CanvUs

SENG 513 - Web Based Systems
Group 23

Kaitlin de Chastelain Finnigan, Dora Tan, Youup Kim, Kirk Elumir
Prof. Lorans Alabood, TA: Asif Mehmuda

Introduction

The project objective is to create a virtual whiteboarding space in which various users may come together in order to digitally collaborate by using the space to visually represent ideas and processes. *CanvUs* is a collaborative whiteboarding tool that allows users to collaboratively work together on a canvas, save their work and return to it later, as well as adding others to the canvas so they can make changes and work together. The major features of the project include creating a new canvas, viewing existing canvases, editing a canvas, collaboratively editing the canvas, chatting with others who are on the canvas, adding other users to canvases, and downloading the canvas.

Users upon logging into the application can view all the canvases that they have created or have been added to. Creating a new canvas allows users to give their canvas a name, after which opens their newly created canvas. When users have an active canvas open, they are able to edit the canvas by drawing on it, changing the brush colour and size, as well as erasing unwanted work. Additionally, users can see the other users' edits in real time. In order to collaborate with others, users can add other registered users to the canvas and communicate with active canvas users using the built-in chat function. Lastly, users may download the current canvas state directly from the website.

This document will cover the background motivations, the goals of the project, what users can do, how the system works, and what changed from mockups to implementation. Additionally, this document will cover how this project fits into the requirements, future work we would like to see this project expanded into, and lessons learned along the way.

Background

Do you ever find yourself in a meeting seeking to explain something that just needs to be drawn out? You open MS Paint and show what you mean, then suddenly someone has an additional thought, but they can't simply add to your diagram collaboratively, so they try their best to explain what they mean as you draw. It's painstaking, frustrating and unsustainable long-term. CanvUs takes these challenges and addresses them with our collaborative canvas. Your ideas can be drawn out and stored in our system for collaborative review.

Use cases for this project are:

1. Remote Classroom Support - for students learning out of a classroom environment, activities can be done in the collaborative whiteboard.
2. Brainstorming - one of the tools of a good brainstorming session is drawing.
3. Artistic Planning - from show directions and stage setup to overall planning, this is a fantastic collaborative tool to employ.
4. Business Strategy and Planning - visually lay out your plan and come back to it when it changes to make further edits.
5. Collaborative Artwork - CanvUs provides a flexible medium for artists to collaboratively create digital pieces.

Users:

1. Teachers - Teachers are able to create and manage canvases for their students and check in on each group's progress in real time.
2. Students - Students can create projects together in real time even when working remotely.
3. Designers - Getting an initial mockup drawn out and allowing for direct real-time feedback from clients.
4. Artists - Artists can freely utilize the canvas spaces to express their creativity anytime they want and with whomever they want.
5. Developers - Developers would easily be able to plan, meet, work and collaborate remotely with each other.

What motivated us to complete this project: with the knowledge of React and Socket IO, creating a real-time collaborative canvas page seemed like an interesting project to undertake. Having a product that can be used on a daily basis like Google docs with the use of sockets for transferring data from client to server made this project a perfect solution.

Project Goals

The overall goals for the project were to:

1. Allow users to draw and place sticky notes on a collaborative canvas.
2. Have the canvas state saved to be reviewed and edited at later dates.
3. Have a chat feature for users on the same canvas to communicate.
4. Have a login account and sign up system for all users.
5. Display Canvases that users have been added to or created.
6. Add other users to canvases.
7. Erase unwanted drawings.
8. Allow users to export their canvases to save specific states of their canvases.

Project Accomplishments

Users are able to accomplish the following with the project:

1. Create a new account with CanvUs.
2. Login to a previously created account.
3. View previously created and shared canvases.
4. Create a new canvas with a unique name.
5. Add other users to the canvas.
6. Draw on the Canvas.
7. Change the colour of their brush strokes.
8. Change the size of the brush strokes.
9. Chat with other active users.
10. Export their saved canvases as a PNG file.

