

Ripple Chat

Austin Sternberg, Noah Al-lahabi, Colton Staiduhar, Ian Rohrbacher, Paige Ogden

1. Description and Functionality of the Project

Ripple Chat is designed to provide users with public chat rooms where they can send both text messages and drawn images. It also allows users to join public voice calls and activate their cameras for video chat. The platform draws inspiration from older, now-defunct services like AOL Instant Messenger, Omegle, and PictoChat, aiming to revive the social and casual feel of early internet communication tools.

Technical Overview:

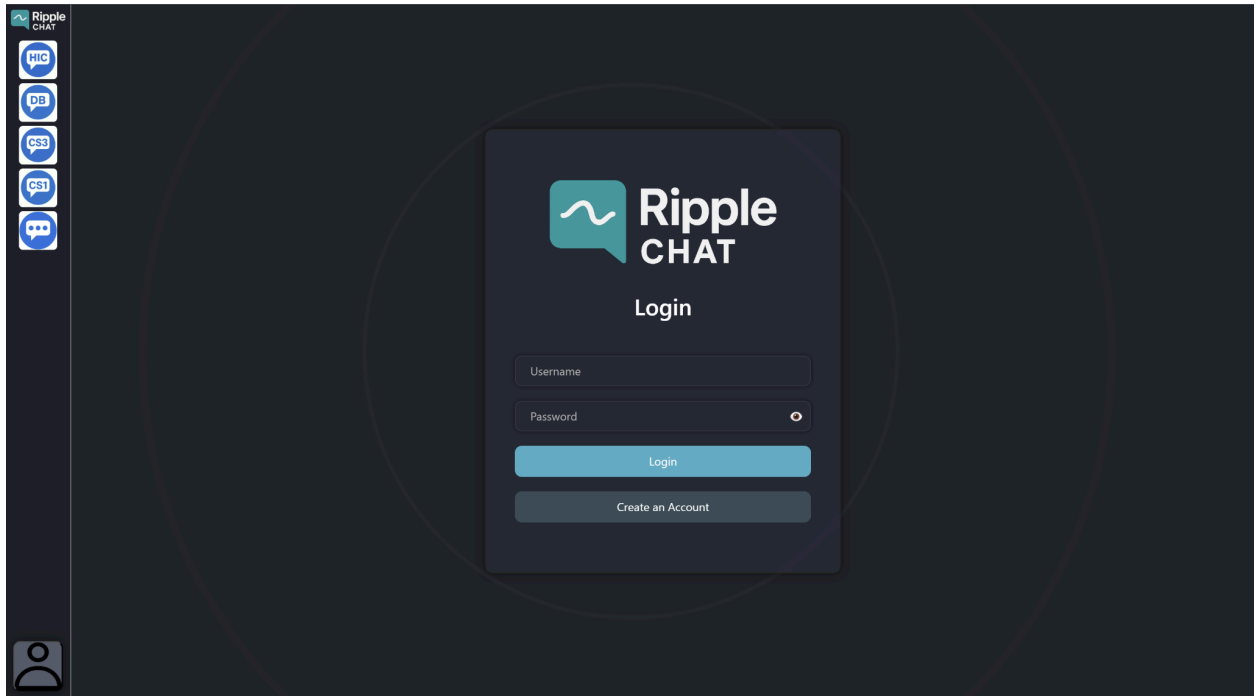
- Frontend: Built with React.js for a dynamic, responsive user interface.
- Backend: Powered by Node.js & Express, which manages voice and video traffic as well as interactions with the database.

Database:

- MongoDB is used to store chat logs and user account information. For security, user passwords are hashed before being saved.

