Team Chat App : Deliverable 5 – Project Phase 3

Requirements

Phase 3 Implementation Requirements:
1.Video channels: Implemented creating of video channels which can created by admin in his own server, where all the users of the server can join and communicate, and the users can screen share use microphone to speak by giving it access, and by default the video will be on, where he can turn it off if the user needs to or can even leave the call by clicking on the leave button.. Leverage WebRTC for implementing direct peer-to-peer communication channels, offering users enhanced interaction options beyond text.
2.Audio channels: Implemented the creating of audio channels this can be done by admin of the server, where by default he video will be off, but this can be turned on by giving access to the camera, and users can also speak by giving access to microphone, or can even leave by clicking the leave button. The user can even share their screens in order to present their screens.
3. Text channels: Implemented the creation of text channels so that the users can form groups and chat which be related to specific task or topic. This exactly behaves like general channel where users can only communicate by text messages and do file sharing and can even use emojis.

Phase 3 development deliverable plan has been changed a bit :
Completed
1. Comprehensive Testing : Reason : As part of our dedication to producing software that meets the highest standards, my colleagues and I have put in place a strict testing procedure that includes unit and integration testing for every part of our program. The goal of the unit testing phase was to make sure that every module functions as intended on its own by independently verifying the dependability and functionality of each component under varied circumstances. We performed integration testing after unit testing to assess how the application's components interacted with one another. Finding any inconsistencies, dependencies, or problems that arise only when various system components interact was made possible thanks to this. These tests have been crucial in verifying that our program works flawlessly both when integrated as a whole and when used independently.

Added
2. Text/Audio/Video channel creating : Reason : We've included text, audio, and video channels in our team chat software to meet a variety of communication needs and improve usability and user engagement. All users are able to join, interact visually, share screens, utilize microphones, control visibility, and turn off video using the admin-created video channels on their server. This functionality is essential for supporting rich, peer-to-peer WebRTC communication during dynamic engagements that depend on visual signals, like team meetings or presentations. Similar to video channels, audio channels are admin-created and by default turn off video for voice-focused communication. However, they can also share screens and videos, meeting varied user needs and comfort levels. Text channels facilitate more conventional digital communication methods, such as file sharing and text messaging; they are perfect for ongoing, talks focused on a particular subject and brief information sharing. Our program caters to a broad spectrum of communication scenarios and preferences thanks to its diverse channel strategy, which also improves productivity and teamwork.

Removed
3. Security Audits : Reason : We are depending on the strong security frameworks of Clerk, UploadThing, LiveKit, and Aiven.io for Prisma DB throughout this phase of the project. These companies are well-known for their dedication to security and compliance. These services employ a number of stringent safeguards, including advanced authentication processes, data encryption, and compliance with international standards such as GDPR, HIPAA, and SOC 2. We've deliberately decided to take use of their well-established security infrastructures because of their experience, which frees up our resources for core development. Although we have faith that these platforms uphold strict security protocols, we continue to be on the lookout for new security threats and are prepared to modify our approach as needed.
4.User Feedback Collection : Reason : Our technique to gathering user feedback has been updated in our current report as a result of our professor's extensive peer review procedure. In this step, other teams do comprehensive usability testing that covers every facet of user experience and interface assessment. This structured feedback system eliminates the need for further direct user interaction at this stage, enabling us to quickly improve our application based on focused feedback from experienced colleagues.

Modified set of requirements for other phases:

Phase 1 Updated Implemented Requirements (Done):
1. Real-time Messaging & File Sharing: Implemented a core messaging system supporting text and file messages across different chat settings for one on one and inside a general group setting, utilizing Socket.io for seamless, real-time communication.
2. User Management with user signup, signout, delete and signin the user account via Clerk:
- Integrated Clerk Service API which gives the built in functionalities for all streamlined user authentication and authorization, enabling users to authenticate and register using their Google accounts, thus prioritizing security and user convenience.

3. UI Customization and Themes:
- Introduced customizable UI themes, specifically implementing dark, light and system default mode, enhancing user experience by allowing personalization according to user preferences using next js ui components.

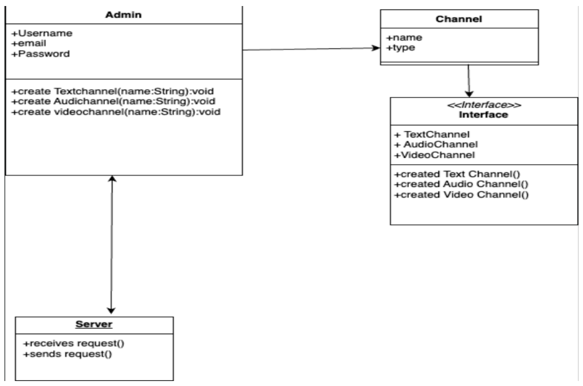
4. Invite Code Functionality for Group Channels for friends invite:
- Developed an invite code system allowing admin to invite his friends, focusing on easy expansion and collaboration within group settings where we can invite friends through the invite code system to the friend list channels.

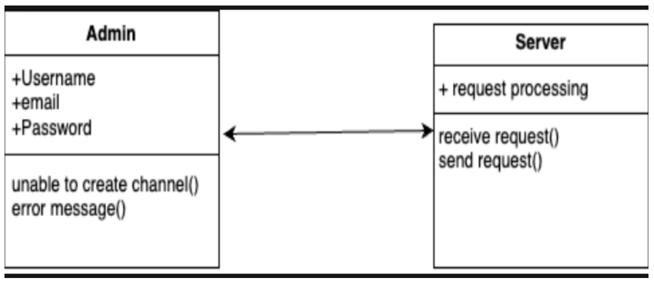
5. Create server:
- Implemented create server functionality for making the app functional with group channel, one on one communication, invite friends and file sharing. Where the admin has the capability to create his server and invite his friends to communicate.