Example 1:

Katie is a new user to CanvUs, she is able to navigate to the login page and click sign up, creating a username and password. After a successful signup, she is taken to the main dashboard and is able to create a new canvas giving it a name and proceeds to draw a diagram to be shown later. She knows her friend Dave will also need to edit the diagram so she adds him to the canvas.

Dave has been using the system previously and logs in to see he has a new canvas in his list. He opens it while Katie is still working on it, Dave chooses a new colour for his additions to the diagram and chats with Katie as they work together.

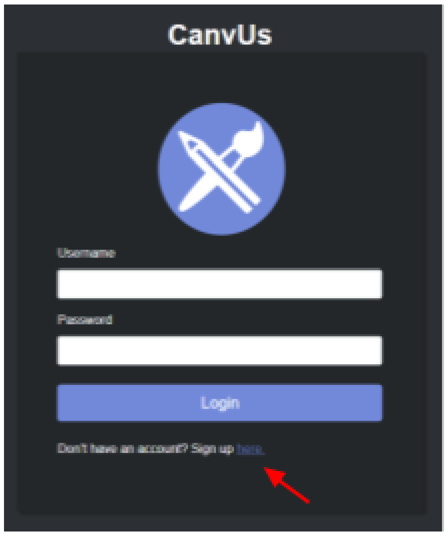
Example 2:

Chelsea is a teacher and she needs to set up an activity for her students, she is able to create 5 different groups with 5 different canvases adding each student to the canvas assigned to their groups. The students are able to log in and collaborate with their group by drawing together. Chelsea is able to go into each canvas as needed to moderate and instruct using the chat.

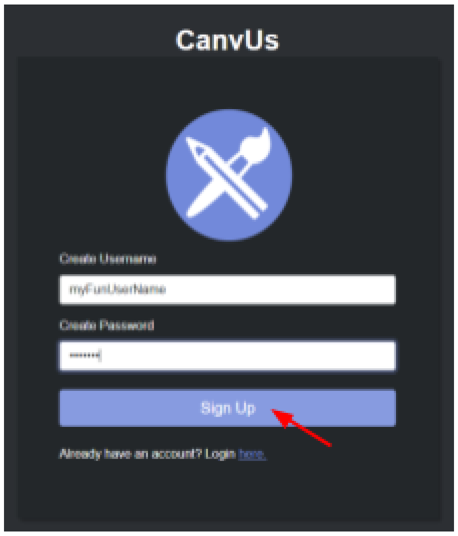
Project Description

Create a New Account with CanvUs

From the landing page, click the *sign up here* button on the bottom of the login window. From there, fill out your desired username and password. If the username has not been taken already then you will be taken to the All Canvas Page. If the username has been taken already then an error message will appear

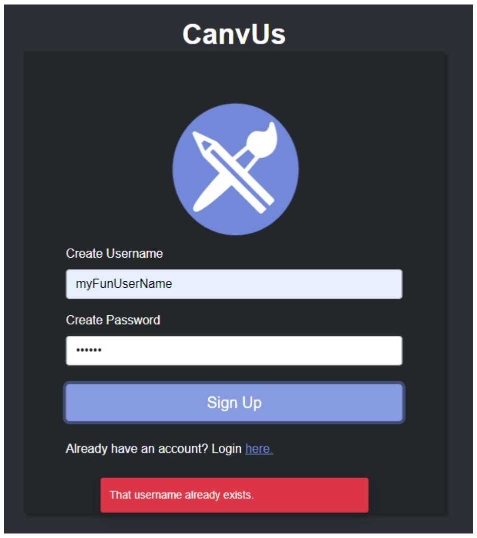


Don't have an account? Sign up here.



myFunUserName

Already have an account? Login here.



