
CONVO FUSION

A project report submitted in partial fulfilment
of the requirement for the degree of

Bachelor of Science

Computer Science

By:

Mr. Sihan Shaikh

Mr. Shreyash Devali

Mr. Devarsh Naik

Mr. Abhishek Johnson

Mr. Asher Pereira

Mr. Vedang Parab

Under the supervision of:

Mrs. Jyoti Kankonkar

(Assistant professor in Computer Science)

GOVERNMENT COLLEGE OF ARTS, SCIENCE & COMMERCE QUEPEM – GOA

DECLARATION BY CANDIDATES

Class: T.Y. B.Sc. (COMPUTER SCIENCE)

Roll no.	PR Number	Name	Signature
S21-613	202112134	Sihan Shaikh	
S21-626	202112174	Shreyash Devali	
S21-641	202112162	Devarsh Naik	
S21-645	202112843	Abhishek Johnson	
S21-606	202112180	Asher Pereira	
S21-646	202112179	Vedang Parab	

CERTIFICATE

.

This is to certify that the project work titled

"Convo Fusion"

has been successfully carried out by

Mr. Sihan Shaikh

Mr. Shreyash Devali

Mr. Devarsh Naik

Mr. Abhishek Johnson

Mr. Asher Pereira

Mr. Vedang Parab

Under the guidance of **Mrs. Jyoti Kankonkar**, as a part of the curriculum for the degree course in Bachelor of Science in Computer Science during the academic year 2023-24

Date:

Head of the Department:

Project Guide:

Principal

External Examiner:

ACKNOWLEDGEMENT

The completion of any interdisciplinary project depends upon cooperation, coordination, and combined efforts of several sources of knowledge.

We are extremely grateful to our Principal **Dr. Joydeep Bhattacharjee,** for allowing us to pursue this project.

We express our special gratitude to **Prof. Philip Rodrigues**, HOD of the Computer Science department, for providing us guidance to complete all the formalities related to the project within the deadline.

We are grateful to our project guide, **Mrs. Jyoti Kankonkar**, for her willingness to give us important advice and directions whenever we approached her with a problem. We are thankful to her for providing immense guidance for this project.

We would be failing in our duties if we did not thank our colleagues who have constantly been a source of motivation for us throughout this project work.

Finally, we thank our parents whose love and affection were the energy behind us throughout this alluring experience.

CONTENT

INTRODUCTION

In today's digital age, communication plays a pivotal role in connecting individuals across the globe. The advent of mobile applications has revolutionized the way we interact, enabling seamless and instant communication regardless of geographical barriers. Among these, chatting apps have emerged as a dominant force, offering users a platform to exchange messages, share media, and connect in real time.

This report delves into the design, development, and features of a modern chatting app, aiming to provide a comprehensive understanding of its functionality and user experience. The app integrates innovative technologies such as AI chatbots, music-sharing APIs, and custom image generation to enhance user engagement and create a dynamic messaging environment.

Through an exploration of its key components, user interface, and integration of external APIs, this report seeks to highlight the app's unique features and capabilities. Additionally, it examines the challenges faced during the development process and the strategies employed to overcome them, providing valuable insights for future app development projects.

Overall, this report serves as a detailed analysis of a modern chatting app, showcasing its potential to revolutionize the way we communicate and interact in the digital age.

1.1 Project Overview

The proposed project is to develop a chatting app that seamlessly integrates music streaming, AI chat capabilities, and image-generating AI services. Key features include user authentication, individual and group chat functionality, Spotify integration for music streaming, ChatGPT API for AI chat, Dall-e API for image generation, search functionality, customization options, and a notification system.

Existing systems like WhatsApp, WeChat, and Snapchat are popular competitors in the messaging app market. WhatsApp offers unlimited calling and group texting but raises security concerns due to potential integration with Facebook. WeChat provides quick registration and high-quality voice/video calling but has a less intuitive interface. Snapchat excels in capturing real-time moments with its visual approach but faces criticism for a confusing interface.

In summary, the proposed app aims to combine the strengths of existing messaging apps with additional features like music streaming and AI capabilities to provide users with a unique and engaging chatting experience.

1.2 Existing systems

WhatsApp stands out as a widely-used chat app for both Android and iOS, boasting over a billion users. It offers numerous advantages including unlimited calling, free app-to-app messaging, group texting, and video calling features.

However, concerns regarding future security arise due to potential integration with Facebook Messenger and Instagram, alongside the necessity for strong internet connectivity for clear video calls. WeChat, originating from China in 2011, has evolved into a global social media platform with hundreds of millions of users worldwide. It offers quick registration via mobile numbers or Facebook logins, free high-quality voice and video calling, and a unique "Look around" feature facilitating new connections. Nevertheless, its interface lacks the smoothness and intuitiveness of other major messaging apps.

Snapchat, on the other hand, excels in capturing real-time moments effectively, offering instant access to camera mode, a plethora of lenses and filters, and the ability to create engaging stories. However, its recent interface changes have led to user confusion and hiding of stories from friends, family, and customers.

1.3 Purpose

The purpose of our chatting app with APIs like Spotify, ChatGPT, and DALL-E image-generating API would be to provide users with a comprehensive and innovative messaging experience. By integrating the DALL-E image-generating API, users can create and share unique and personalized images within their chats, adding a creative and visual element to their conversations.

The app aims to offer a multi-faceted communication platform where users can not only chat with each other, share music from Spotify, and engage in AI-powered conversations through ChatGPT but also create and share custom images generated by DALL-E. This feature enhances the overall user experience by allowing for more creative expression and visual communication within the app.

Overall, the app's purpose is to offer a dynamic and engaging messaging experience that combines music, AI chat capabilities, custom image generation, and traditional messaging features in one integrated platform. This app targets users who are looking for a more interactive, creative, and personalized way to communicate with others while enjoying various multimedia elements.

TECHNOLOGY & SERVICES IMPLEMENTED

2.1 Services Implemented

- One-to-one chat
- Group Chat
- Audio Calling
- Video Calling
- Screen Sharing
- Authentication (Login/Register/resetting password)
- Media Transfer
- AI bots
- AI Chatbot using Spotify AI
- AI Chatbot using OpenAI

1. **One-to-One Chat:** One-to-one chat allows users to communicate privately with another user. It typically includes features such as text messaging, emojis, file sharing, and voice messages. Users can have personal conversations, share media, and engage in real-time communication without other users being able to see the conversation.

2. **Group Chat** Group chat enables multiple users to communicate in a single chat room. Users can create groups, add or remove members, and send messages that are visible to all group members. Group chat is useful for coordinating with teams, organizing events, or simply chatting with a group of friends.

3. **Audio Calling:** Audio calling allows users to make voice calls to other users. It uses WebRTC(Web Real-Time Communication) technology to transmit audio data over the Internet, enabling users to have real-time conversations without the need for a traditional phone line. Audio calling is useful for situations where users prefer to communicate verbally.

4. **Video Calling:** Video calling allows users to have face-to-face conversations over the Internet. It uses video and audio data to enable users to see and hear each other in real time. Video calling is useful for situations where visual communication is important, such as business meetings, catching up with friends and family, or remote interviews.

5. **Screen Sharing Features:** Screen sharing allows users to share their screen with others during a chat session. This feature can be useful for collaboration, troubleshooting technical issues, or simply showing something to another user. For example, users can share their screens to show a presentation, demonstrate how to use a software application, or browse the web together.

6. **Authentication:** (Log in/Register/Resetting) Authentication is the process of verifying the identity of a user. It typically involves a login and registration system, where users create an account with a username and password to access

the app's features securely. Authentication ensures that only authorized users can access the app and protects user data from unauthorized access.

7. **Authentication:** (Login/Register) Authentication is the process of verifying the identity of a user. It typically involves a login and registration system, where users create an account with a username and password to access the app's features securely. Authentication ensures that only authorized users can access the app and protects user data from unauthorized access.

8. **AI Chatbot using Spotify AI:** You can leverage Spotify AI to create an AI chatbot that understands natural language and can engage in conversations with users. This chatbot can help users discover music, create playlists, and get recommendations based on their preferences. For example, users can ask the chatbot to suggest songs based on their mood or music taste, and the chatbot can respond with personalized recommendations.

9. **AI Chatbot using OpenAI:** OpenAI provides powerful natural language processing capabilities that can be used to create an AI chatbot for your app. This chatbot can understand and respond to user queries, provide information, and assist users with various tasks. For example, users can ask the chatbot for the weather forecast, news updates, or general information, and the chatbot can provide relevant answers.

2.2 Technology

- Frontend: React v18, Mui v5, Redux
- Backend: NodeJs, ExpressJS, MongoDB, Mongoose, Socket.io, ZEGOCLOUD WebRTC API
- Deployment Stack: AWS EC2, S3

MERN

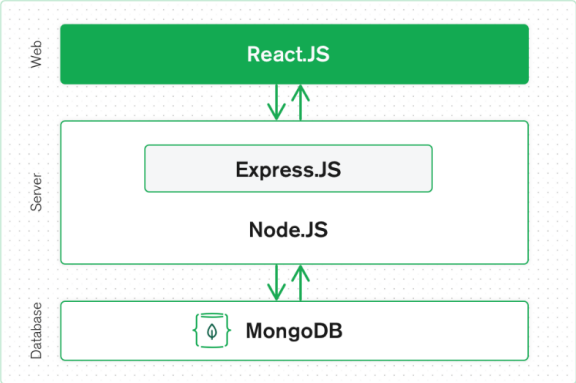
MERN stands for MongoDB, Express, React, Node, after the four key technologies that make up the stack.

- MongoDB — document database
- Express(.js) — Node.js web framework
- React(.js) — a client-side JavaScript framework
- Node(.js) — the premier JavaScript web server

Express and Node make up the middle (application) tier. Express.js is a server-side web framework, and Node.js is a popular and powerful JavaScript server platform. Regardless of which variant you choose, ME(RVA)N is the ideal approach to working with JavaScript and JSON, all the way through.

How does the MERN stack work?

The MERN architecture allows you to easily construct a three-tier architecture (front end, back end, database) entirely using JavaScript and JSON.



React.js front end

The top tier of the MERN stack is React.js, the declarative JavaScript framework for creating dynamic client-side applications in HTML. React lets you build up complex interfaces through simple components, connect them to data on your back-end server, and render them as HTML.

React's strong suit is handling stateful, data-driven interfaces with minimal code and minimal pain, and it has all the bells and whistles you'd expect from a modern web framework: great support for forms, error handling, events, lists, and more.

Express.js and Node.js server tier

The next level down is the Express.js server-side framework, running inside a Node.js server. Express.js bills itself as a "fast, unopinionated, minimalist web framework for Node.js," and that is indeed exactly what it is. Express.js has powerful models for URL routing (matching an incoming URL with a server function), and handling HTTP requests and responses.

By making XML HTTP Requests (XHRs) or GETs or POSTs from your React.js front end, you can connect to Express.js functions that power your application. Those functions, in turn, use MongoDB's Node.js drivers, either via callbacks or promises, to access and update data in your MongoDB database.

MongoDB database tier

If your application stores any data (user profiles, content, comments, uploads, events, etc.), then you're going to want a database that's just as easy to work with as React, Express, and Node.