# Phase 2 Updated: Enhancements and Integrations (Done)
1. Customize Server: Admins can change the server's name and image via server settings.
2. Delete Server: Admins can delete a server, which also removes all channels, members, and related data.
3. Leave Server: Guests and moderators can leave the server, losing access and having their messages removed.
4. Search Functionality: Users can quickly find channels and server members using a search bar.
5. Member Kicking: Admins can remove members who are unfit or misbehave, also removing all their associated chats.
6. Role Assigning: Admins can assign "Guest" or "Moderator" roles, with each role having specific permissions and responsibilities.
7. Chat and File Deletion: Admins and moderators can delete messages and files in group channels, but cannot access private direct messages.

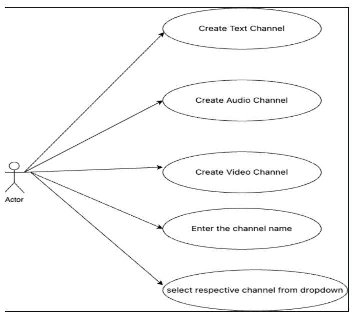
UML design

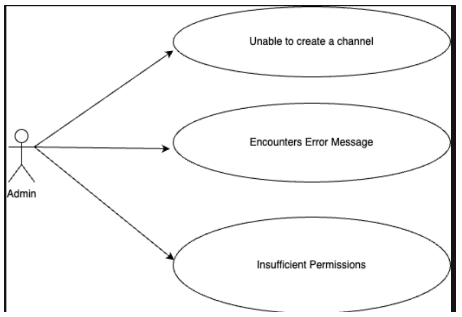
The following are the various UML diagrams for phase 1: **Class diagram for "Channel creation by admin" feature a) working case b)Error case**



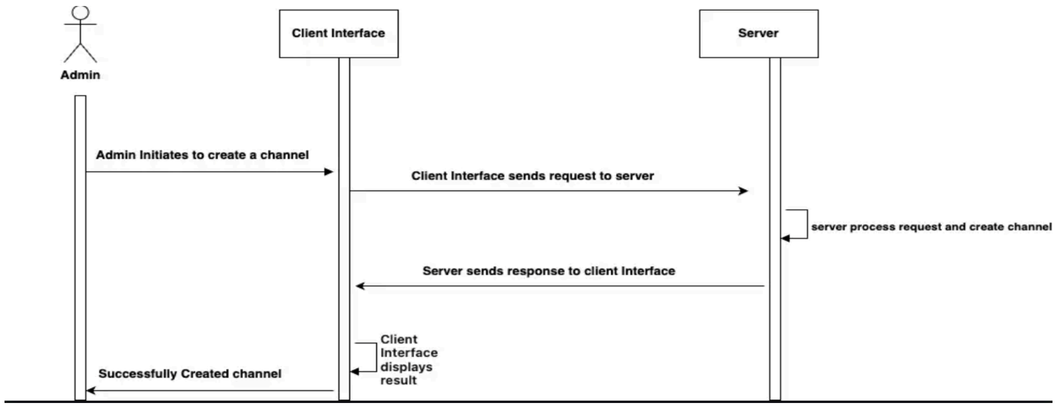


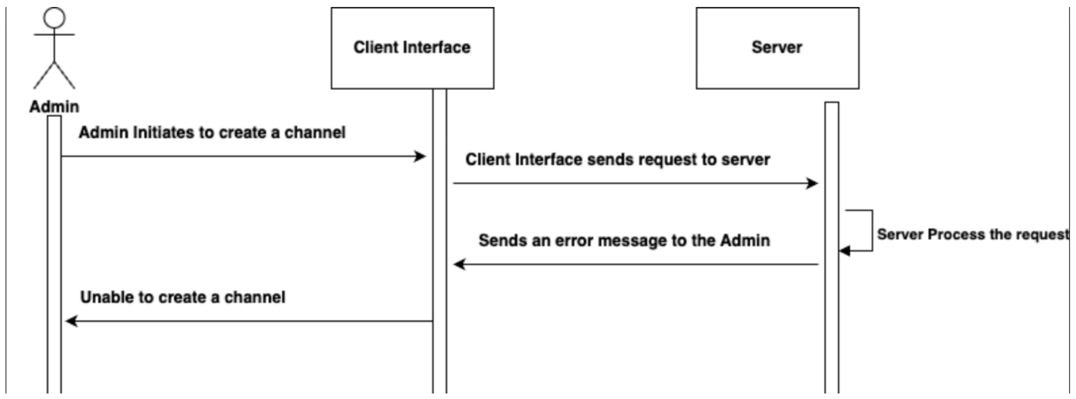
Use case diagram for Channel creation by admin feature a) Working case b) Error case





Sequence diagram for "Channel creation by admin" feature a) Working case b) Error case


Admin
Client Interface
Server
Admin Initiates to create a channel
Client Interface sends request to server
server process request and create channel
Server sends response to client Interface
Client Interface displays result
Successfully Created channel


Admin
Client Interface
Server
Admin Initiates to create a channel
Client Interface sends request to server
Server Process the request
Sends an error message to the Admin
Unable to create a channel

Test Cases

User Registration, authentication, signout and deletion via Clerk:

1. Test User signup through clerk authentication service:

Description: This test verifies if a user can successfully sign up with his google account.

Input: User directly sign up with his existing google account.

Expected Output: User is sign up successfully using clerk authentication service.

2.Test User Authentication sign in and sign out:
Description: This test verifies if a registered user can sign in securely and sign out after signing in to the app.
Input: User signup using google account and signs out using signout option.
Expected Output: User is authenticated successfully and granted access to the app if he signs in and sign out when he clicks on sign out after signing in.

3. Test Invalid User Authentication:
Description: This test checks the system's response to invalid login credentials.
Input: User logs in with unregistered google account.
Expected Output: User receives an authentication error message and is not granted access.