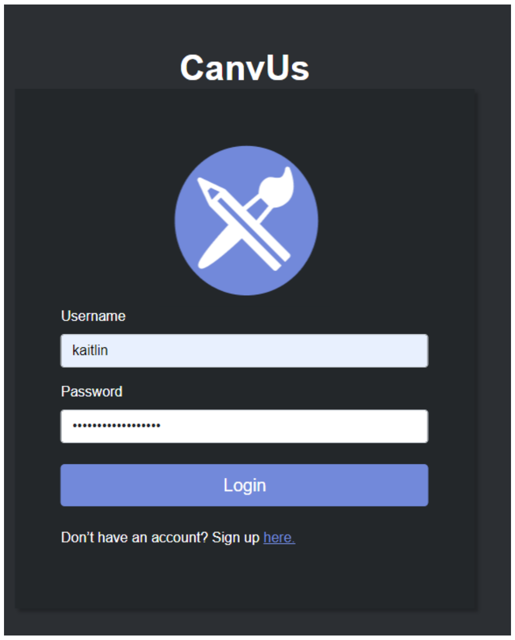
myFunUserName

Already have an account? Login here.

That username already exists.

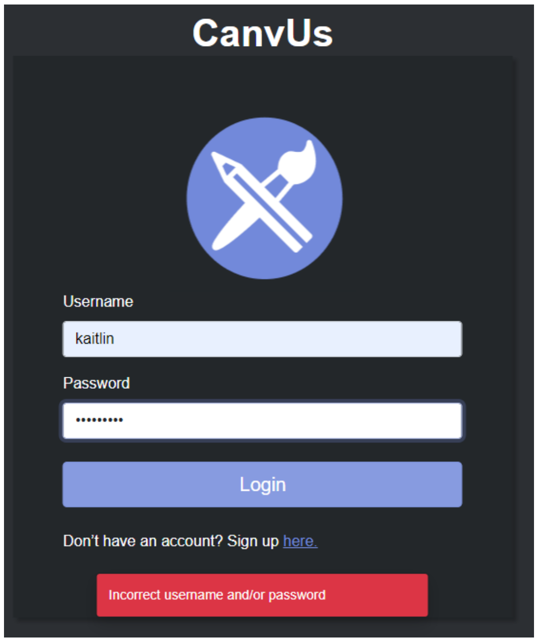
Login to a Previously Created Account

If you already have an account, you may log in using the login form on the landing page. If your username is incorrect then you will see an error message appear, otherwise you will be taken to the All Canvases Page.



kaitlin

Don't have an account? Sign up here.



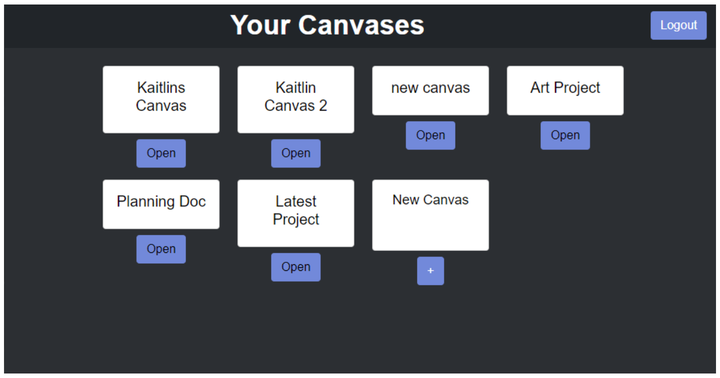
kaitlin

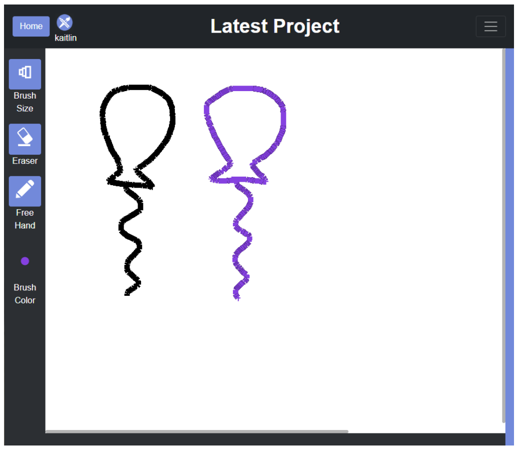
Don't have an account? Sign up here.