That's where MongoDB comes in: JSON documents created in your React.js front end can be sent to the Express.js server, where they can be processed and (assuming they're valid) stored directly in MongoDB for later retrieval. Again, if you're building in the cloud, you'll want to look at Atlas. If you're looking to set up your own MERN stack, read on!

Our Usage of these technologies

Material-UI v5

Material-UI (Mui) v5 is a popular React UI framework that provides pre-designed components following Google's Material Design principles. Here's some key information about Mui v5:

1. Component Library: Material-UI offers a wide range of reusable and customizable components such as buttons, cards, inputs, dialogs, and more, which help you build responsive and visually appealing user interfaces.

2. Customization: Mui components are highly customizable, allowing you to easily change colors, typography, spacing, and other visual properties to match your design requirements.

3. Utility Functions: Material-UI provides utility functions like `make styles` and `styled` to help you style your components using CSS-in-JS, making it easier to manage styles within your React components.

4. Theming: Mui v5 introduces a new theming system that allows you to define global styles and customize the default theme to create a consistent design language throughout your application.

5. Accessibility: Material-UI emphasizes accessibility, ensuring that components are usable by everyone, including those with disabilities. Components are designed to meet WCAG standards.

6. Performance: Material-UI focuses on performance optimization, providing features like lazy loading and tree shaking to reduce bundle size and improve loading times.

7. Community and Support: Material-UI has a large and active community, with extensive documentation, tutorials, and examples to help you get started and solve common problems.

8. Compatibility: Material-UI v5 is designed to work with React 16.8 and later versions, and it provides support for both modern browsers and older versions, including Internet Explorer 11.

Overall, Material-UI v5 is a powerful and flexible UI framework for React developers, offering a rich set of components and features to streamline the development of modern web applications.

In ConvoFusion, we have majorly used Material UI components to render our application design to make our frontend elements.

Redux

Redux is a predictable state container for JavaScript apps, most commonly used with libraries like React or Angular for managing application states. Here's some key information about Redux:

1. State Management: Redux provides a centralized store to manage the entire state of your application. This makes it easier to maintain and manage the state, especially in large or complex applications.

2. Predictable State Changes: Redux uses a concept called "reducers" to describe state mutations. Reducers are pure functions that take the previous state and an action and return the next state. This makes state changes predictable and helps in debugging.

3. Unidirectional Data Flow: Redux follows a unidirectional data flow pattern, where data flows in a single direction from the store to the components. This helps in maintaining a clear and consistent data flow throughout the application.

4. Actions and Action Creators: Actions are payloads of information that describe the type of change to be made to the state. Action creators are functions that create actions. They are used to trigger state changes in Redux.

5. Reducers: Reducers are pure functions that take the current state and an action, and return the next state. They are responsible for handling state transitions in Redux.

6. Store: The store is a single source of truth for the state of your application. It holds the current state and provides methods to dispatch actions, subscribe to state changes, and access the state.

7. Middleware: Redux middleware provides a third-party extension point between dispatching an action, and the moment it reaches the reducer. It can be used for logging, crash reporting, asynchronous actions, and more.

8. Dev Tools: Redux Dev Tools is a powerful tool that allows you to inspect every action and state change in your application, making it easier to debug and understand how your application state evolves over time.

9. Community and Ecosystem: Redux has a large and active community, with a rich ecosystem of libraries and tools that extend its functionality and make it easier to integrate with other libraries and frameworks.

Overall, Redux is a powerful state management tool that helps you write applications that behave consistently, run in different environments, and are easy to test. It is especially useful for complex applications with a lot of state that needs to be shared across multiple components.

In ConvoFusion, we have implemented Redux to manage our states to make it run with optimization in most used environments.

Socket.IO

Socket.IO is a JavaScript library for real-time web applications. It enables real-time, bidirectional, and event-based communication between clients (usually web browsers) and a server. Here's some key information about Socket.IO:

1. Real-time Communication: Socket.IO enables real-time, two-way communication between clients and servers. This is especially useful for applications requiring instant updates or notifications, such as chat applications, live sports updates, or collaborative editing tools.

2. WebSocket Support: Socket.IO uses WebSocket protocol if available, which provides a persistent connection between the client and the server. This allows for efficient, low-latency communication compared to traditional HTTP requests.

3. Fallback Mechanism: Socket.IO includes a fallback mechanism that uses alternative transport protocols if WebSocket is not available. This ensures compatibility with older browsers or networks that block WebSocket connections.

4. Event-Based Communication: Socket.IO uses events to communicate between clients and servers. Clients can emit events to the server, and the server can emit events to clients, enabling a flexible and dynamic communication model.

5. Room Support: Socket.IO allows clients to join "rooms," which are virtual channels that can be used to broadcast messages to multiple clients at once. This is useful for implementing features like group chat or live updates for specific groups of users.

6. Scalability: Socket.IO can be scaled horizontally to handle large numbers of clients and connections. It supports clustering and load balancing, making it suitable for high-traffic applications.

Overall, Socket.IO is a powerful tool for building real-time web applications that require fast and efficient communication between clients and servers.

In ConvoFusion, we have integrated Socket.IO to mandate our real-time communication systems to mitigate any sort of latencies and to introduce flexible, event-driven communication.

Mongoose

Mongoose is an Object Data Modelling (ODM) library for MongoDB and Node.js. It provides a straightforward way to model your application data and interact with the MongoDB database. Here are some key points about Mongoose:

1. Schema Definition: Mongoose allows you to define schemas, which represent the structure of your documents in MongoDB. Schemas define the fields and their types, along with any validation rules.

2. Model Creation: Once you have defined a schema, you can create a model based on that schema. Models are constructor functions that allow you to create instances of your documents.

3. CRUD Operations: Mongoose provides methods for performing CRUD (Create, Read, Update, Delete) operations on your MongoDB database. These methods make it easy to interact with your data.

4. Validation: Mongoose allows you to define validation rules for your schemas. This ensures that your data meets certain criteria before it is saved to the database.

5. Middleware: Mongoose supports middleware functions, which are functions that are executed before or after certain operations (e.g., before saving a document). This allows you to add custom logic to your data operations.

6. Query Building: Mongoose provides a powerful query builder that allows you to build complex queries for retrieving data from your database.

7. Population: Mongoose supports population, which allows you to reference documents in other collections and automatically retrieve them when querying.

Overall, Mongoose simplifies the process of working with MongoDB in Node.js applications by providing a higher-level abstraction for data modeling and interaction.

In ConvoFusion, Mongoose ODM was used because of its abundance in helpful functions when it comes to the usage and maintenance of database in MongoDB.

WebRTC

WebRTC (Web Real-Time Communication) is a free, open-source project that provides web browsers and mobile applications with real-time communication capabilities via simple application programming interfaces (APIs). Here's some key information about WebRTC:

1. Real-Time Communication: WebRTC enables real-time communication (RTC) between browsers and mobile applications. It allows for peer-to-peer audio, video, and data sharing without the need for plugins or additional software.

2. Key Features:
- Audio and Video Calls: WebRTC supports high-quality audio and video calls between browsers and mobile devices.
- Data Sharing: It allows for the exchange of arbitrary data between peers, enabling use cases like file sharing and multiplayer gaming.
- Network and Audio/Video Quality Adaptation: WebRTC dynamically adjusts to network conditions to provide the best possible audio and video quality.
-Security: WebRTC uses encryption standards to ensure secure communication between peers.

3. Components:
- Media Stream: Represents a stream of media content, such as audio or video.
-RTC Peer Connection: Enables audio and video calling and data sharing between peers.
-RTC Data Channel: Provides bidirectional communication of arbitrary data between peers.

4. Use Cases:
-Video Conferencing: WebRTC is commonly used for real-time video conferencing applications.
- Live Streaming: It can be used to stream live audio and video content over the web.
- File Sharing: WebRTC's data channel allows for efficient file sharing between peers.
-Online Gaming: WebRTC enables real-time multiplayer gaming experiences.

5. Browser Support: Most modern web browsers, including Chrome, Firefox, Safari, and Edge, support WebRTC. Mobile platforms like Android and iOS also have WebRTC support.

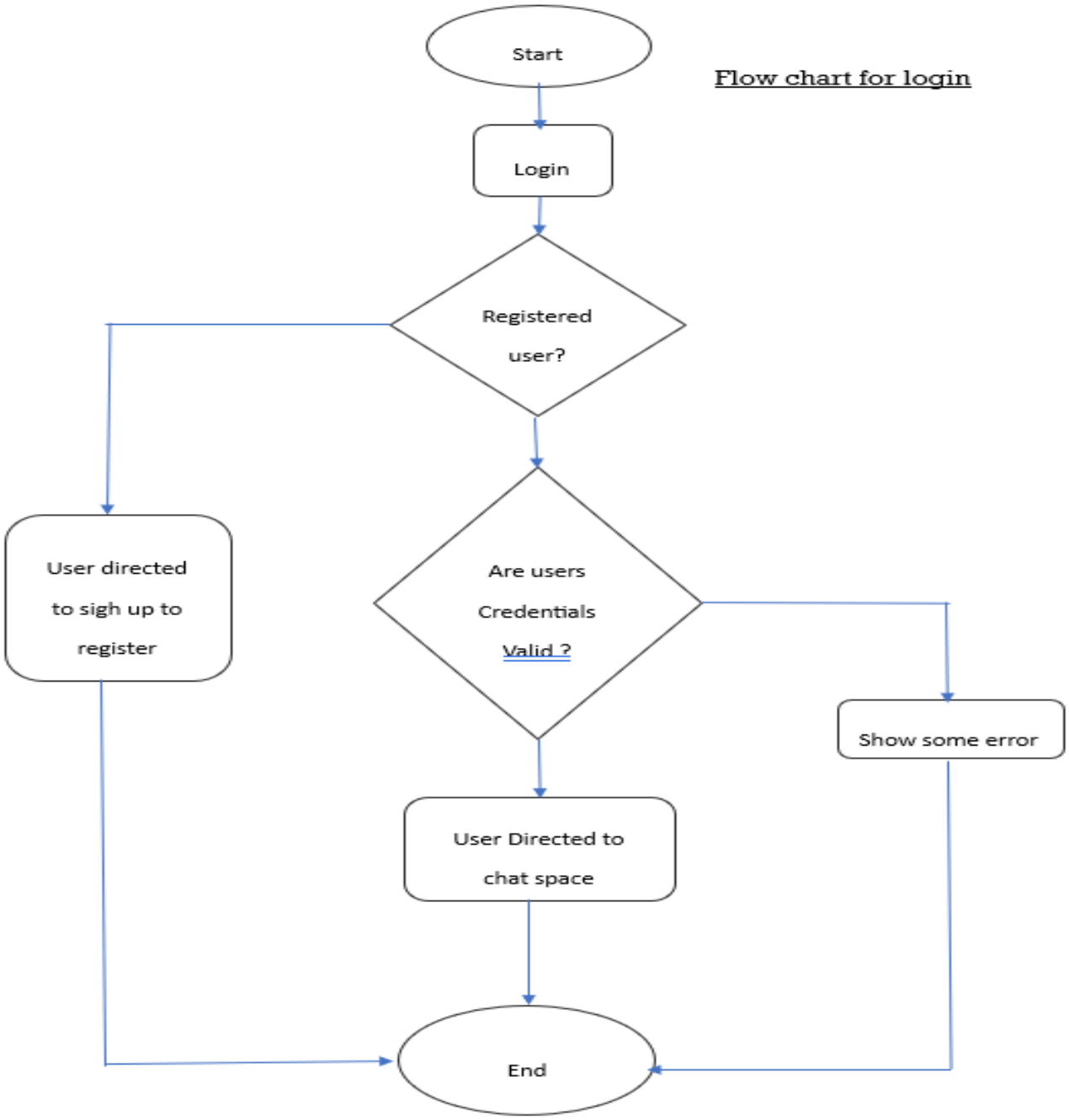
Overall, WebRTC is a powerful technology that simplifies the development of real-time communication applications on the web and mobile platforms. It is widely used for a variety of applications that require low latency, and high-quality audio, video, and data communication.