4. Test user Delete action his profile via clerk services:
Description: This test checks the app response to delete user from the app.
Input: User clicks on delete account option provided by clerk.
Expected Output: User account gets deleted.

Create Server:
5. Test if the create server is working fine:
Description: This test checks if a server can be created within which channels can be created in the future for communication and one on one communication.
Input: User clicks on create server options providing server name and image.
Expected Output: New server that is specific to the user gets created where the created user is now the admin, and he can invite his friends for one on one communication with general channel being common.

Invite Code functionality for friends:
6. Test Invite friends code url generation:
Description: This test checks if the admin of the server can generate invite code to invite his friends to his server.
Input: Clicks on the invite people button where the invite url will be generated or we can even generate new code by clicking generate a new link button in invite modal.
Expected Output: Generates invite friends url.

7. Test if the friends can join the server using the invite code:
Description: This test checks if the friend is able to join the server using the generated invite code by using it in the browser.
Input: Friend pastes the invite code url in his browser and registers his account using clerk.
Expected Output:user should be able to join the server of this friend through invite code and see his name in the left column below the general channel.

Real time messaging:
8. Test communication in general channel and in one on on one communication where general is set as default for a server:
Description: This test checks if the messages can be sent in the general channel and direct communication through socket.
Input: Type a message in the chat input of the general channel or select a user in the user column and enter a chat message from chat input.
Expected Output: Receives successful message from both general channels and on direct message in one on one communication.

9. Chat with emojis:
Description: This test checks if we can send emoji as messages.
Input: User clicks on the emoji option on the chat input and sends a message.
Expected Output:Successful can communicate with emojis.

10. Edit and delete messages:
Description: This test checks we can edit and delete the sent messages.
Input: User clicks on edit and delete option for the messages he sent.
Expected Output:Successful can edit and delete his messages.

Real time file sharing:
11. Test functionality of file sharing:
Description: This test checks if the images and pdf are sent..
Input:Upload image and pdf from the chat.
Expected Output: Receives successful message from both general channels and on direct one on one communication where upon clicking them will opens up in the new tab in the browser.

UI customization and themes :
12. Test functionality of UI themes:
Description: This test checks if the user can switch between dark, light and system default modes.
Input: Click on the button to toggle between dark,light and system default modes.
Expected Output: The UI toggles between dark,light and system default modes.

Server Management :
13. Customize your server:
Description: This test checks if the admin can customize the server by changing the server by its name and by its image
Input: Clicks on the Server Settings button from drop down menu of the server, where a modal appears to edit and save server name and image, where after changing them the admin can click on save button on that modal to save his changes.
Expected Output: The name of the server and its image needs to change accordingly with the changes made by admin.

14. Delete server
Description: This test checks if the admin can delete the server which he created.
Input: Click on the delete server drop down option from the drop down menu of the server

Expected Output: The server which is previously created by the admin should be deleted.

15. Leave server
Description: This test checks if the guest or the moderator can leave the server
Input: For this user needs to be either guest or moderator of the server, and note the admin can only delete his server and not leave.
Expected Output: The user will leave the server and he will no longer be accessible inside the server in channels / direct messages.

Search functionality :
16. Search functionality
Description: This test checks if the user can search and navigate the channels and members and who are present in the server.
Input: Click on the search bar and type the channel/member name who the user likes to search for. And navigate to a specific channel/member when the user clicks on the search result.
Expected Output: Displays all the channels/members present in the server initially, and need to show correct search results with respect to the entered text, after entering name, and navigate to specific channel/member after clicking on the channel/member.

Manage member :
17. Admin Member kicking
Description: This test checks if the admin can kick (remove) any member from the server.
Input: Click on manage member button from server drop down menu, and after the display of all the members in the channel, click on more options button for the specific member you want to kick, and then select kick option to kick them out of the server.
Expected Output: The user needs to get kicked out, and all the members of the channel should not be seeing the kicked out member and even that person needs to be removed from the channels.

18. Admin Role assigning
Description: This test checks if the admin can change the roles of the users in his server to either “Guest” or “Moderator”
Input: Click on manage member button from server drop down menu, and after the display of all the members in the channel, click on more options button for the specific member you want to change the role to “Guest” or “Moderator”
Expected Output: The roles and their specific permission of the users needs to be updated when changed to :
“Guest” : He is allowed to message or file share in channels/direct message, and he can delete or edit his own messages, and he gets option to leave the server after click on his server drop down menu. And he has access to his manage his account by deleting, editing his account and he has even access to change the UI Theme.
“Moderator” : He has the additional functionality to delete any message in group channels when compared to “Guest”.

Chat message/ File Sharing deleting :
19. Admin & Moderator chat message/file delete functionality
Description: This test checks if moderator or admin can delete the shared messages or file in group channels or in direct messages.
Input: Click on the delete button after the end of message or shared file like image or pdf field box, by any user.
Expected Output: The message or the file sharing will be deleted when clicked on delete button.

Create text/audio/video channel :
20. Admin create text/audio/video channel in this server
Description: Description: The admin can create text, audio, and video channels. In audio and video channels, users can toggle their audio/video settings, with video off by default in audio channels. Text channels support messaging with text and emojis.
Input: Admin click on create channel from his server dropdown, enter text/audio/video channel name and can select respective channel type in dropdown.
Expected Output: The respective channel which admin tried to create should be created.

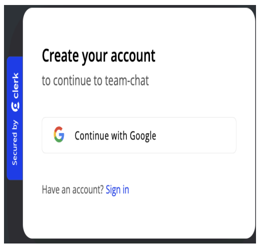
User Manual
Before you can use the application, please make sure your system meets the following requirements:
Operating System: Windows, macOS, or Linux
Web Browser: Chrome, Firefox, Safari, or Edge

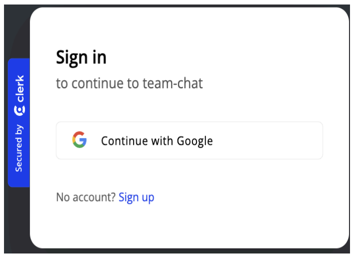
Installation
Downloading the Application
1. Request the latest version of the application from our team member or repository or TA.
2.Extract the downloaded archive to your preferred location.