Incorrect username and/or password

View/Open Previously Created Canvases

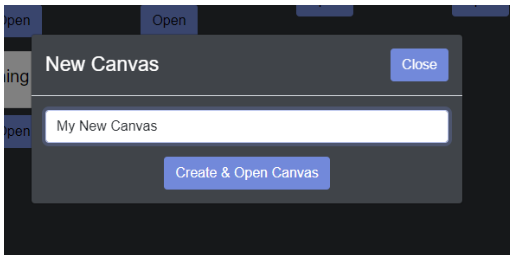
If you have created canvases previously, they will be displayed on the All Canvases Page. Click the *Open* button under a canvas to view it. The example below shows the opening of the project *Latest Project*.

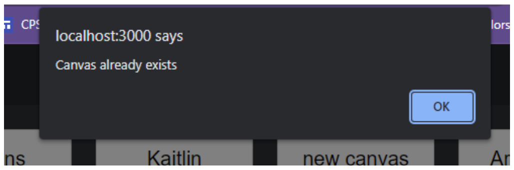




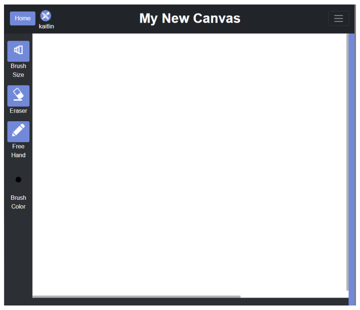
Create a New Canvas with a Unique Name

From the All Canvases Page, selecting the (+) button under New Canvas will open the new canvas modal, where you can create a new canvas with a unique name. If the name has been used already, then an alert will pop up, if not then a blank canvas will appear.



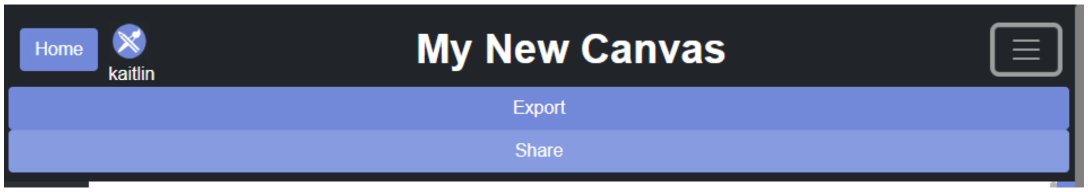

localhost:3000 says

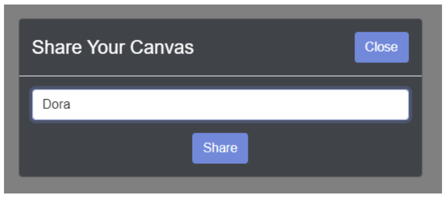
Canvas already exists

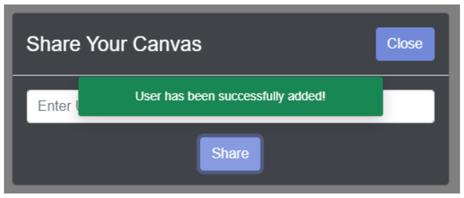


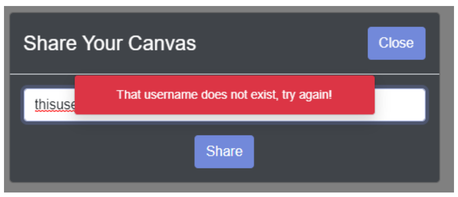
Add Other Users to the Canvas

To add another user to a Canvas, click the hamburger menu on the right and click *Share*. This will open a modal which will allow you to add another username to the canvas. Upon success you will see a confirmation, if the user doesn't exist you will see an error.