In ConvoFusion, we chose to use WebRTC because of its prevalence in the realm of communication in providing a real-time secure connection to mandate voice calls, video calls, file transferring and for screen sharing features.

ANALYSIS & DESIGN

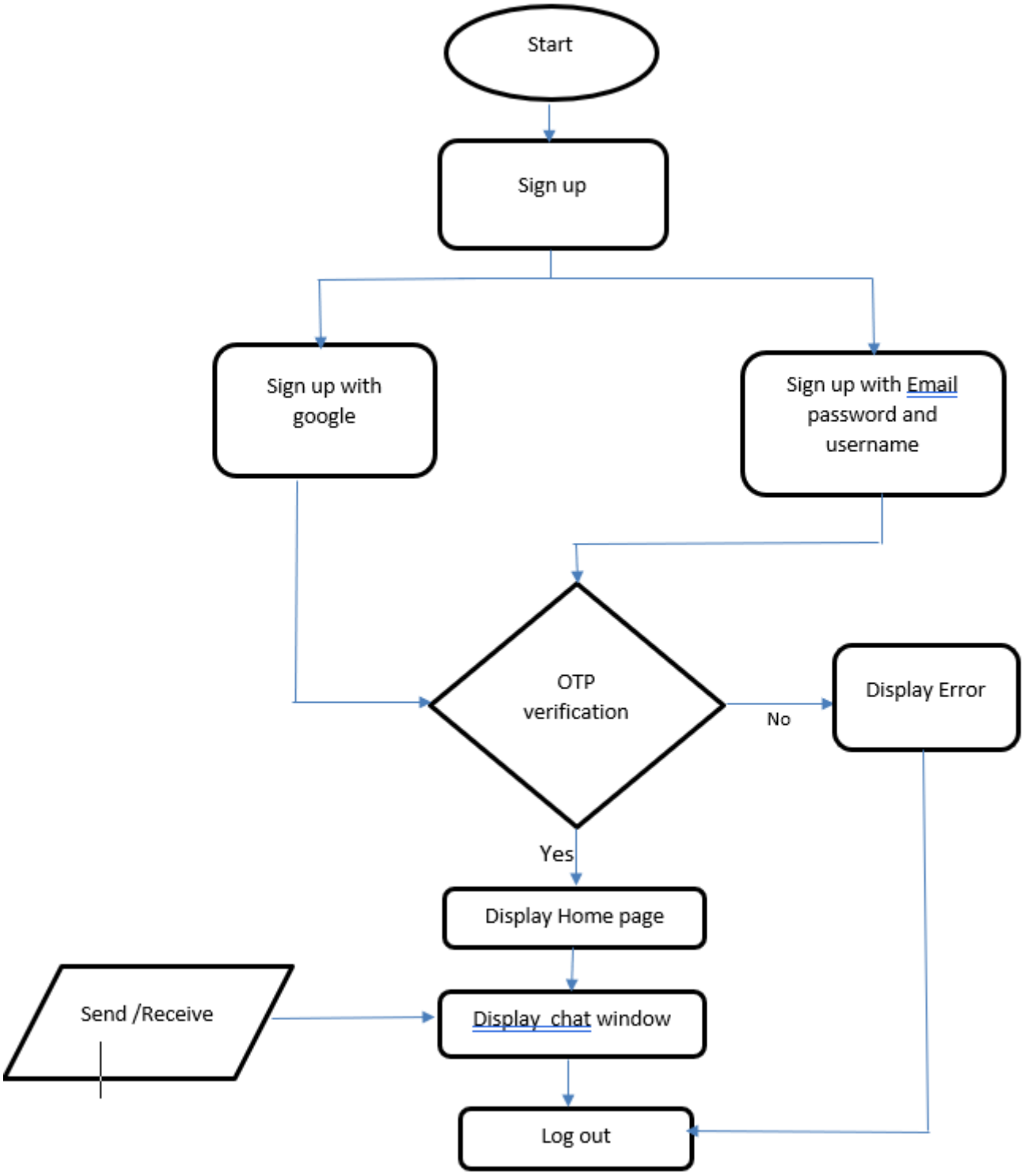
3.1 Log-in

Below is the flowchart showing the flow of the Convofusion app for signing in, once a user is authorized, he can directly sign in whenever he wants to use the Convofusion app using his valid credentials.

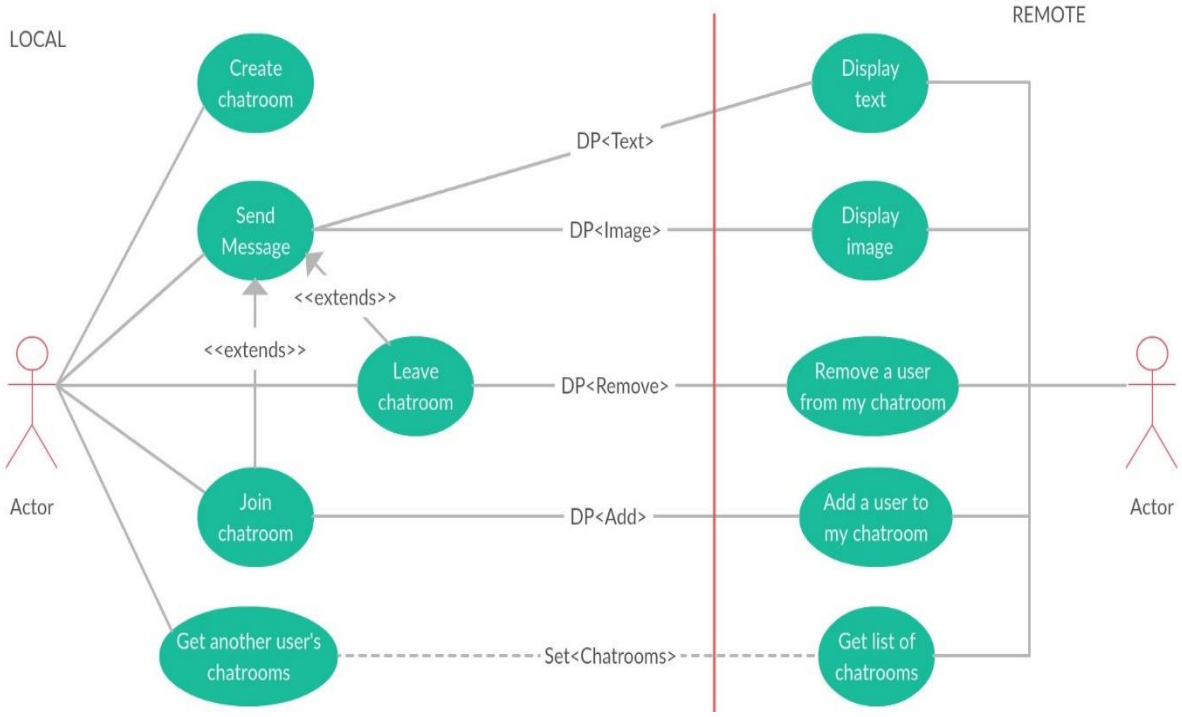


3.2 Sign-Up

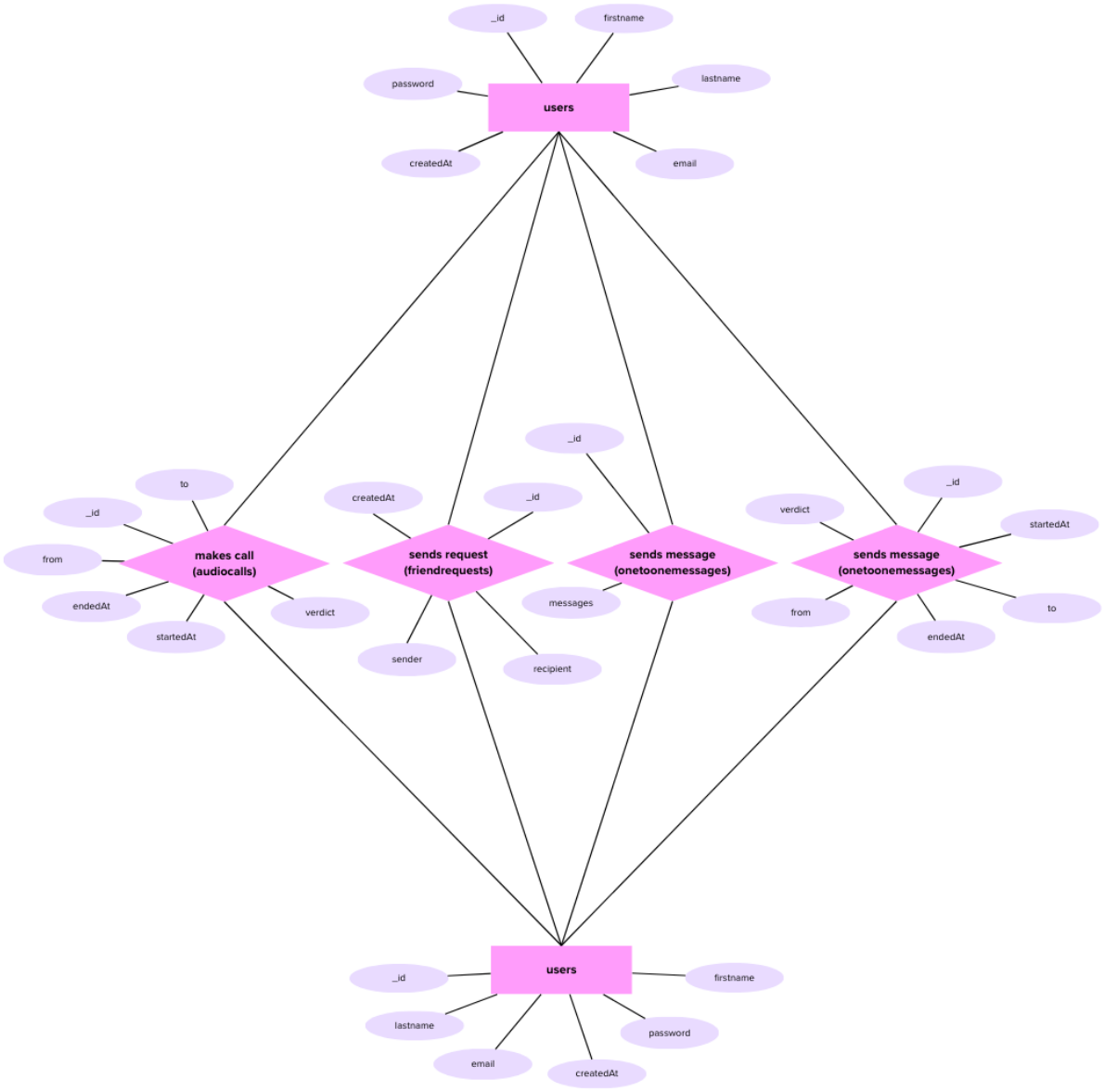
In the flow chart below, we can see how the flow of the sign-up page works in convofusion. During a Sign-Up email, a username, and password is required to register as an authorized user.