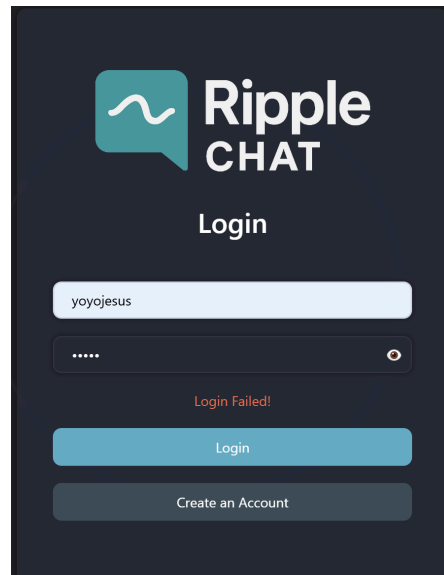
2. Login Page

Our login page would allow a user to login with their account (assuming they have one), or be directed to the registration page (via the 'Create an Account' button)

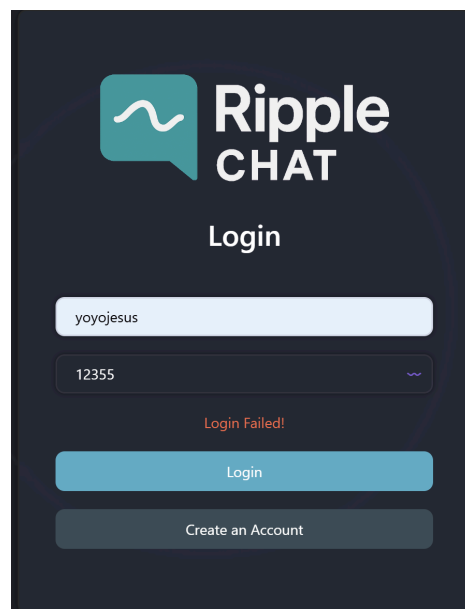


Golden Rules:

- Strive for Consistency: Fonts, colors, and styles are consistent throughout the project (especially the navigation bar since it's the same component). Inputs work as expected and fit into the designed style that matches buttons, etc.
- Shortcuts: Users can use tab and enter to quickly move from input field to input field and login or create an account - mouse use is not directly required.
- Feedback: If a user goes to login and their account doesn't exist or if their password is wrong, they will see a message:



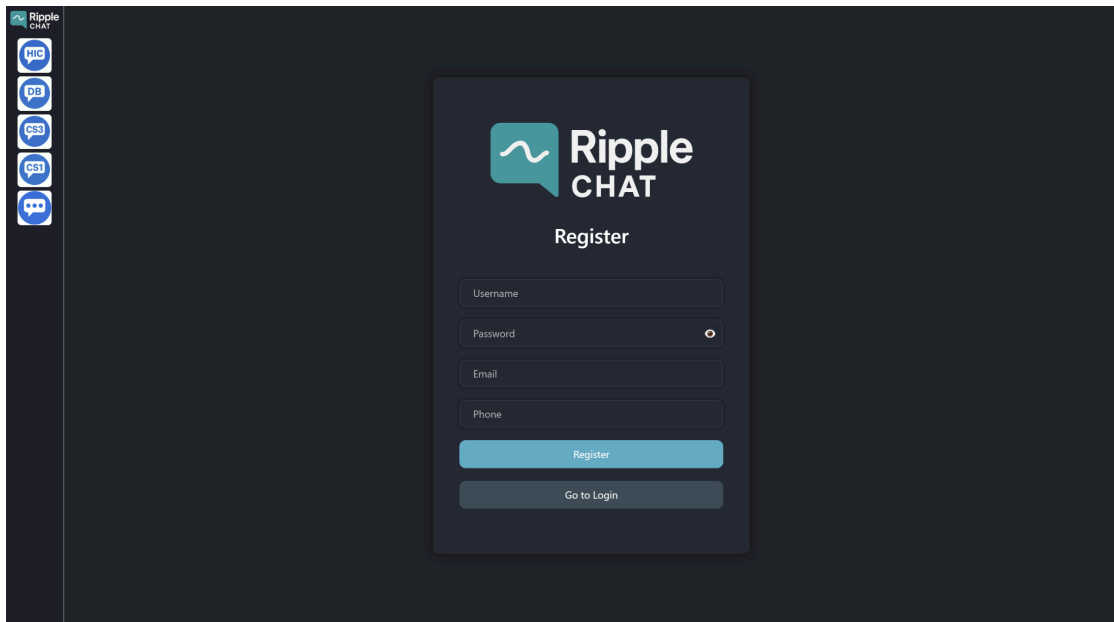
Showing that something they typed is incorrect. The user can then click on the eye to see what password they typed as a way to check if that is the problem:



- Offering informative feedback: The error message is an example of this, as we inform the user if they typed something wrong/if their account doesn't exist. The eye symbol also gives the user the ability to see what they typed into the password box.