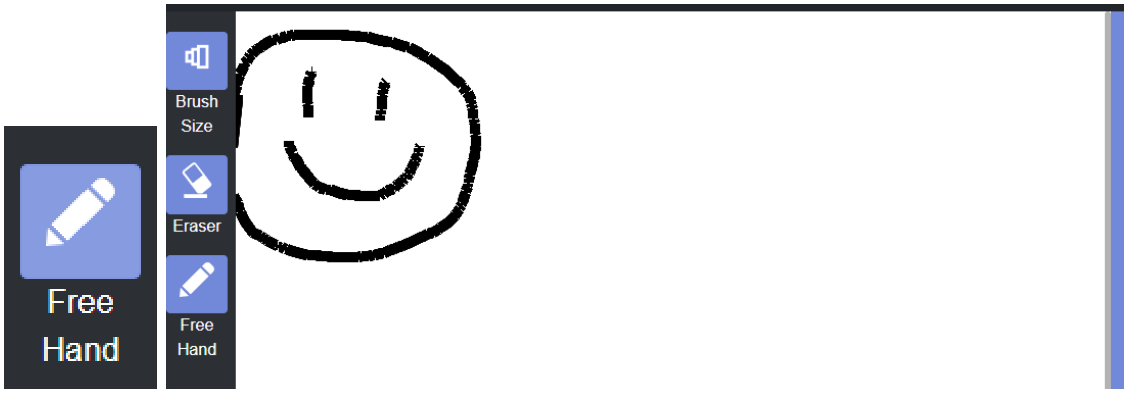



User has been successfully added!


That username does not exist, try again!

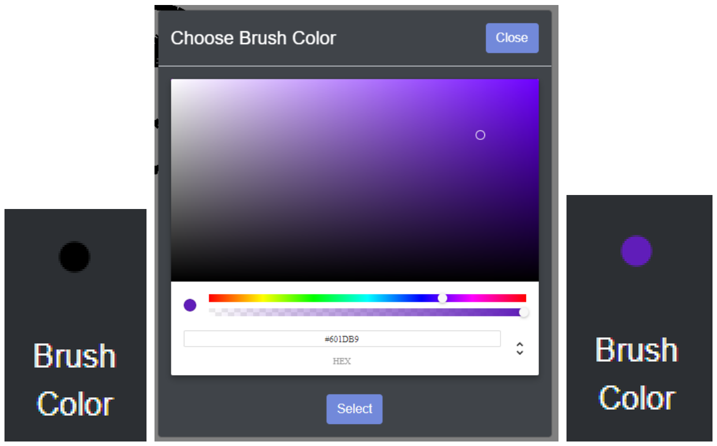
Draw on the Canvas

When on the active canvas, select the *Free Hand* tool then drag the cursor with the left mouse button down around to draw on the canvas.



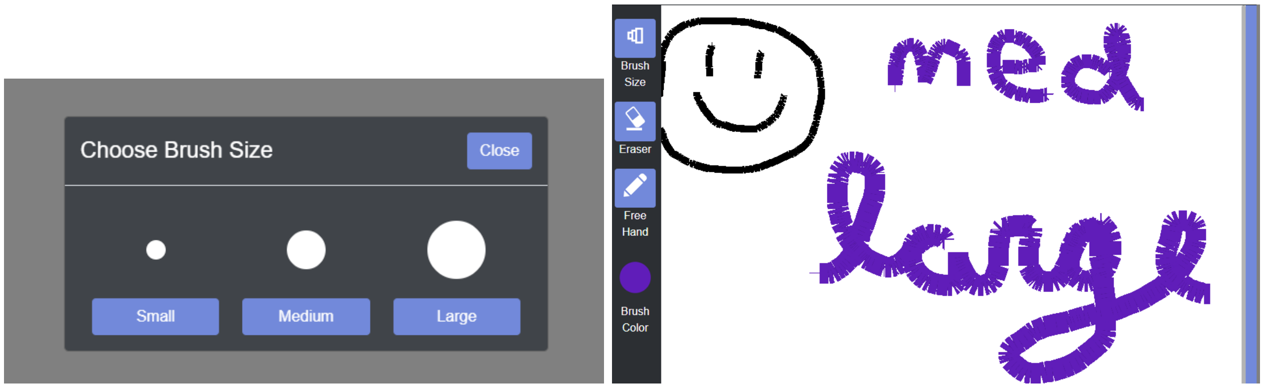
Change the Colour of Brush Strokes

Select the *Brush Color* button to open a colour choosing modal, select your colour of choice and the updated colour will appear on as the new brush colour.