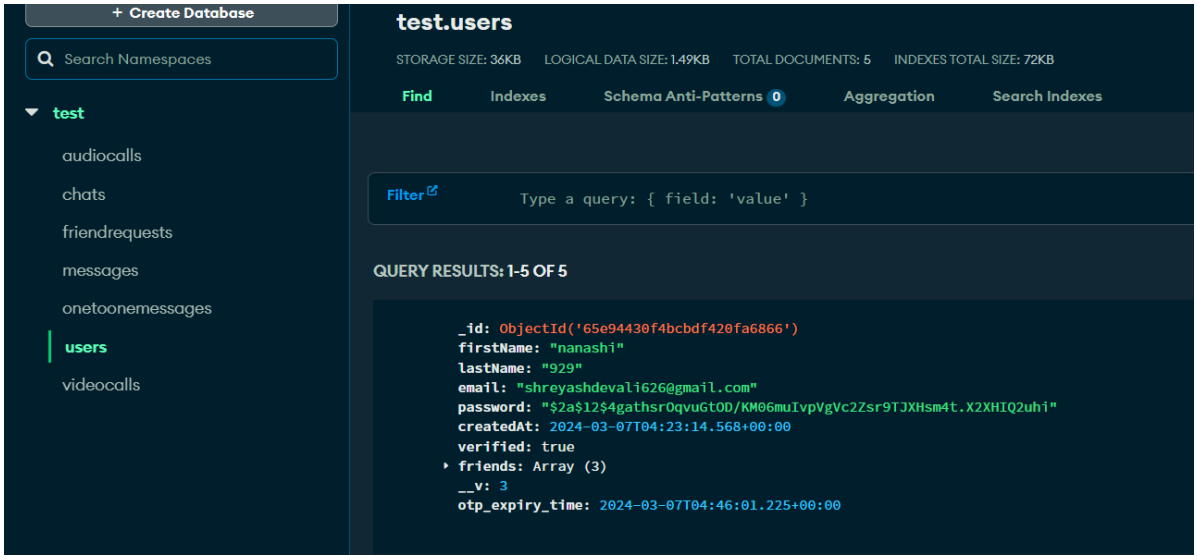
3.3 Use Case Diagram

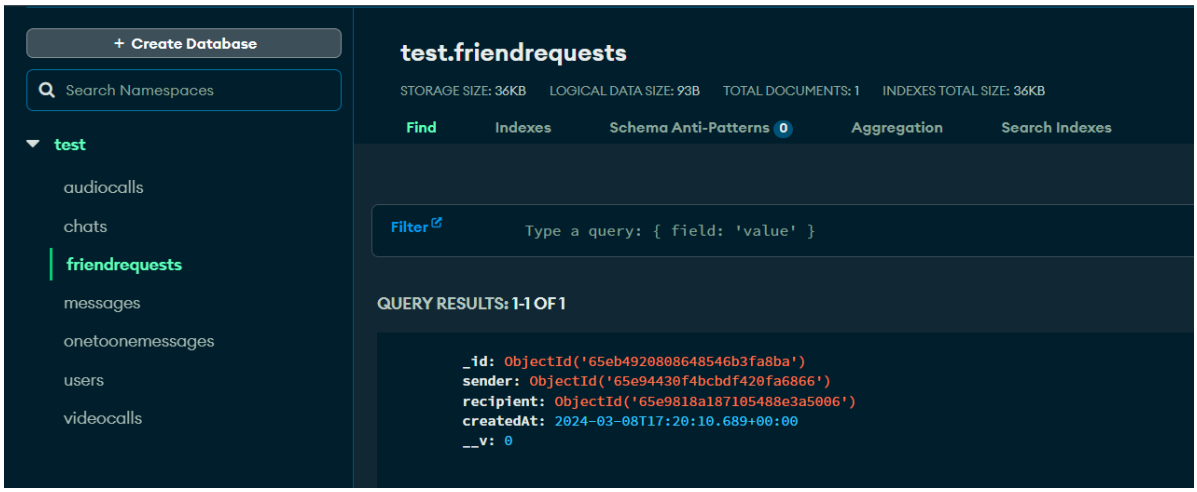


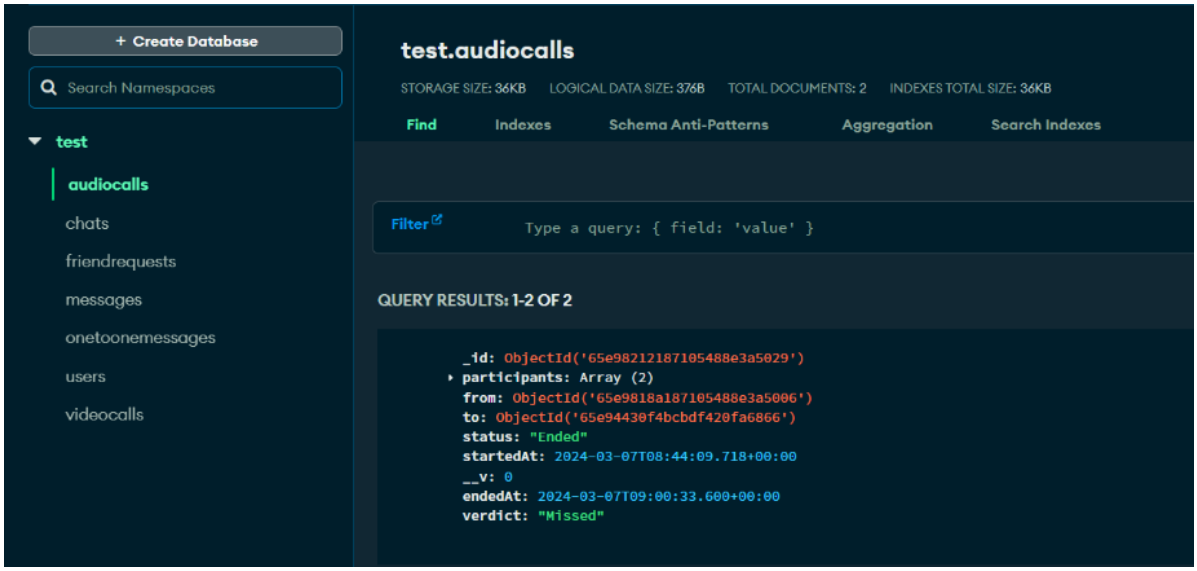
3.4 ER Diagram

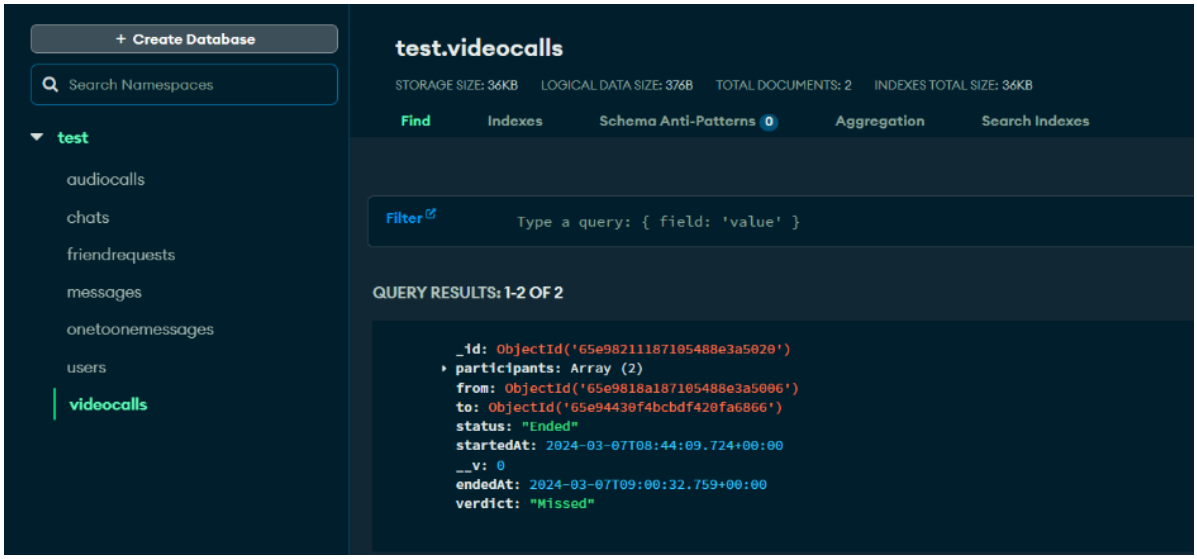


3.5 Database samples






+ Create Database
Search Namespaces
test
audiocalls
chats
friendrequests
messages
onetoonemessages
users
videocalls
test.audiocalls
STORAGE SIZE: 36KB
LOGICAL DATA SIZE: 376B
TOTAL DOCUMENTS: 2
INDEXES TOTAL SIZE: 36KB
Find
Indexes
Schema Anti-Patterns
Aggregation
Search Indexes
Filter
Type a query: { field: 'value' }
QUERY RESULTS: 1-2 OF 2
_id: ObjectId('65e98212187105488e3a5029')
participants: Array (2)
from: ObjectId('65e9818a187105488e3a5006')
to: ObjectId('65e94430f4bcbdf420fa6866')
status: "Ended"
startedAt: 2024-03-07T08:44:09.718+00:00
__v: 0
endedAt: 2024-03-07T09:00:33.600+00:00
verdict: "Missed"


+ Create Database
Search Namespaces
test
audiocalls
chats
friendrequests
messages
onetoonemessages
users
videocalls
test.videocalls
STORAGE SIZE: 36KB
LOGICAL DATA SIZE: 376B
TOTAL DOCUMENTS: 2
INDEXES TOTAL SIZE: 36KB
Find
Indexes
Schema Anti-Patterns 0
Aggregation
Search Indexes
Filter
Type a query: { field: 'value' }
QUERY RESULTS: 1-2 OF 2
_id: ObjectId('65e98211187105488e3a5020')
participants: Array (2)
from: ObjectId('65e9818a187105488e3a5006')
to: ObjectId('65e94430f4bcbdf420fa6866')
status: "Ended"
startedAt: 2024-03-07T08:44:09.724+00:00
__v: 0
endedAt: 2024-03-07T09:00:32.759+00:00
verdict: "Missed"

IMPLEMENTATION

4.1 Signing up as a user

To use ConvoFusion you must sign up and verify your account through the OTP that will be sent to the same email used in the signing-up process. Signing up is essential to use ConvoFusion. To make it easier for the user, ConvoFusion offers two methods to sign up.

- Using email and password
- Using Google

Both methods work flawlessly without any problem and it is up to the user to decide which one to use.