- Yield Closure: The 'login' task is simple and clear, as the user inputs their credentials and hits 'login', then are redirected to the profile page upon successful login or told of unsuccessful login. Similarly the 'create an account' button provides a simple task to get the user to the page to make a new account.
- Error Handling/Prevention: Again, the error message when logging in is sufficient here, showing when logins aren't successful.
- Mistake Reversal: The user is able to easily see what they typed and make changes before hitting submit or after seeing login failed errors.
- Internal Locus of control: Users feel in control on this page, they can choose to login or make a new account. The navbar hints as to what they can explore once they login.
- Reduce Short-term memory Load: All of the text on the screen is clear and concise about what it does or what the user needs to do. This section only asks the user for a username and password, so not much complexity.

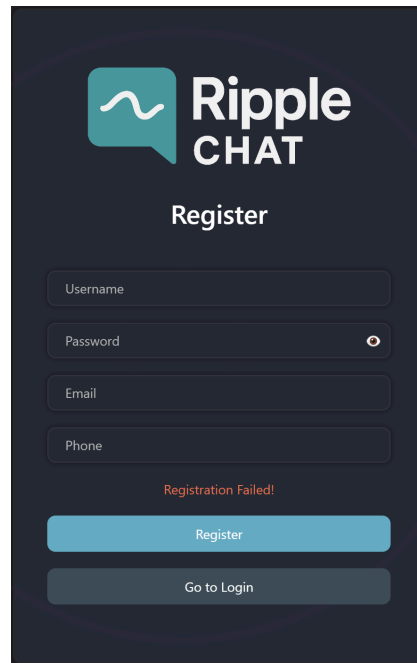
3. Registration Page



On this page, users can register with our system. We only ask for a phone number, email, username, and password. It has very similar functionality to the login page and, upon successful registration, the user will be told that it worked, or told that login failed in that event.

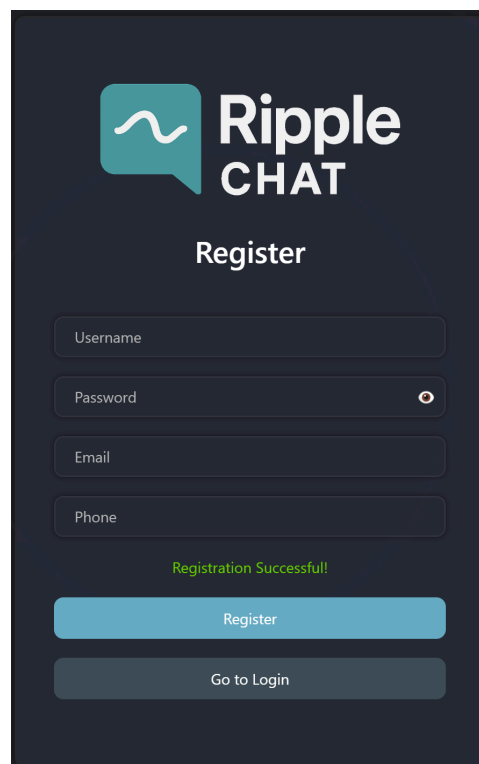
Golden Rules:

- Consistently: Again color and styling is consistent across the entire website. The navbar helps with this too, as it's a separate component brought into the file.
- Shortcuts: Like the login page, users can use tab and enter to navigate through the registration form, use of the mouse is not required.
- Offer Feedback: If the users where to enter invalid information they will see the following:



The image shows a dark-themed registration form for 'Ripple CHAT'. At the top is the logo, which consists of a teal speech bubble with a white wavy line inside, followed by the text 'Ripple CHAT' in white. Below the logo is the word 'Register' in white. The form contains four input fields: 'Username', 'Password', 'Email', and 'Phone'. The 'Password' field has a small eye icon to its right. Below the input fields is a red error message that reads 'Registration Failed!'. At the bottom of the form are two buttons: a teal 'Register' button and a dark grey 'Go to Login' button.