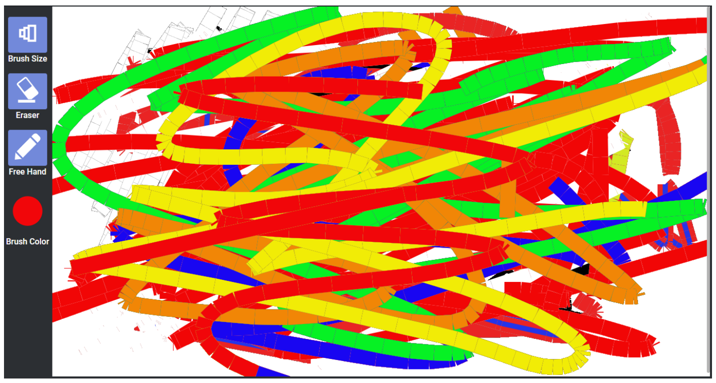
Change the Size of Brush Strokes

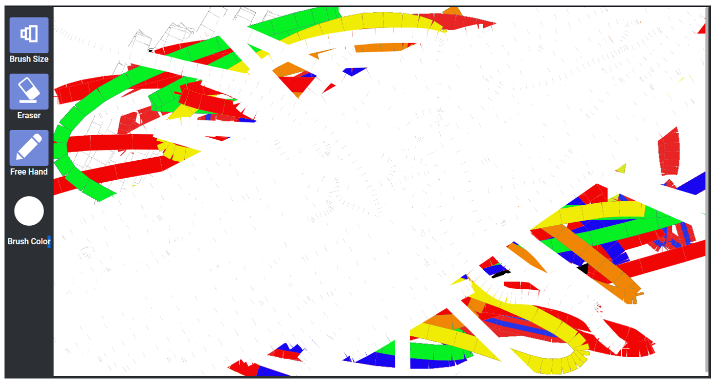
Select the *Brush Size* Button, this opens up a modal that allows you to choose the brush stroke size. Small, Medium and Large are depicted in the right image.



Erase Unwanted Work

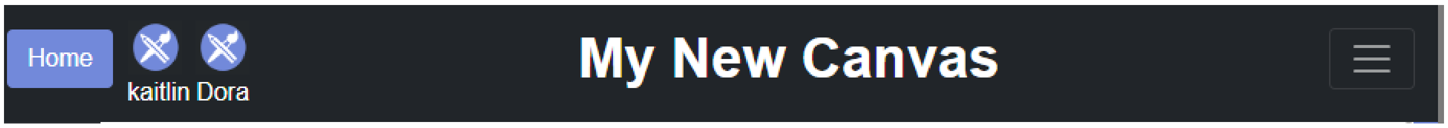
Selecting the *Eraser* tool changes your brush color to white. Since the default colour of the canvas is white, this will “erase” any brush strokes you go over with your brush.