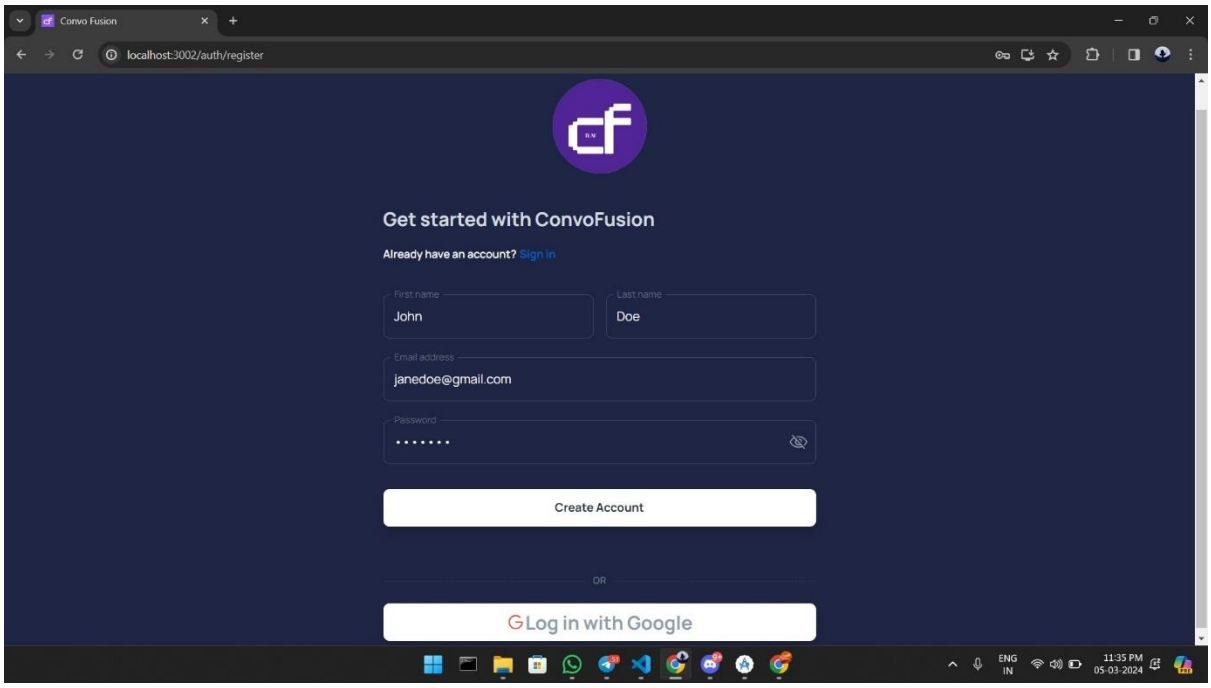


Fig 1

4.2 OTP Verification

In the Figure below, we can see the OTP verification page where the user has to put OTP sent to the user on Email ID which was provided during sign-up to verify the email ID

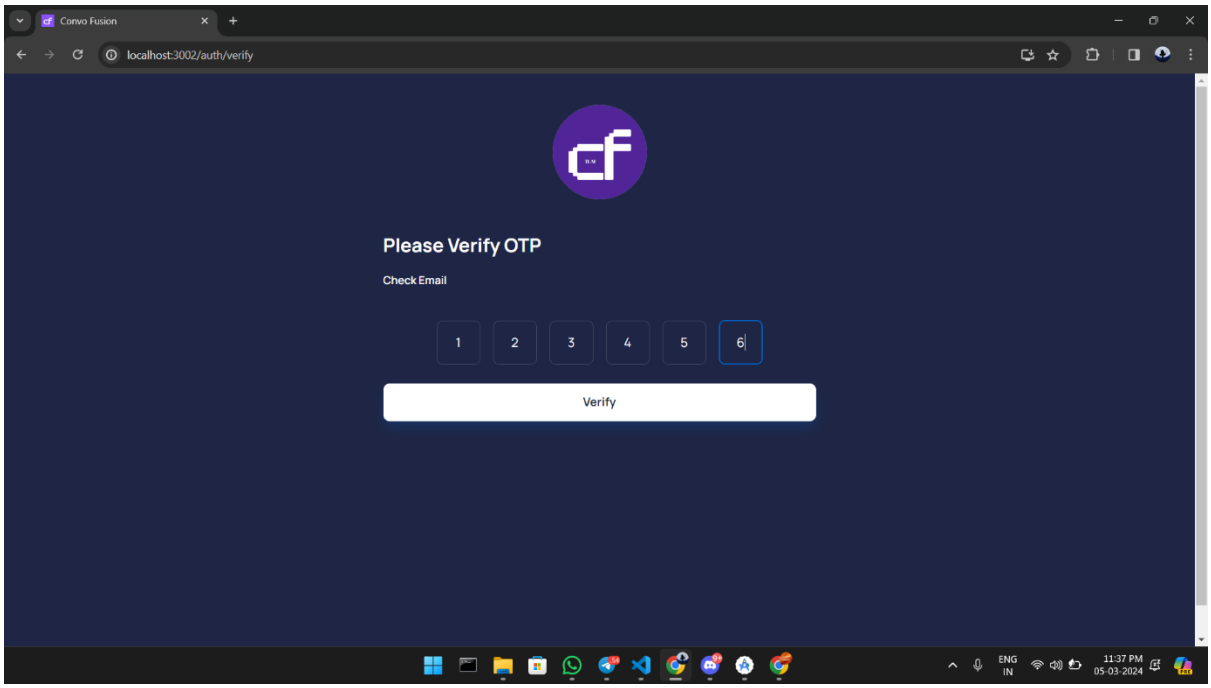


Fig 2

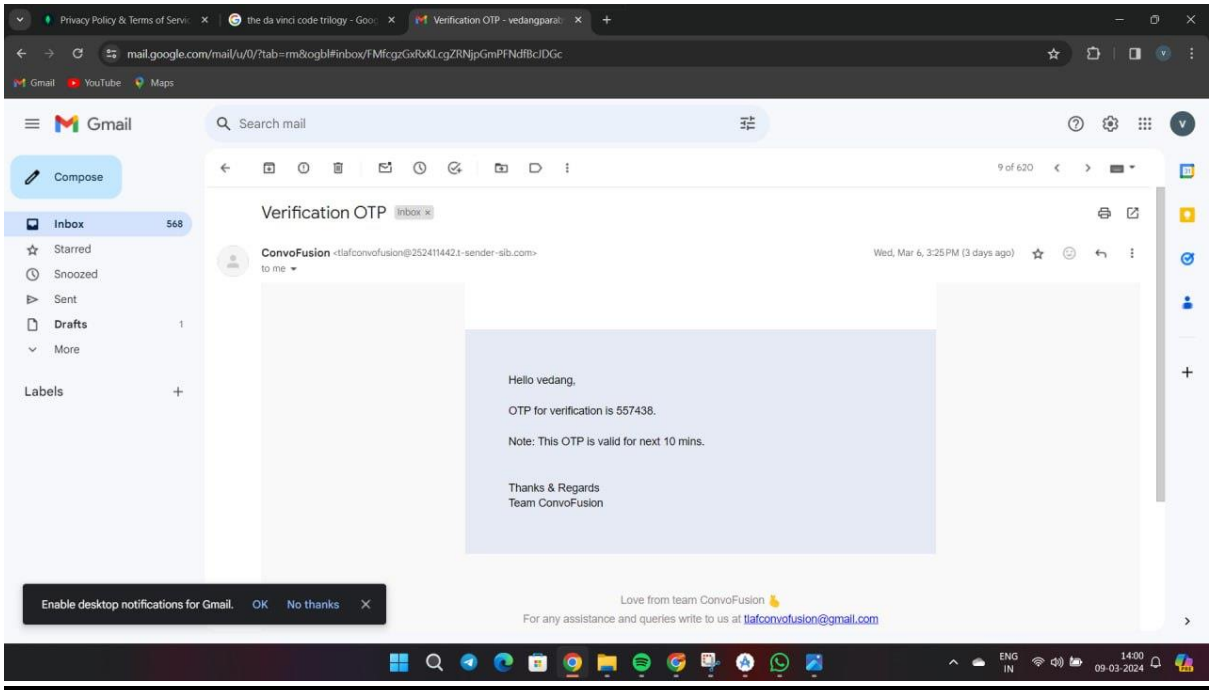


Fig 3

4.3 Log-in

In the Figure below, we can see the log-in page where after signing up user can log in with a valid username and Password

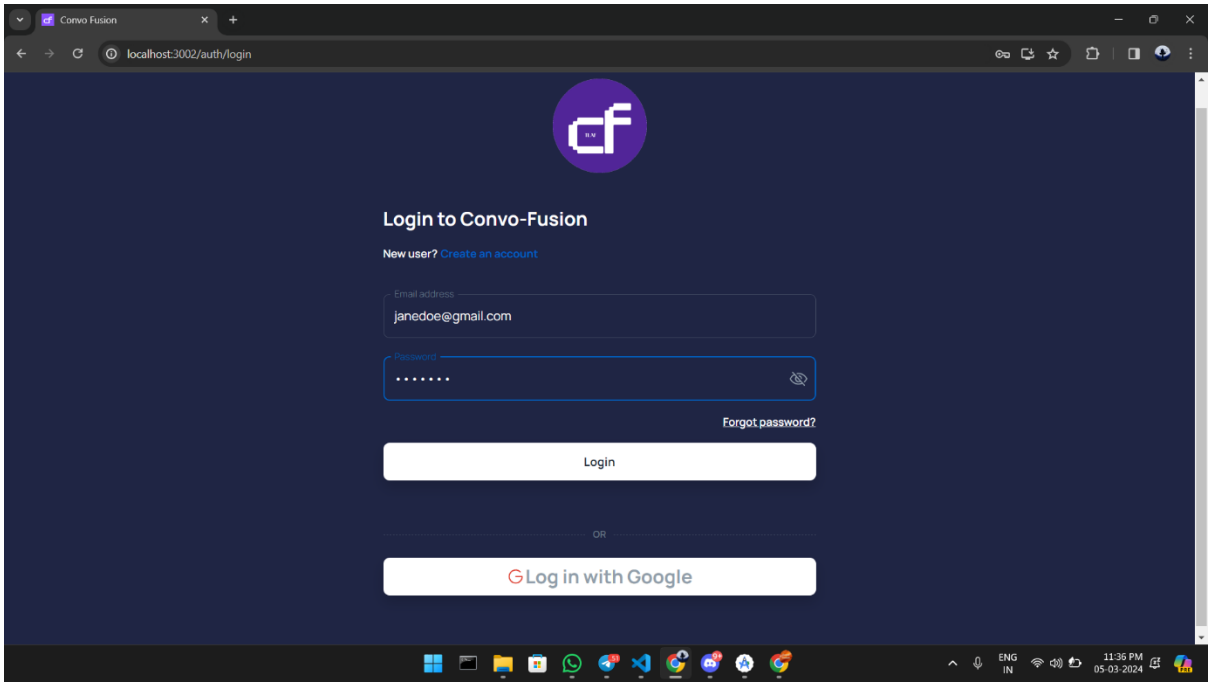


Fig 4

4.4 Forgot Password

In the Figure below, we can see the Forgot Password feature where the user can change his password if he/she forgets their password.