That informs the user that they likely have an account already or that our server is down. The user can also click on the eye to see the password they typed in. On successful login they will see this when the page reloads:

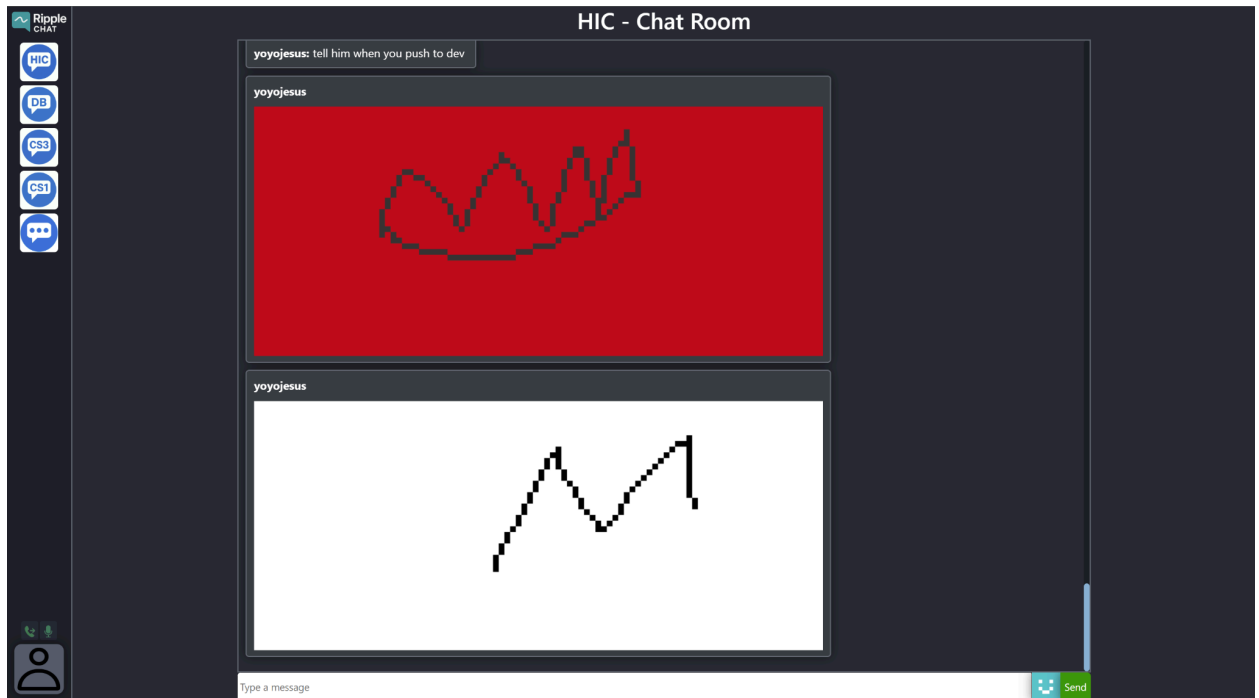


The image shows the same dark-themed registration form for 'Ripple CHAT' as above. However, the error message has been replaced by a green success message that reads 'Registration Successful!'. The rest of the form, including the logo, title, input fields, and buttons, remains the same.

Which would tell the user to now go to login and login with their new credentials.

- Yield Closure: Users complete the account registration by filling out the form and clicking register, they are then told if they registered successfully or if there was an error. The go to login button ends the registration process and moves the user to the login process.
- Error Prevention/Error Handling: The registration failed message make sure the user is informed when something goes wrong and the show password toggle (the eye symbol) helps to make sure that the user can check what they have entered into the box (preventing typos)
- Reversal of Mistakes: The user is able to change fields in the form without hitting submit. Being informed of failed registration also helps the user to see if an account with that username exists already.
- Locus of Control: User is allowed to make the decision to register an account with the system or not. They may also go back to the login page. The user is not forced to do any of these actions.
- Reduce Memory Load: The registration form only asks for a small amount of needed information (username, password, email, and phone number) and all instructions/buttons are clearly labeled. The user doesn't need to have much previous experience with our platform to register.