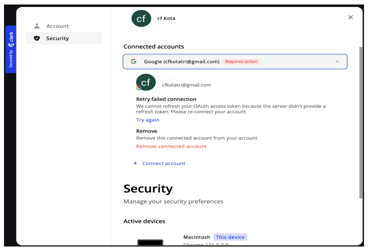
Installing required libraries and Running the application
1.Navigate to the root directory of the app.
2.Install all the packages listed on package.json using npm: npm i
3.Build the next js app: npm run build
4.Start the app locally: npm run dev
5.Visit port http://localhost:3000/ on successfully building and running the application.

Using the Application

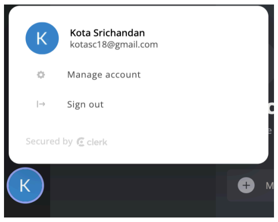
User Registration, sign in
1.Create your account using Clerk by sign up and sign in using your google account.
2.Sign In using your registered google account.
3.Try sign up and sign in again if you did not do the sign in or registration properly, the app indicates “requires action” on your account if you haven’t set up your account properly.

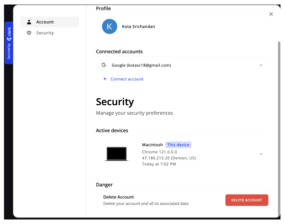


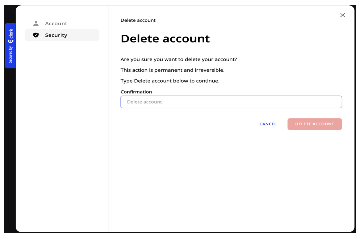




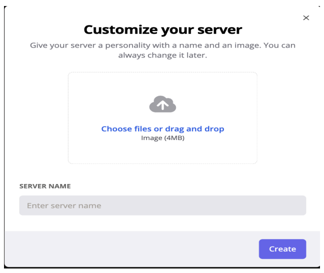
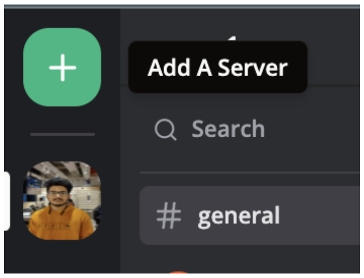
User Sign out and Delete account:
1.Click on the sign out button after signing to the app.
2.Click on delete account upon clicking on Manage Account to delete your account and confirm your delete account after typing “Delete account” in confirm delete screen



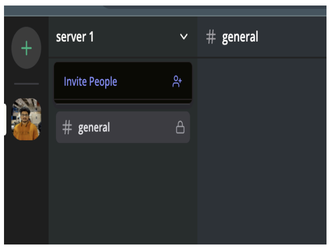


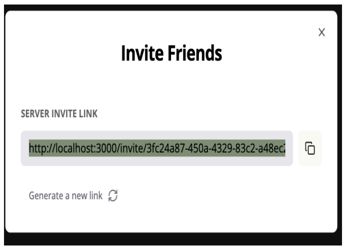

Account
Security
Delete account
Are you sure you want to delete your account?
This action is permanent and irreversible.
Type Delete account below to continue.
Confirmation
CANCEL
DELETE ACCOUNT

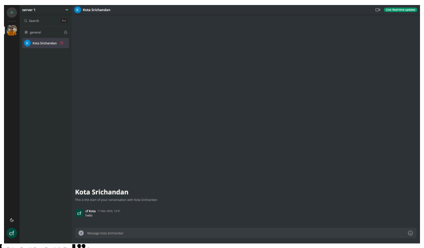
Create server:
1.Create your own server by providing the name and image of your server.
2.And create any additional servers by clicking on the add server button (+) with your custom name and image.


Customize your server
Give your server a personality with a name and an image. You can always change it later.
Choose files or drag and drop
Image (4MB)
SERVER NAME
Enter server name
Create


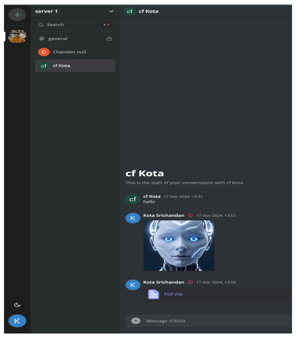
Invite friends:
1.Generate your invite code url to invite your friends to your server, by clicking on the invite people button on your server, this generates a new invite code link, if it expires try to generate a new one using the “generate a link” button.
2.After successful invite code link generation, invite your friends by sharing your invite link, and after your friend pastes this link in his browser he will be added to your server after he does the registration and signing process, and he will be shown with his name as mentioned in his google account name.

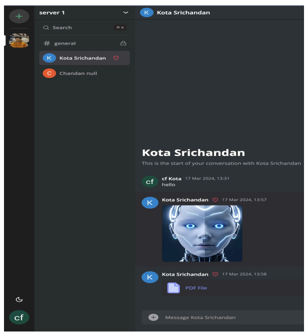


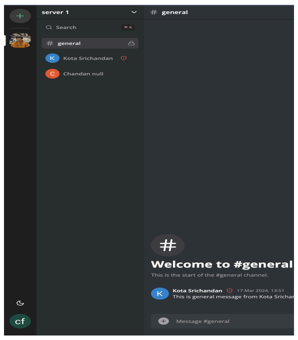
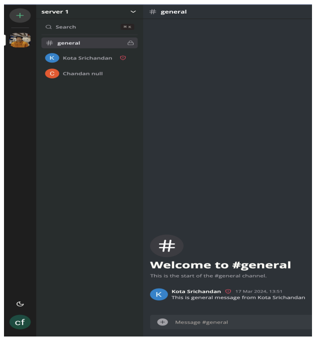