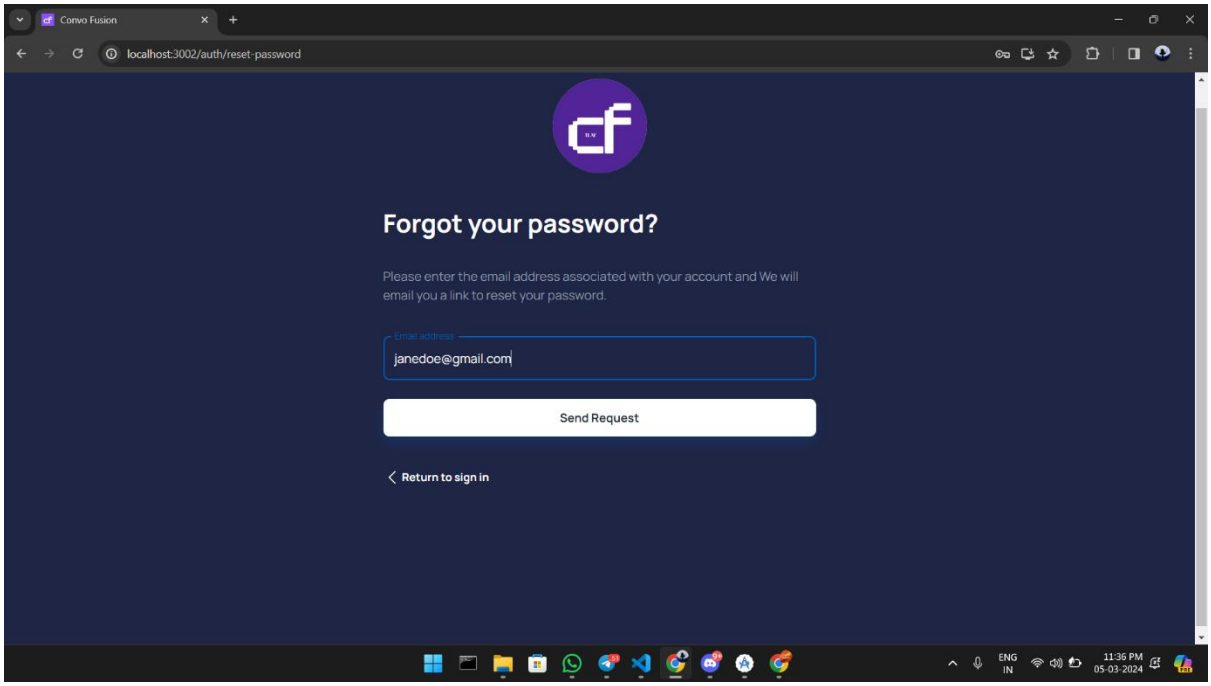


Fig 5

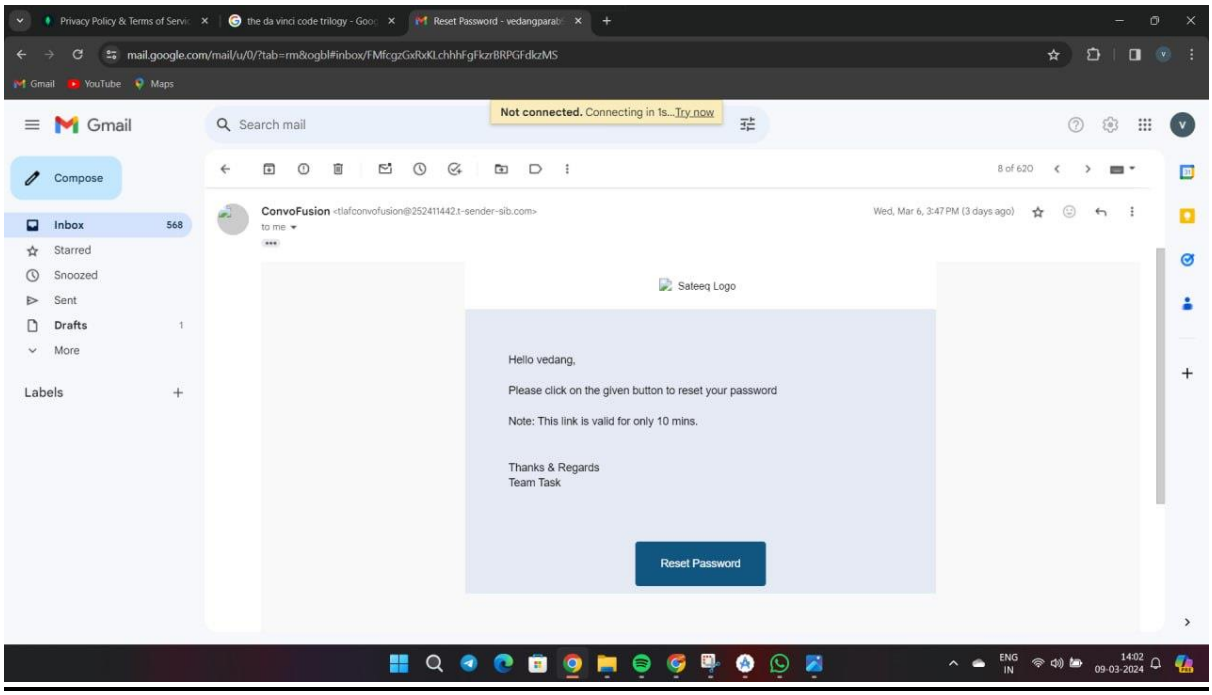


Fig 6

4.5 Home Page

In the Figure below, we can see the Home Page. It appears when the user signs in using the proper credentials

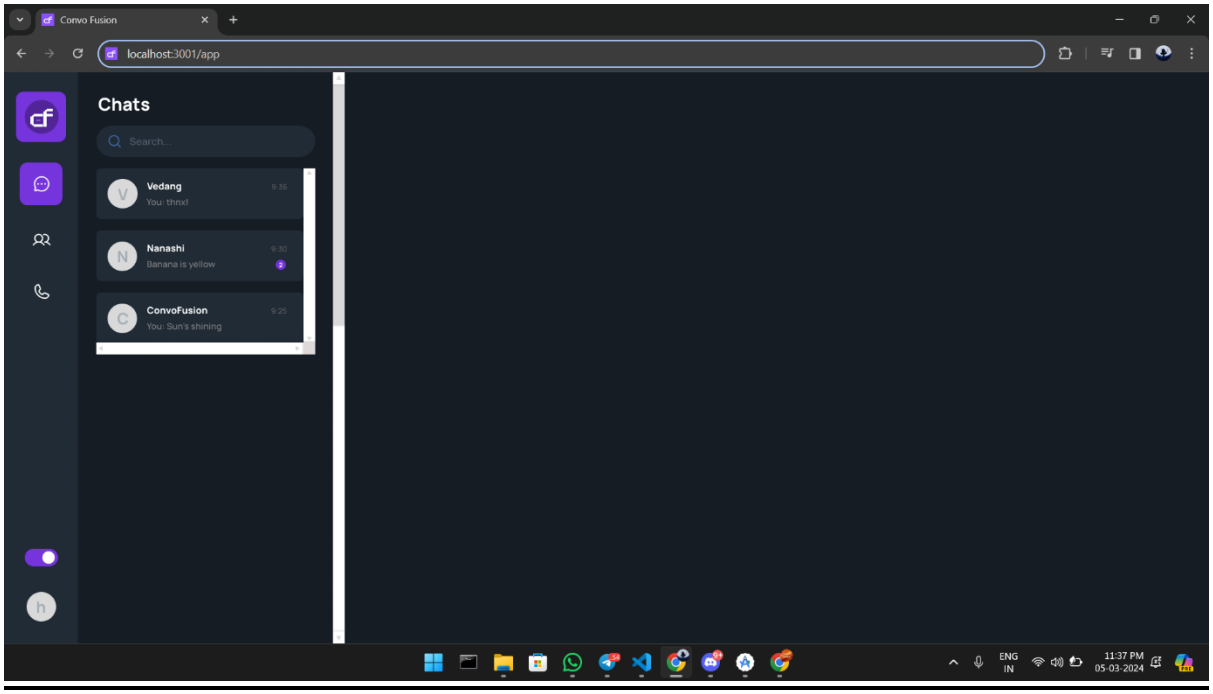


Fig 7

4.6 One-to-one chat space

In the Figure below, we can see the one-on-one chat page where the user is chatting with another user

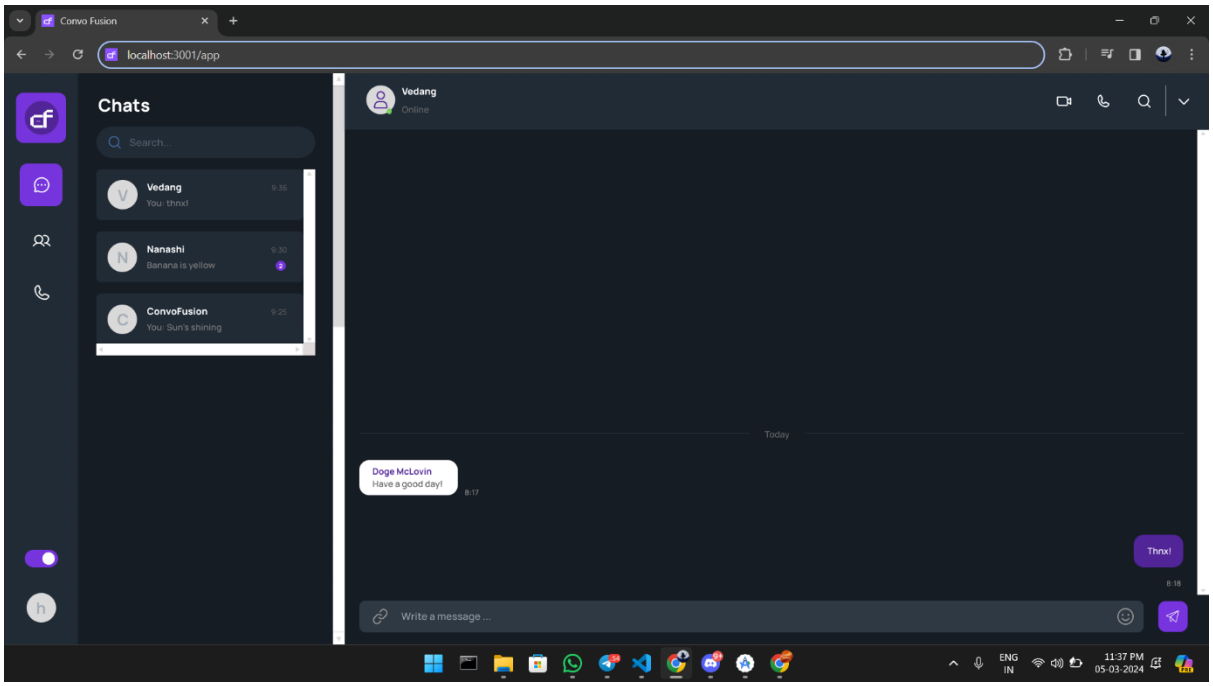


Fig 8

4.7 One-to-one voice call space

In the Figure below, we can see the one-on-one voice call interface where the user is talking to other users.