4. Chat Page(s)



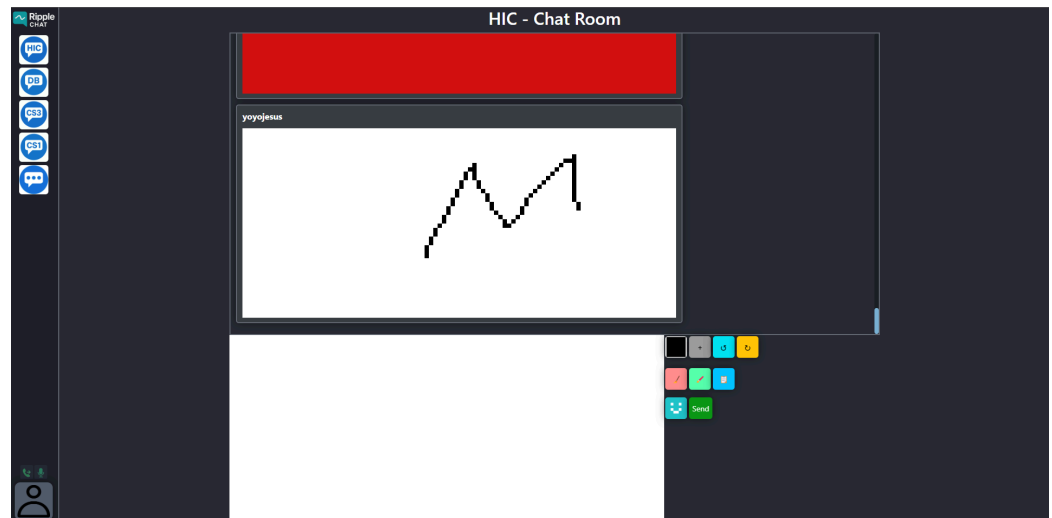
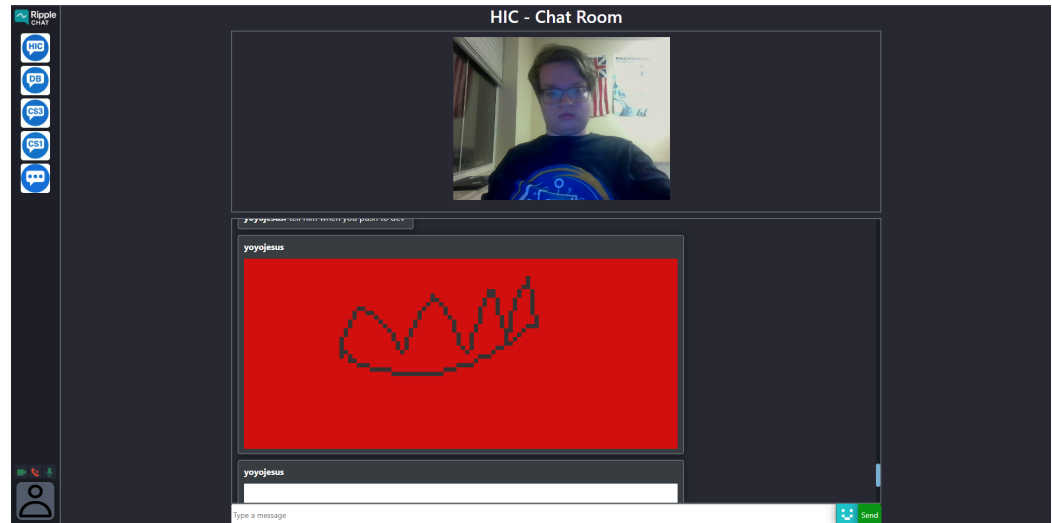
On this page users can send text messages to the public rooms and custom drawn images.

The images part is inspired by PictoChat so it's pixel-based and allows for colors to be selected. This functionality is within the face button next to the send button. This page also allows users to connect to the public voice call for the channel and use their camera if they wish. This functionality appears as the camera, microphone, and phone icons at the bottom of the navigation bar. Moving between chat rooms means clicking on the separate room icons.