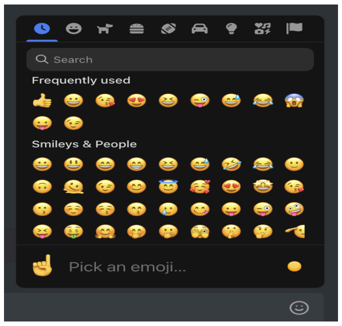
Real time messaging in direct messages and common channels like “general”:
1.Try sending out messages in the “general” channel which is a default channel provided for every server using the chat input box.
2.Try sending out messages by manually selecting the users you want to have direct one on one communication which will be below the “general” channel and use the chat input field to type your messages and press enter to send messages.

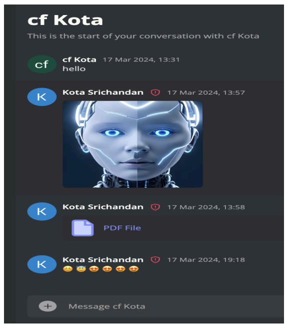





server 1
general
Kota Srichandan
Chandan null
Welcome to #general
This is the start of the #general channel.
This is general message from Kota Srichandan
Message #general

server 1
general
Kota Srichandan
Chandan null
Welcome to #general
This is the start of the #general channel.
This is general message from Kota Srichandan
Message #general

Emoji during communication: Use emoji during communication by clicking on the emoji button the chat input field, and use your desired emoji for communication.




cf Kota

This is the start of your conversation with cf Kota

cf Kota 17 Mar 2024, 13:31
hello

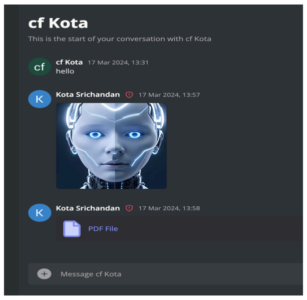
Kota Srichandan 17 Mar 2024, 13:57

Kota Srichandan 17 Mar 2024, 13:58
PDF File

Kota Srichandan 17 Mar 2024, 19:18
😀😀😀😀😀😀

Message cf Kota

Real time pdf and images sharing:
1. Share your pdf and images during communication on both one on one communication and channels like "general" using the upload button in the chat input field (+).
2.Then attach your attachments in the attach modal and click on send to send them.


cf Kota

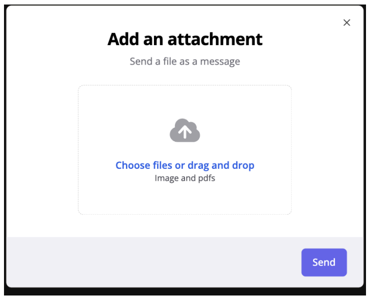
This is the start of your conversation with cf Kota

cf Kota 17 Mar 2024, 13:31
hello

Kota Srichandan 17 Mar 2024, 13:57

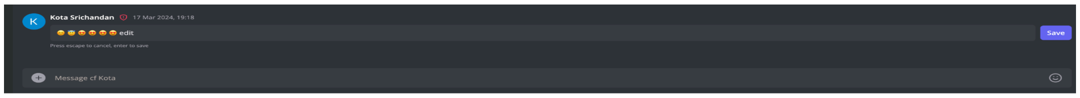
Kota Srichandan 17 Mar 2024, 13:58
PDF File

Message cf Kota


Add an attachment

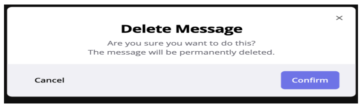
Send a file as a message

Edit and delete messages during communication:
1**.** Use edit and delete features for your messages during communication.
2. click on the edit and delete button on messages after sending them to edit or delete the message. For editing upon editing the message click on the save to save the edited message. For delete confirm if you really want to delete the message in delete modal.
3. After deleting the message "This message has been deleted" text will be shown in place of the deleted message.

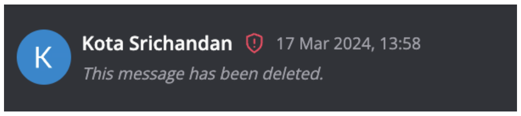

Kota Srichandan 17 Mar 2024, 19:18

😀😀😀😀😀😀 edit

Press escape to cancel, enter to save

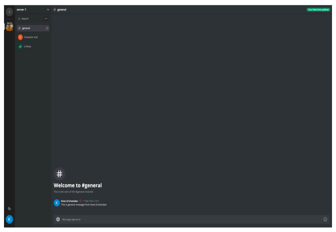

Delete Message

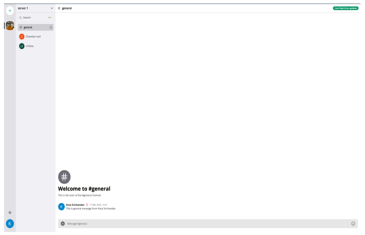
Are you sure you want to do this?
The message will be permanently deleted.

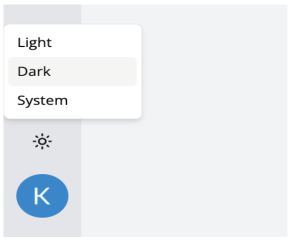

Kota Srichandan 17 Mar 2024, 13:58

This message has been deleted.

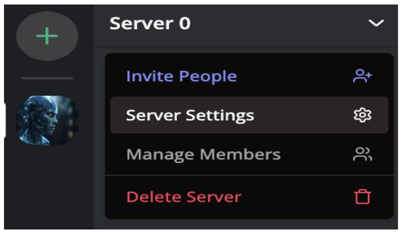
UI theme customization:
1.Click on the theme customization button to customize your theme for dark, light or keep it system default for UI customization.