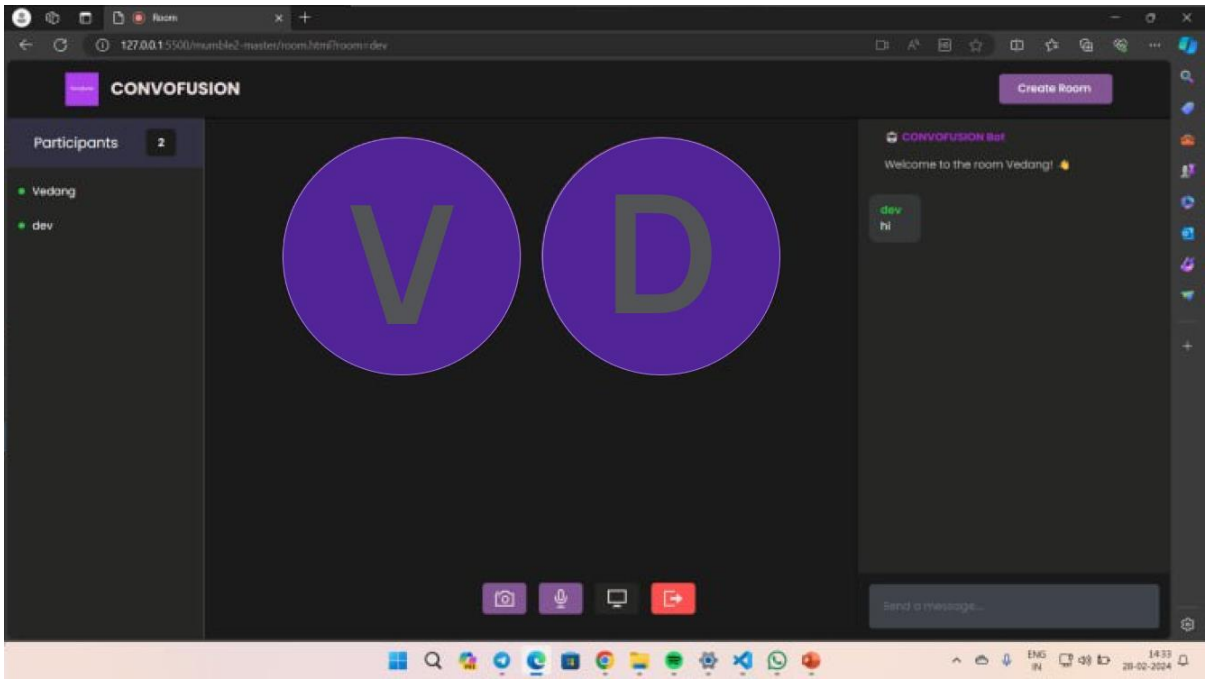


Fig 9

4.8 One-to-one video call space

In the Figure below, we can see the one-on-one video call interface where the user is communicating with the other users using a video call

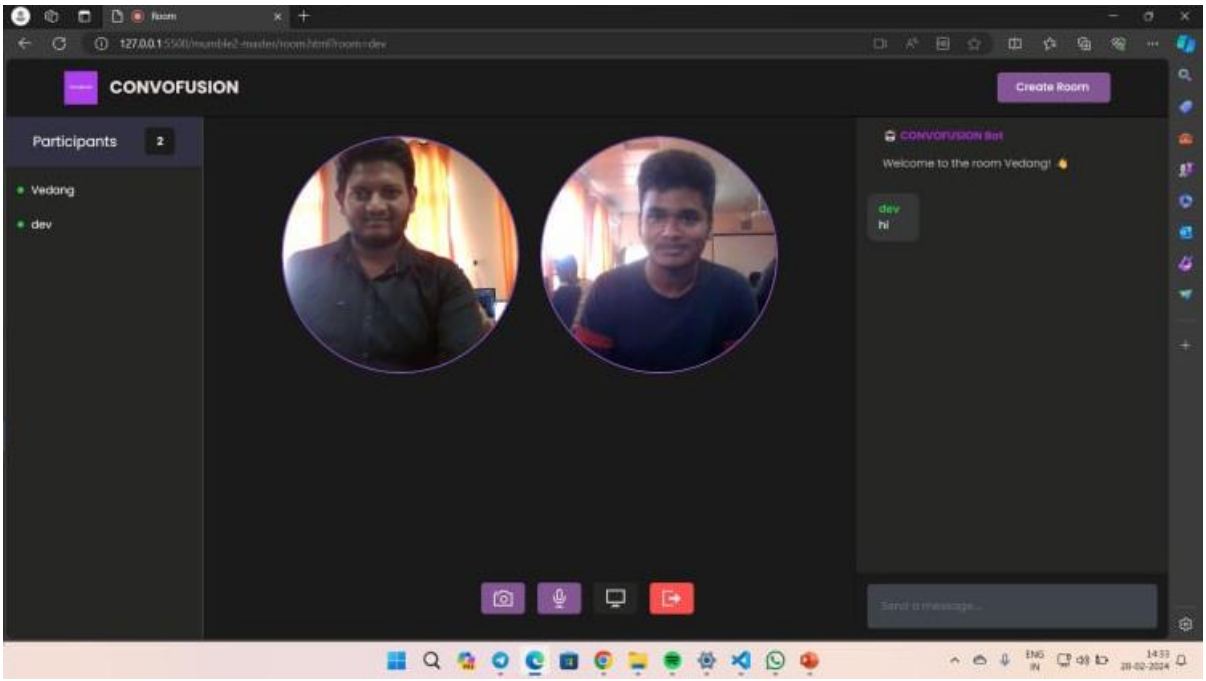


Fig 10

4.9 Call log space

In the Figure below, we can see a call log space where the user can see the history of calls

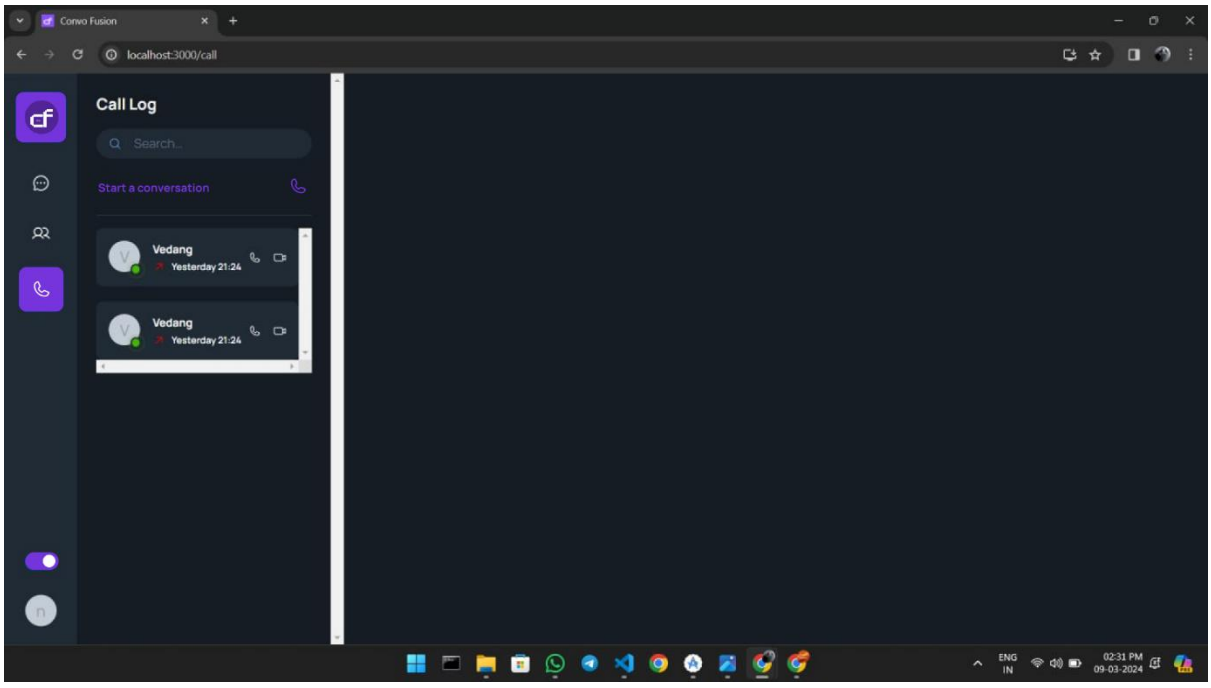


Fig 11

4.10 Group chat space

In the Figure below, we can see the group chat page user chatting with a group

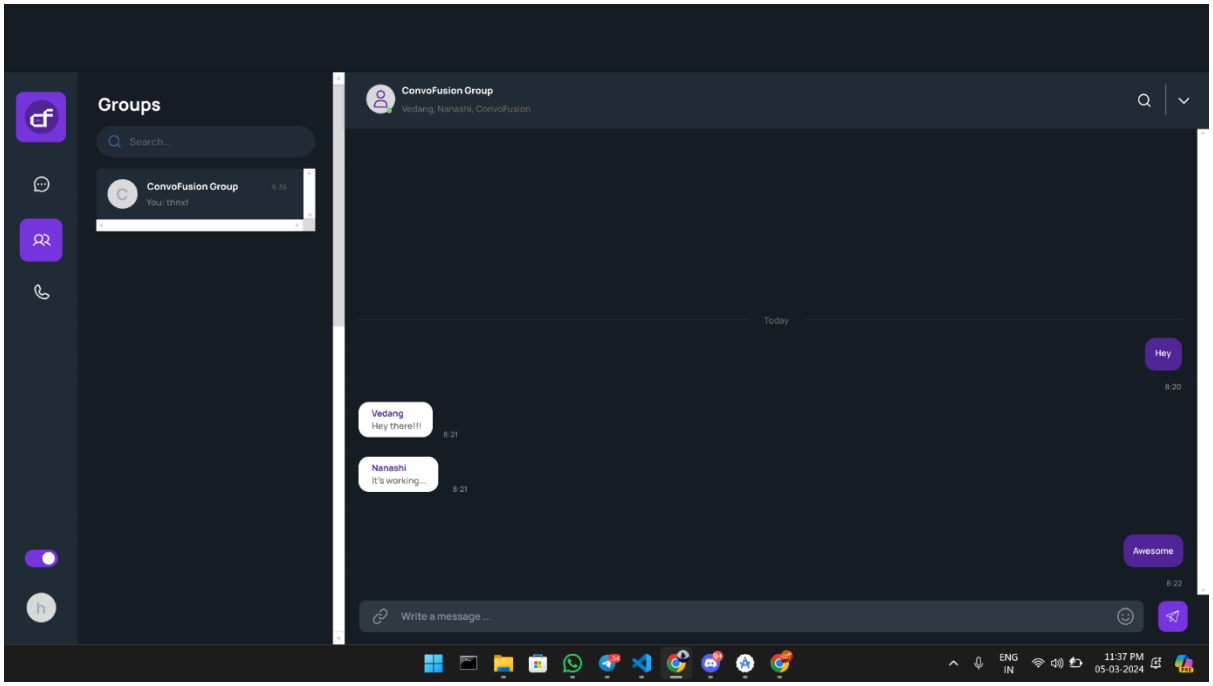


Fig 12

4.11 Group video call Interface

In the Figure, we can see the group call interface where users are using the group video call function to interact with each other