Golden Rules:

- Consistency: Just like the previous pages, our design uses similar colors and formatting rules across these chat rooms. With this one, all messages are formatted the same, its user's name on the left and message/image afterwards. White outlines define the chat area, and when the camera opens another box

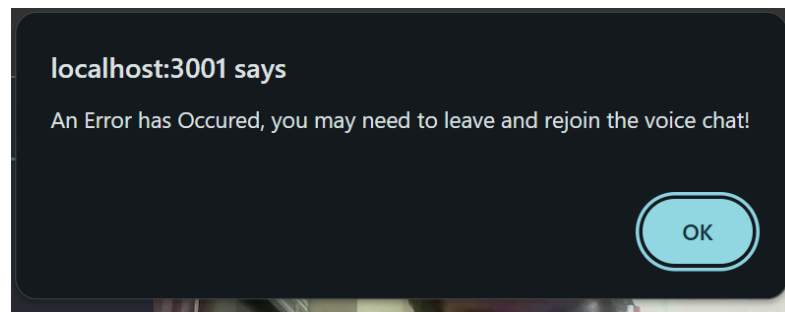
opens to differentiate the two. The canvas and text sections share the same space at the box to replace each other. See the below images for examples of these two properties:



Then all of the canvas buttons follow the same box style to keep everything simple and easy. Icons represent their function.

- Shortcuts: Users can use tabs to get to the text box and enter to send them. They also can use `ctrl + z` to undo drawing on the canvas or `ctrl + y` to redo actions. (note `ctrl + r` cannot be used because it's the refresh command on browsers).

- Feedback: Sent messages and drawings appear instantly in the chat history. When video is turned on, the user can see their camera feed, providing visual feedback that the video is in fact active. In the event that there is a video/voice error, the user will be prompted to leave and rejoin:

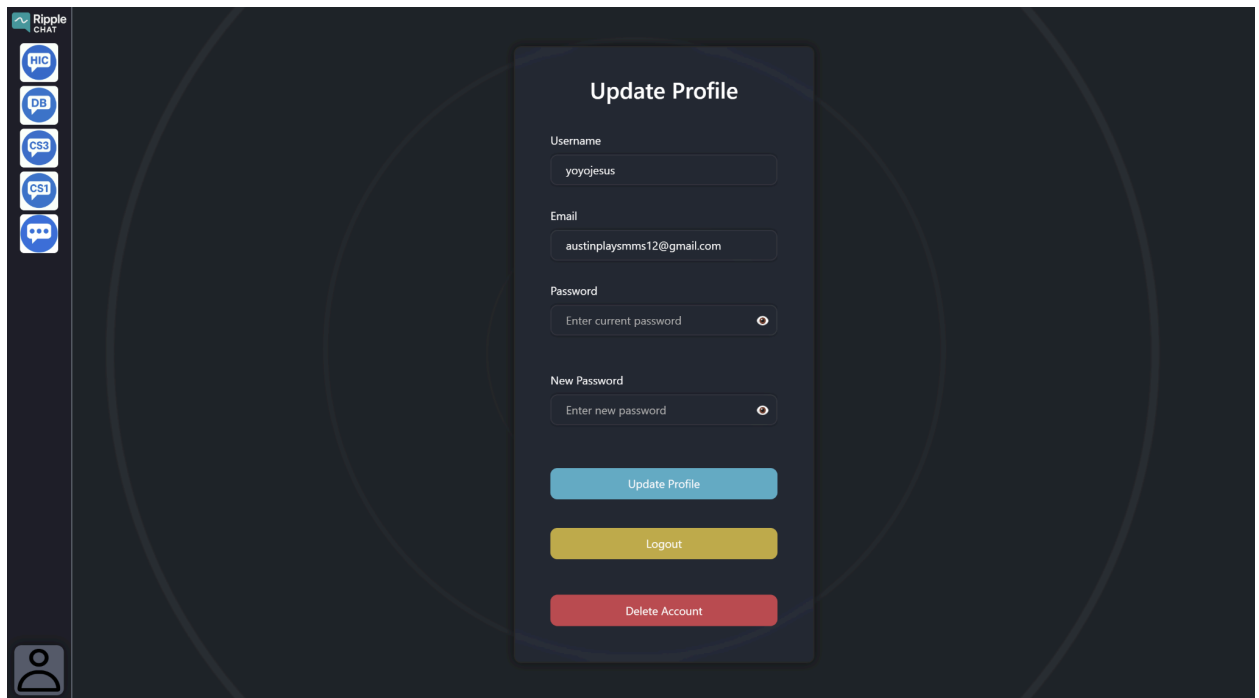


On the canvas, switching colors results in immediate color switching on both the active color button and when the user draws on the canvas.