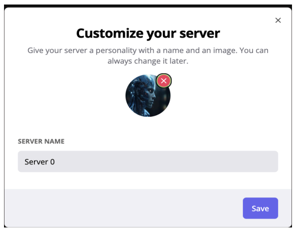






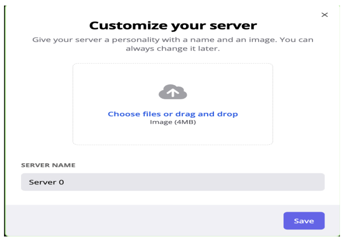
Server Management :
Customize your server:
1. Click on the server settings option from the server drop down menu, after which customize your server modal would be displayed under which you will have the option to change the server image by clicking on the red cross button to remove the existing server image and then choose files to upload your required server image then click on "Save" button to save your new changes.
2. In the server name field you can edit the existing server name and click then click on "Save" button to save your new changes.




Customize your server

Give your server a personality with a name and an image. You can always change it later.

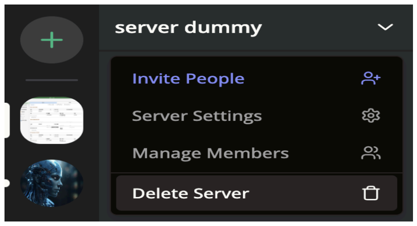
SERVER NAME: Server 0

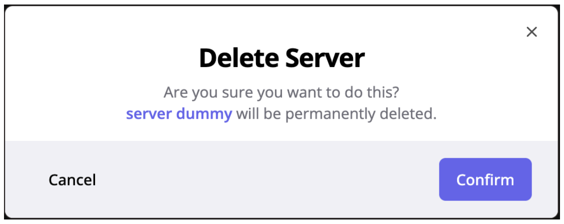

Customize your server

Give your server a personality with a name and an image. You can always change it later.

SERVER NAME: Server 0

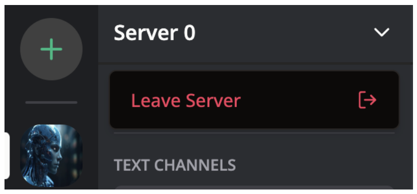
Delete server
You can delete your server by clicking on the delete server button from the server drop down menu, and after which the delete server modal will be popping up, in which you need to click on confirm button to delete your server.



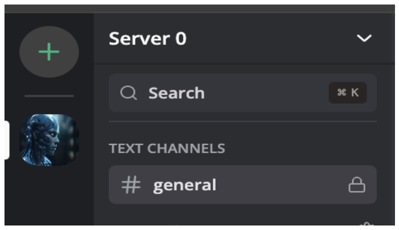


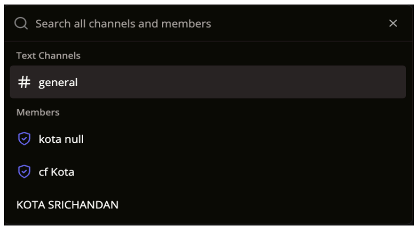
Are you sure you want to do this?
server dummy will be permanently deleted.

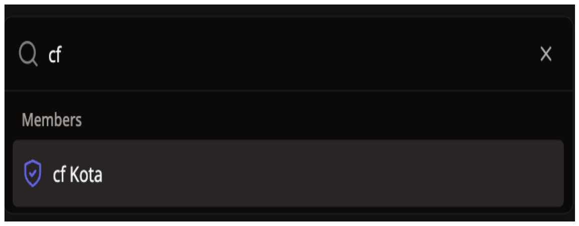
Leave Server
The moderators and guest can leave the server by clicking on the leave server button on their server drop down menu. Note : The admin who has created the server can only delete the server, he does not get the option to leave the server.



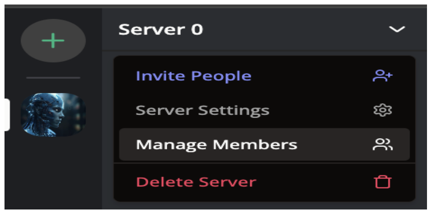
Search functionality :
Search functionality
1. You can search for channel names or members in the server by clicking on the search bar under server drop down menu.
2. After which the search modal will be displayed initially displaying all the existing channels and members in the server.
3. You can enter the name of the channel or the member in the server to get search results related to your entered text.

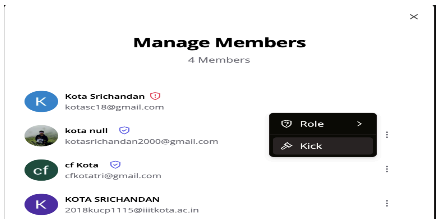






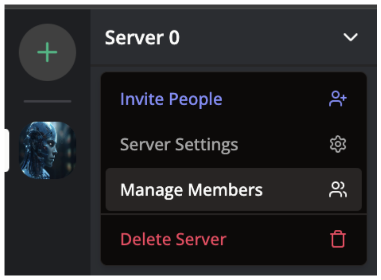
Manage member :
Admin Member kicking
1. Click on the manage members button from the server drop down menu, after wich Manage members modal will be displayed.
2. Click on the kick option by clicking on the specific user after clicking on more options button which is 3 vertical drop button

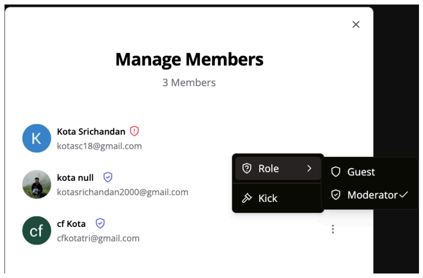




Kota Srichandan
kotasc18@gmail.com
kota null
kotasrichandan2000@gmail.com
cf Kota
cfkotatri@gmail.com
KOTA SRICHANDAN
2018kucp1115@iiitkota.ac.in

Admin Role assigning
1. Click on the manage members button from the server drop down menu, after wich Manage members modal will be displayed.
2. Hover on the Role button to display the existing roles to switch the current user to :
"Guest" : He is allowed to message or file share in channels/direct message, and he can delete or edit his own messages, and he gets option to leave the server after click on his server drop down menu. And he has access to his manage his account by deleting, editing his account and he has even access to change the UI Theme.
"Moderator" : He has the additional functionality to delete any message in group channels when compared to "Guest".