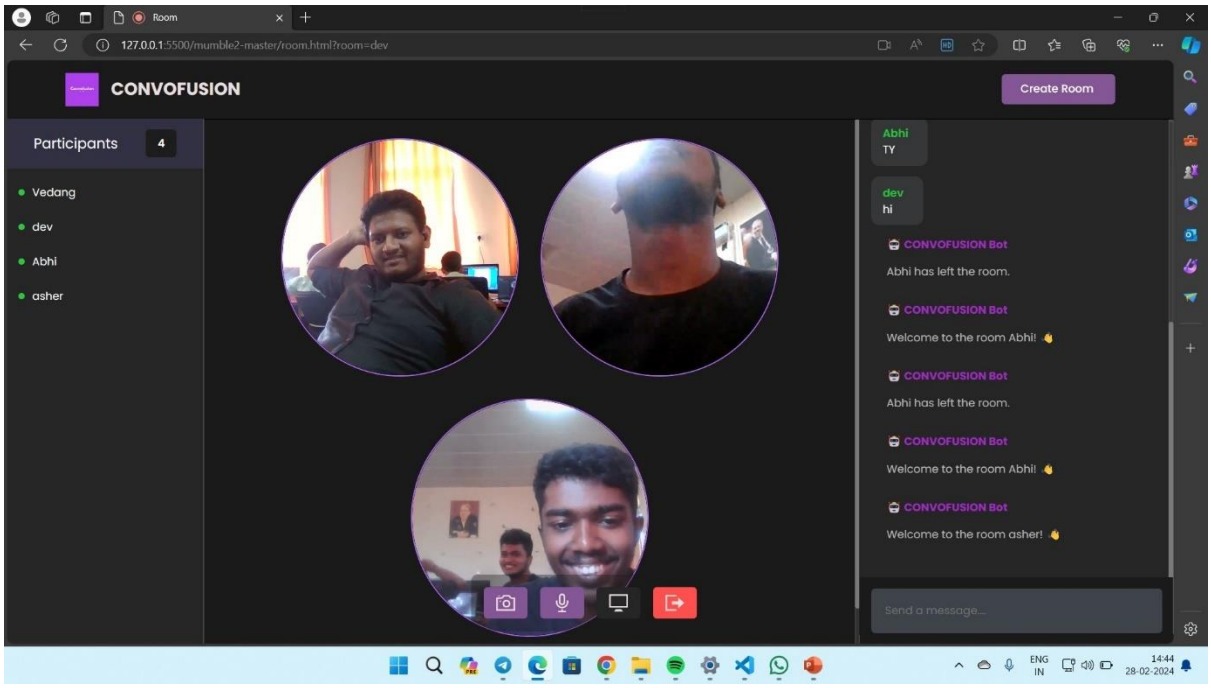


Fig 13

4.12 Screen Sharing Interface

In the Figure, we can see the screen screen-sharing interface where the user able to see the other user's screen

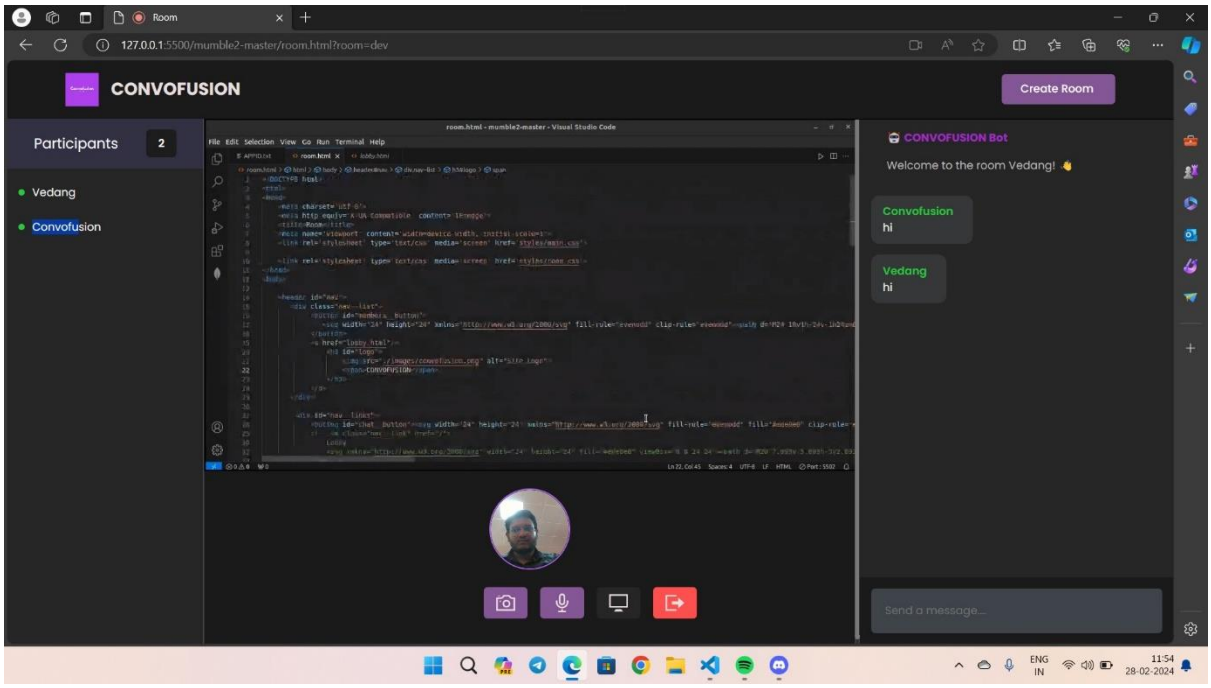
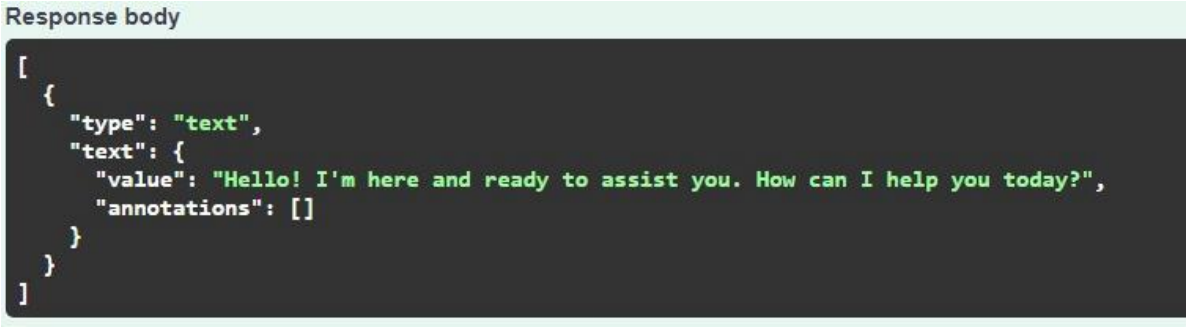


Fig 14

4.13 Chat bot

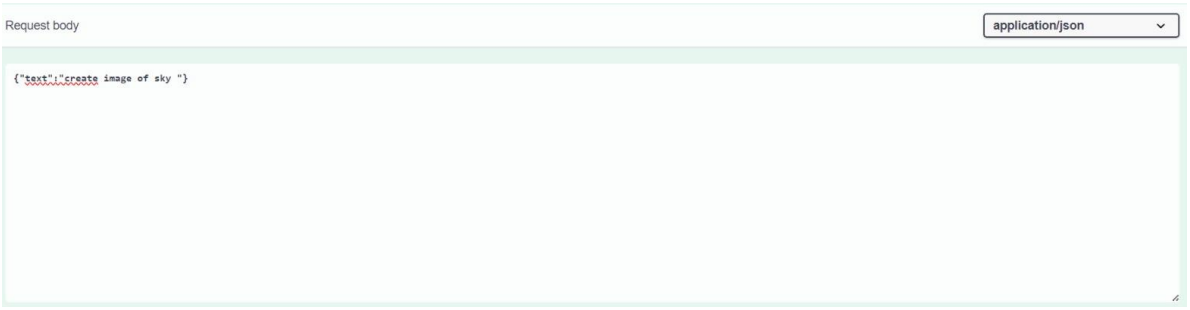
Prompt:

Output:



4.14 Image creation

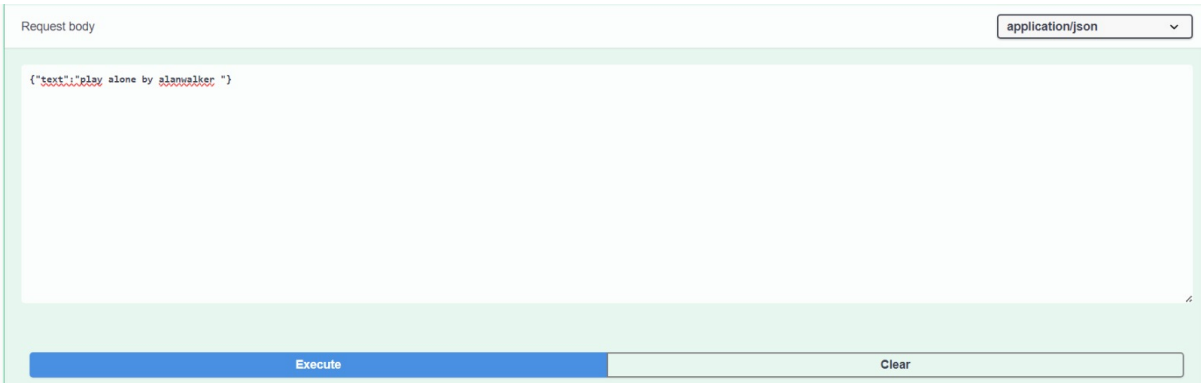
Prompt:



Output:

4.15 Spotify connection

Prompt:



Output:

CONCLUSION

The proposed project is to develop a chatting app that seamlessly integrates music streaming, AI chat capabilities, and image-generating AI services. Key features include user authentication, individual and group chat functionality, Spotify integration for music streaming, ChatGPT API for AI chat, Dall-e API for image generation, search functionality, customization options, and a notification system.

Existing systems like WhatsApp, WeChat, and Snapchat are popular competitors in the messaging app market. WhatsApp offers unlimited calling and group texting but raises security concerns due to potential integration with Facebook. WeChat provides quick registration and high-quality voice/video calling but has a less intuitive interface. Snapchat excels in capturing real-time moments with its visual approach but faces criticism for a confusing interface.

In summary, the proposed app aims to combine the strengths of existing messaging apps with additional features like music streaming and AI capabilities to provide users with a unique and engaging chatting experience.

BIBLIOGRAPHY

YouTube:

https://youtu.be/SqcY0GlETPk?si=Wi1M_fkgTzrELwih

https://youtu.be/k4mjF4sPITE?si=loV6EIUy6hDSgMwa

https://youtube.com/playlist?list=PLu0W_9lII9agiCUZYRsvtGTXdxkzPyItg&si=wd59AwmQ8je39OvY

GitHub:

https://github.com/safak/youtube2022/tree/react-chat

Others:

https://www.w3schools.com

https://stackoverflow.com

https://javapoint.com

https://chat.openai.com