- Closure: Sending any kind of message or enabling video/mic shows direct results on the screen. Task completion is immediate and simple, it's not possible to leave a task in a half completed state.
- Error Prevention/Error Handling: Text, drawing, and voice/video features are separated cleanly so that users cannot accidentally send the wrong type of input. Error messages are handled and tell the user to take action when they occur, see the video one above. Buttons on the canvas screen are colored code so users will not be confused by what actions do.
- Reversal: In the Canvas section, users can undo and redo tasks as needed to fix mistakes.
- Yield Internal Control: Users have full control, they choose to send messages or drawings, and can choose whether they want to join voice and turn on camera. Users make decisions on which room to be in via the sidebar.

- Reduce Memory Load: Chat room is simple. The layout visually and mentally separates operations; drawing canvas is for sketches, text input for typing, voice controls for the voice rooms. Everything is clearly labeled and visible on the screen so the user doesn't have to remember any complex sequences of operations to take actions.

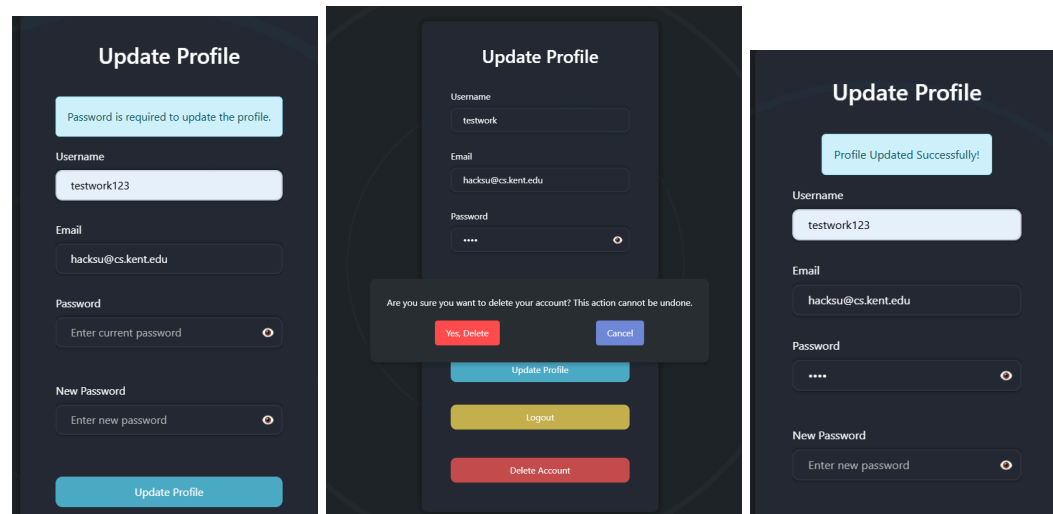
5. Profile Page



The profile is the 'main' page of the application so long as a user is logged in. Here, the user not only shows they are logged in and which account they are logged in as, but they are also able to change important information so long as they enter their password to make these changes. Navigation to this page is done by the gray profile icon at the bottom of the navigation bar.

Golden Rules:

- Consistency: Again like the previous, the colors and formatting are the same, with a dark aesthetic and clearly laid out buttons and form inputs. We are also using the same color-coded button system to denote action type (blue -> update, yellow -> logout, red -> delete account).
- Shortcuts: Tabs can be used to navigate the fields quickly and without mouse input. Enter is then used to submit the changes.
- Feedback: Password fields let the user toggle between concealing and viewing their password in plain text. Pressing the update profile button gives feedback to whether or not the change went through, and further prompts the user to input their password to be able to make changes. Delete account prompts the user twice to confirm account deletion. See the below images:



