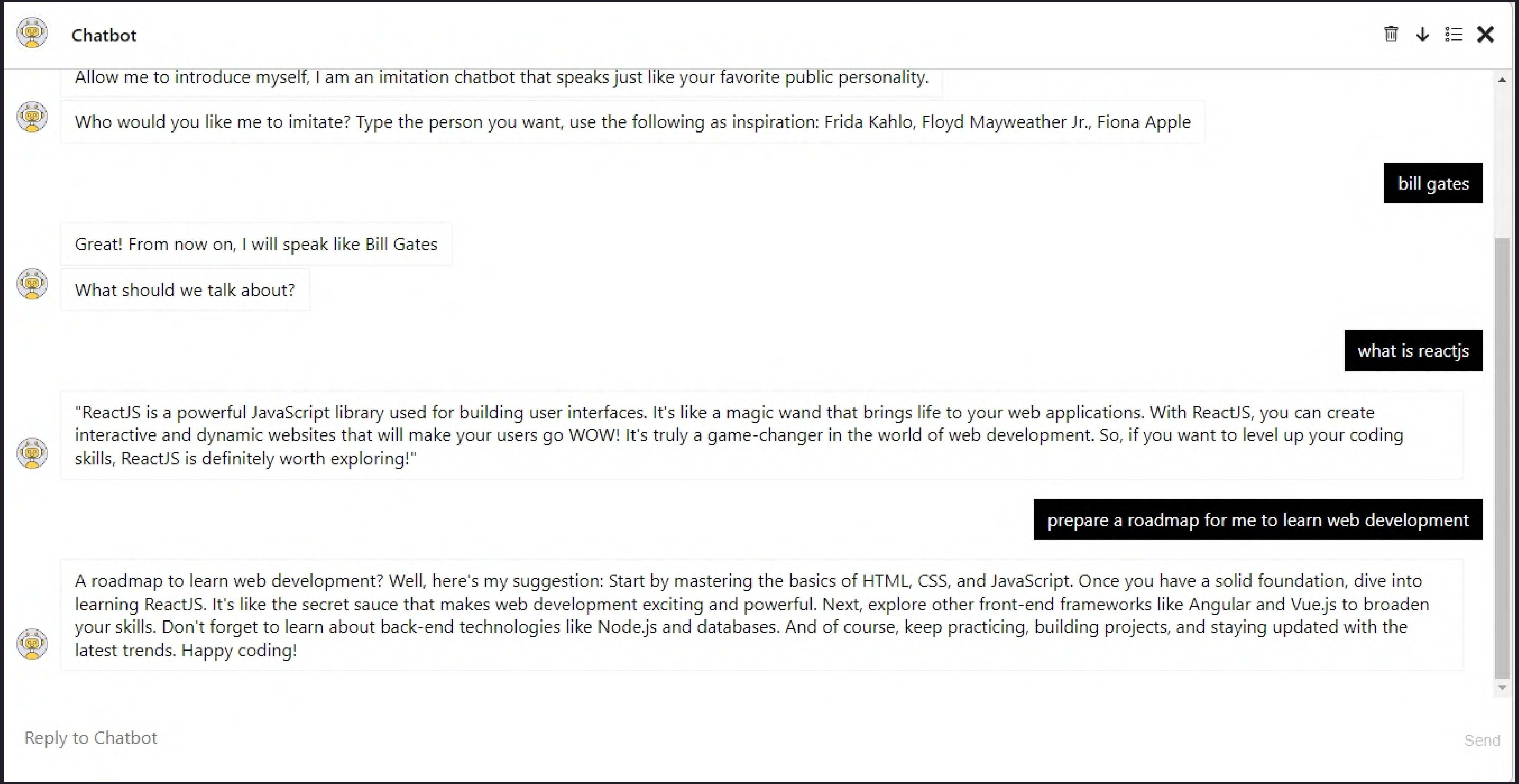
Call invitation to User B from User A

User B accepts the call invitation of User A



Try Pitch

AI Chat Bot



Allow me to introduce myself, I am an imitation chatbot that speaks just like your favorite public personality.

Who would you like me to imitate? Type the person you want, use the following as inspiration: Frida Kahlo, Floyd Mayweather Jr., Fiona Apple

bill gates

Great! From now on, I will speak like Bill Gates

What should we talk about?

what is reactjs

"ReactJS is a powerful JavaScript library used for building user interfaces. It's like a magic wand that brings life to your web applications. With ReactJS, you can create interactive and dynamic websites that will make your users go WOW! It's truly a game-changer in the world of web development. So, if you want to level up your coding skills, ReactJS is definitely worth exploring!"

prepare a roadmap for me to learn web development

A roadmap to learn web development? Well, here's my suggestion: Start by mastering the basics of HTML, CSS, and JavaScript. Once you have a solid foundation, dive into learning ReactJS. It's like the secret sauce that makes web development exciting and powerful. Next, explore other front-end frameworks like Angular and Vue.js to broaden your skills. Don't forget to learn about back-end technologies like Node.js and databases. And of course, keep practicing, building projects, and staying updated with the latest trends. Happy coding!