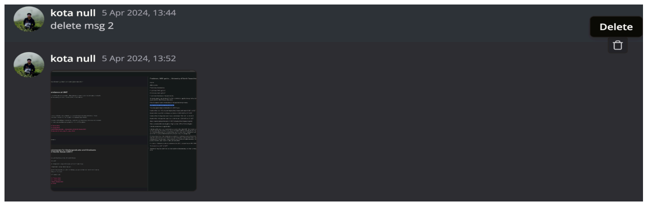




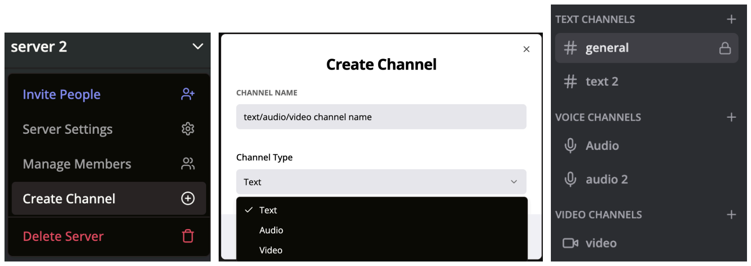
Kota Srichandan
kotasc18@gmail.com
kota null
kotasrichandan2000@gmail.com
cf Kota
cfkotatri@gmail.com

Chat message/ File Sharing deleting :

Admin & Moderator chat message/ file delete functionality
1. For this the user role needs to be either admin or moderator. He can delete any message. He can delete any message or file shred either in group channels or in direct communication by click on the delete button on the message box.


kota null 5 Apr 2024, 13:44
delete msg 2
kota null 5 Apr 2024, 13:52

Create text/audio/video channel :
Admin create text/audio/video channel in this server
For this Admin click on create channel from his server dropdown, enter text/audio/video channel name and can select respective channel type in dropdown. The respective channel which admin tried to create should be created.


Create Channel
CHANNEL NAME: text/audio/video channel name
Channel Type: Text

Compilation Instructions

Installing required libraries and Running the application
Prerequisites :
0. A computer with at least 8GB Ram.
1. Need to have node.js and npm run installed
2. A text editor like VS code
3. Git (Optional) for cloning the repo
4. Web Browser (Chrome or Safari)

After prerequisites :
1.Navigate to the root directory of the app (team-chat-app).
2.Install all the packages listed on package.json using npm: **npm install**
3. Make sure to set your create **env** file with "`NEXT_PUBLIC_CLERK_PUBLISHABLE_KEY`", "`CLERK_SECRET_KEY`" which are clerk authentication service api keys for user authentication, "`NEXT_PUBLIC_CLERK_SIGN_IN_URL`", "`NEXT_PUBLIC_CLERK_SIGN_UP_URL`" for handling both sign up and signin services, "`NEXT_PUBLIC_CLERK_AFTER_SIGN_IN_URL`", "`NEXT_PUBLIC_CLERK_AFTER_SIGN_UP_URL`" for handling after sign up and signin services. "`UPLOADTHING_SECRET`", "`UPLOADTHING_APP_ID`" for using UploadThing service which is for file sharing of images and pdfs, and "`DATABASE_URL`" for using the prisma DB for this go to aiven website to create the DB.

Note : .env file needs to be present in the root directory of the project

Check these website docs for additional queries :
Prisma DB : https://console.aiven.io/
Clerk : https://dashboard.clerk.com/
UploadThing : https://uploadthing.com/

As for now, i have updated all the **env** file parameters with the values that i have created, you can use them to run the app:
Use this snippet to add your .env file in the top directory inside "team-chat-app" folder as ".env" to run the app successfully.

```
NEXT_PUBLIC_CLERK_PUBLISHABLE_KEY=pk_test_cGxlYXNlZC1ibHVlZ2lsbC03Ny5jbGVyay5hY2NvdW50cy5kZXYk
CLERK_SECRET_KEY=sk_test_0M8YmK3QM3C1MNPRfdybv128XnWRkj7CfePe5MPCtE
NEXT_PUBLIC_CLERK_SIGN_IN_URL=/sign-in
NEXT_PUBLIC_CLERK_SIGN_UP_URL=/sign-up
NEXT_PUBLIC_CLERK_AFTER_SIGN_IN_URL=/
NEXT_PUBLIC_CLERK_AFTER_SIGN_UP_URL=/

UPLOADTHING_SECRET=sk_live_3d1e2ac22dd353d9d4107a36bece3751024386c780dd1e3f453c69a514c4f6a6
UPLOADTHING_APP_ID=5m3h3jfb5c

LIVEKIT_API_KEY=APIAJLtUCZ5ZWv8
LIVEKIT_API_SECRET=vtFzyPnYvp5p5RaCsxuKHzVqDcs2poS7Fkmo3c8bgcH
NEXT_PUBLIC_LIVEKIT_URL=wss://discord-wdgh6ygk.livekit.cloud
```

```
DATABASE_URL='mysql://avnadmin:AVNS_ZdB9l2GZjcKyCUfG71S@mysql-32092d6d-team-chat.a.aivencloud.com:21748/defaultdb?ssl-mode=REQUIRED'
```

4.Build the next js app: **npm run build**
5. Start the app locally: **npm run dev**
6.Visit port http://localhost:3000/ on successfully building and running the application.