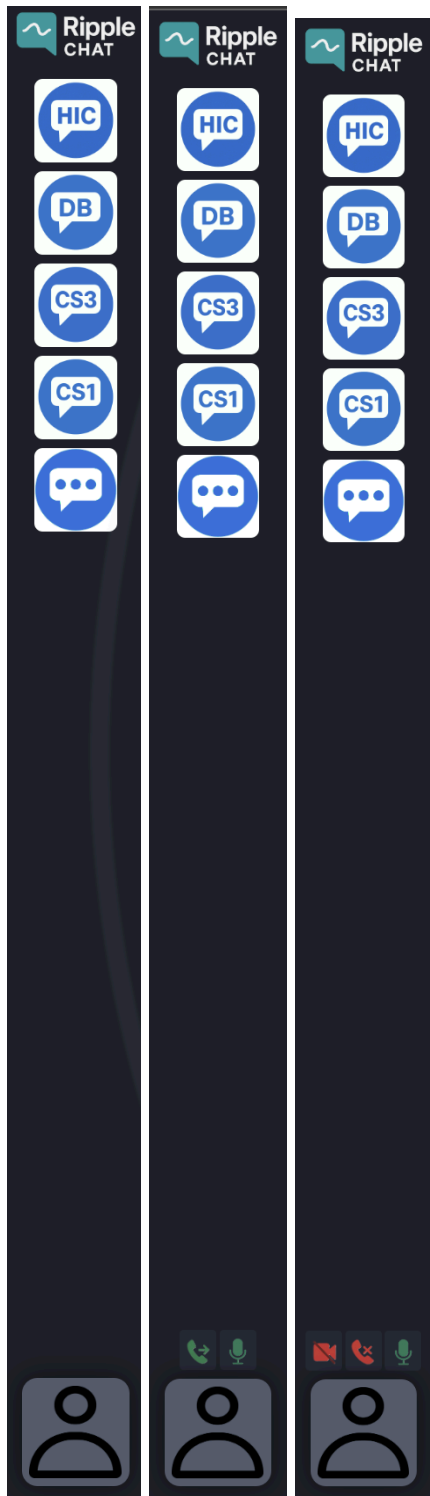
If a user changes their password or deletes their account, they are redirected to the login page - since the token holding their login info is now invalid.

- Closure: Updating a profile, logging out, or deleting an account are fixed tasks. The user inputs info, clicks a button, confirms if necessary, then sees the results. Upon successful completion, the user is told it worked either by a notification or

by redirecting to login. If it fails, the user gets a notification at the top of the screen.

- Error Handling: Users have to enter their current password to update profile information, prevent accidental or malicious changes. The eye icons allow the user to check that they typed things correctly. If an email already exists in the system, the user is prompted for that as well.
- Reversal: Users can stop making changes at any time, and with account deletion, the user is prompted a second time to confirm before the account is deleted, since this is an irreversible action the user must be willing to take.
- Internal Control: Users have full control over the changes they wish to make, they can change their usernames, emails, passwords, or even delete accounts.
- Memory Load: Everything the user needs to know is right there on the screen. It's also clearly labeled to avoid confusion.

6. NavBar



The navbar exists across all pages giving the user the ability to switch between the possible chatrooms, and profile page. When a chat room is accessed, the voice/video call

buttons appear. Specifically the join call and mute/unmute microphone buttons. When the user joins a call, the camera button appears and they may choose to use their camera.

Golden Rules:

- Consistency: Layout again is the same across all pages, especially since this is imported across all pages. Icons are pretty uniform, servers are a nice blue with text that denotes some of the specific rooms. Status icons for voice appear when needed, then show state of functionality.
- Shortcuts: Users are able to navigate between chat pages with tabs and the enter key, and can do the same for the voice controls and getting the profile page.
- Feedback: When a user joins a room, the voice and video controls appear at the bottom. Their color indicates the status of the different voice pieces, it is obvious that if the microphone or camera are red, then those are off and vice versa for green.
- Yield Closure: Clicking a chat room instantly moves the user to that room.
Clicking a join call immediately connects the user to the voice call which not only sets the icon to red, but also will give auditory feedback if others are in there as well.
- Error Prevention/Error Handling: Here we prevent errors by providing clear options and visual states of microphone/video to make it impossible for the user to misclick or talk when they want to be muted.
- Reversal of Actions: Users can easily turn their mic/camera off and on. Leaving a call or chat room is a simple button click away.

- Locus of Control: This navbar gives users full control over what room they are in, what access they provide to the public call, and the ability to change that access at will.
- Reduce Memory Load: Each button is intuitive, speech bubbles for chat rooms, profile icon for profile, call icons for controlling voice/video. Users do not have to remember what room they are in or what data is being sent through the application.