How to start the project in localhost?

Since we've used the MERN stack to build this application, you can start the backend by navigating to the main folder, i.e., ChatApp, and then run the '**npm start**' command to launch the backend on port 5000.
Next, navigate to the frontend folder to start the frontend with the '**npm start**' command on port 3000.
You can log in to our page using guest credentials as 'User A,' or you can also log in by providing the following credentials:
Email id: example@gmail.com
Password: 123.
This will log you in as 'User B.'

After logging in, you can search for 'User B' if you've logged in as 'User A' and initiate a chat with them. Additionally, you'll be able to use various features, including audio/video calling, chatting with users, interacting with our AI chatbot, and participating in group chats.
To join a group chat, you need to sign up with a different user, as the minimum group size is 3, and currently, our database only contains two users: 'User A' and 'User B'.

Try Pitch

Dependencies and Libraries

Front-end

1. Chakra-ui
2. Zegocloud
3. Axios
4. Framer motion
5. React-chips
6. React-lottie
7. React-notification-badge
8. React-router-dom
9. React-scrollable-feed
10. Socket.io-client
11. Styled-components
12. Web-vitals

Backend

1. Bcryptjs
2. Colors
3. Dotenv
4. Express
5. Express-async-handler
6. Json webtoken
7. Mongoose
8. Nodemon
9. Socket.io

Try Pitch

Backend

1. **bcryptjs**:
   Used for securely hashing and salting passwords before storing them in a database. It's essential for user authentication and security.
2. **colors**:
   This is a simple library used for adding colored text and backgrounds to the console output. It's handy for improving the readability of server logs during development.
3. **Dotenv**:
   Dotenv is used for loading environment variables from a .env file into your application's process environment. It's used to keep sensitive information like API keys, database credentials, and other configuration settings separate from your code.
4. **Express**:
   Express is a popular web application framework for Node.js. It simplifies the process of building web applications and APIs by providing a set of features and middleware for routing, handling requests, and more.
5. **Express-async-handler**:
   This is a middleware library that simplifies error handling for asynchronous routes in Express. It allows you to handle errors more gracefully in your route handlers.

Try Pitch

6. **Jsonwebtoken**:
   Jsonwebtoken (JWT) is used for user authentication and authorization. It's a compact and self-contained way to securely transmit information between parties as a JSON object. In web development, JWTs are commonly used for creating and validating authentication tokens.
7. **Mongoose**:
   Mongoose is an Object Data Modeling (ODM) library for MongoDB and Node.js. It provides a structured way to interact with MongoDB, making it easier to work with MongoDB databases, define schemas, and create models.
8. **Nodemon**:
   Nodemon is a tool that helps in the development process by monitoring for changes in your Node.js application and automatically restarting the server when changes are detected. This is particularly useful during development to speed up the testing and debugging process.
9. **Socket.io**:
   Socket.io is a library that enables real-time, bidirectional communication between clients and servers. It's often used for building chat applications, online games, and other applications where real-time updates are crucial.

Try Pitch

Front-end

1. **Chakra UI**:
   Chakra UI is a popular component library for building accessible and customizable user interfaces in React. It provides a set of pre-designed components and styles to accelerate the frontend development process.
2. **Zegocloud**:
   Zegocloud that simplifies the integration of voice and video call features into web applications.
3. **Axios**:
   Axios is a popular JavaScript library used for making HTTP requests from the browser. It simplifies working with REST APIs and handling responses in a convenient way.
4. **Framer Motion**:
   Framer Motion is a library for adding animations and motion to React components. It allows you to create fluid and interactive user interfaces with animations.
5. **React-chips**:
   React-chips is a library for implementing chip input fields in React applications. Chip inputs are often used for tagging and selecting multiple items.
6. **React-lottie**:
   React-lottie is used for integrating Lottie animations into React applications. It's a popular choice for adding engaging animations to web interfaces.

Try Pitch

7. **React-notification-badge**:
   This library likely provides components for adding notification badges to your React application, making it easier to display and manage notifications.
8. **React-router-dom**:
   React-router-dom is a library for handling routing in React applications. It allows you to create multi-page applications with URL navigation and dynamic rendering of components based on the route.
9. **React-scrollable-feed**:
   React-scrollable-feed is likely used for implementing scrollable lists or feeds in React applications, which is common in social media, chat, and news applications.
10. **Socket.io-client**:
    Socket.io-client is the client-side library for Socket.io, which enables real-time communication between the frontend and a server. It's commonly used for implementing real-time features like chat, notifications, and live updates.
11. **Styled-components**:
    Styled-components is a library for styling React components using tagged template literals. It allows you to write CSS directly in your JavaScript code, making it easier to create and manage component-specific styles.

Try Pitch

12. **Web-vitals**:
    Web-vitals is a library or tool for measuring and monitoring web performance metrics. It helps you track and improve aspects of web page performance, such as loading times, interactivity, and visual stability.

Try Pitch

Thank You

Try Pitch