Troubleshooting while compile and run (Typical Problems and Solutions) :
Problem: Problem with any of the dependency or next js app
Solution:Try to delete node_modules directory and install the dependencies again using “npm install” without changing anything in package.json.
Problem: error during any functionality breaking in the app.
Solution: Re-build the app using “npm run build” and run it again using “npm run dev”.

To learn more about Next.js, take a look at the following resources:
https://nextjs.org/docs - learn about Next.js features and API.

Testing the Application
The procedures below should be followed to compile and execute the test cases for the team chat web application:
1.Make sure the project is already setup by building the next js app.
2.Run the app using “npm run dev”
3.Visit 127.0.0.1:3000 to explore the websites and related features.
4.Make sure to use your own env file for testing your own app with your updated parameters..
5.Make sure to create your account and create a server to test various test cases.

Report ending feature summary

We have successfully implemented a wide range of features in our project, such as emoji support during communication, real-time messaging in direct messages and common channels, invite code generation for friend invitations, server creation and customization, role-based server management, search functionality for channels and members, and the ability to create text, audio, and video channels within servers. We have also successfully implemented user registration and authentication through Clerk. All of these features are functioning and have greatly improved our application's functionality and user experience.

Notwithstanding these achievements, there are still limitations, such as possible scalability problems at high user loads and unproven browser compatibility, which may have an impact on performance as the user base increases. We plan to implement a file management system for better organizational capabilities, mobile optimization to support users on different devices, performance optimization to guarantee the application can handle higher volumes of data and users effectively, and two-factor authentication for enhanced security in the upcoming development phase. These improvements aim to improve our application, fix any problems that may arise, and get ready for more developments.

In the next phaseWe intend to significantly improve our application in the next stage of our project by adding cutting-edge features like augmented reality (AR) tools for interactive communication, blockchain-based data security for unmatched privacy and dependability, and AI-driven sentiment analysis and automated moderation. In addition, we'll build a full analytics suite, connect it to IoT devices for in-the-moment interactions, and build virtual event spaces for online parties. The user experience will be further enhanced by capabilities like dynamic interface modification, voice-controlled navigation, and real-time language translation. These significant improvements aim to keep our app competitive and relevant in a quickly changing digital landscape by adding transformative value to it in addition to expanding its functionality.

Reflection

what has been accomplished : Our team has successfully added a number of features to our application over the course of this semester, greatly improving its functionality and user experience. Among the major achievements are the incorporation of Clerk for user identification, the construction of text, audio, and video channels, server management, real-time messaging, and file sharing capabilities. These functionalities were incorporated to provide smooth communication and administration within the program, meeting a variety of user requirements.

what went well : The main features, like real-time messaging, user registration, and server customization, were implemented with remarkable success. The fact that the program can do these fundamental yet important tasks without experiencing any serious problems or defects is evidence of the strong foundation we have established. Users have given modern technologies like Clerk for authentication and real-time communication positive praise since they are dependable and simple to use. These strategies have proven to be especially successful.

Areas for Improvement : Although the project was successful in many regards, there are a few areas that could have been improved:
Performance and Scalability: It's critical to concentrate on scaling the infrastructure as the application starts to draw more users in order to effectively manage growing traffic and data demands. Prioritizing performance optimization is necessary to ensure seamless operation under increased demands, especially for real-time functionality. **Browser Compatibility:** Extensive testing was not carried out on a variety of browsers. In the future, accessibility and user pleasure will be improved by guaranteeing compatibility with all major browsers. **Advanced Feature Integration:** Although the existing feature set offers a solid foundation, including additional cutting-edge technologies like artificial intelligence (AI) for real-time analytics and automatic content moderation could greatly expand the application's potential and increase user engagement.

Member Contribution Table

Here is the member contributions table for Phase 3, including details of each team member's contributions, overall contribution percentage, and any relevant notes:

Member Name	Contribution Description	Overall Contribution (%)	Note
Srichandan Kota	- Led system design and technological selection, focusing on microservices architecture and security planning and GitHub.\ - Contributed in server management	12.5%	Exhibited outstanding leadership and decision-making in system design, ensuring a robust and scalable foundation for the application and Github

Swapna Sonti	- Developed front-end components, particularly for UI customization themes and real-time messaging functionalities. - Worked on the JS part of the system to add more dynamic functionalities to the front-end	12.5%	Played a pivotal role in front-end development, enhancing the application's interactivity and visual appeal and done audio/video text channels creation.
Sandeep Chowdary Ari	- Engineered the backend for real-time communication and managed server and channel functionalities - contributed in developing search functionality	12.5%	Essential in establishing a reliable and efficient backend system, facilitating seamless communication across the application
Venkata Sai Shankar Koppula	- Directed UI/UX design efforts, focusing on usability, accessibility, and implementing dark and light modes	12.5%	Significantly contributed to the application's design, improving user experience and accessibility through thoughtful design choices.
Shivanandha Reddy Vasudevula	- Managed database architecture, ensuring data integrity and optimizing data management strategies	12.5%	His expertise in database architecture enhanced data handling efficiency, supporting the application's core functionalities.
Gana Deekshith	- Led the integration of secure authentication flows, detailing security protocols and user verification processes. - Contributed in member maangement	12.5%	Bolstered the application's security framework, ensuring user data protection and a trustworthy environment.
Kantumutchu Dinesh	- Spearheaded the file sharing implementation and contributed to the real-time messaging system using Socket.io. Enhanced data handling for efficient communication and file exchange within the application.	12.5%	His pivotal role in integrating file sharing and optimizing real-time communication channels significantly bolstered the application's functionality, ensuring seamless interactions and data exchange.
Bhanu Prasad Krishna Murthy	- Led documentation efforts and developed the application's security framework, focusing on authentication and authorization.	12.5%	His comprehensive documentation and focus on security have been vital in establishing a solid foundation for the project.