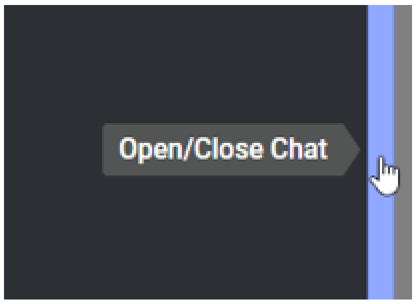


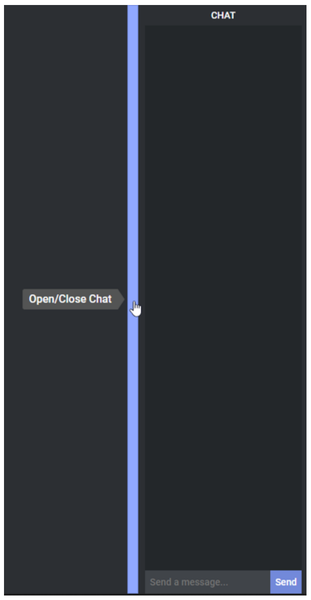


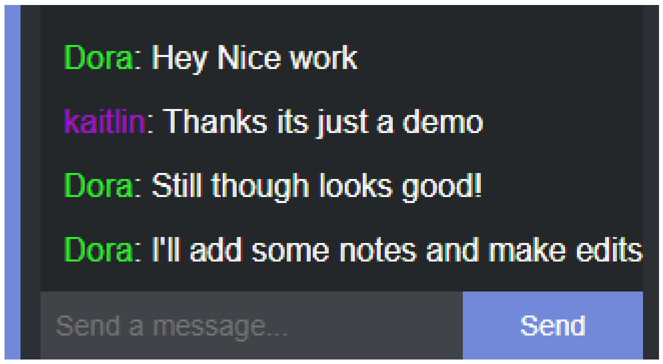
Chat with other Active Users on the Canvas

On the right side of the canvas is a vertical blue bar that extends across the height of the active canvas screen below the header. Clicking it opens up the chat window, which allows you to communicate with any active users on the canvas that you are working on. Your brush colour will determine your chat colour when you send a message. You can view who is actively on the canvas by looking at the top bar.









Dora: Hey Nice work

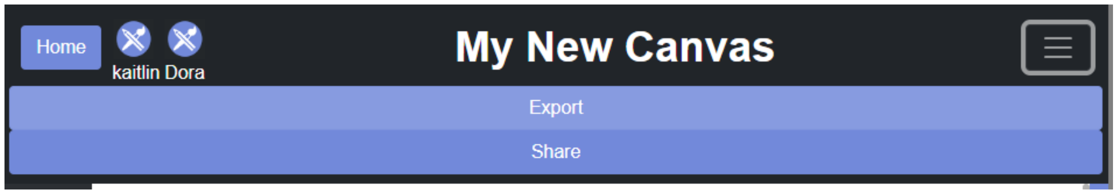
kaitlin: Thanks its just a demo

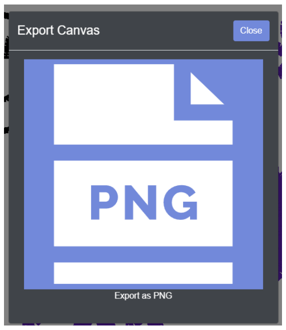
Dora: Still though looks good!

Dora: I'll add some notes and make edits

Export Saved Canvases as a PNG File

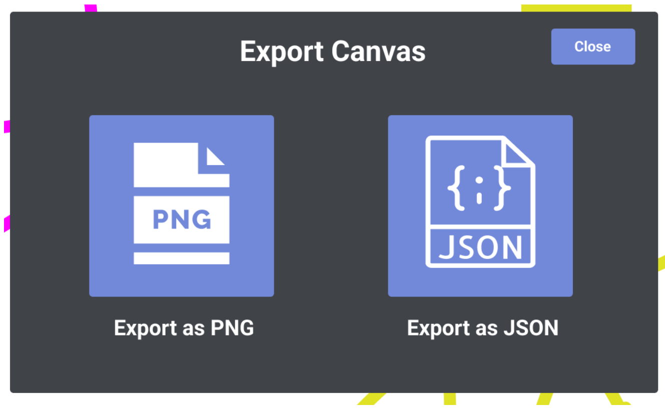
In the hamburger menu at the top right corner, select the *Export* button and select PNG file button to export your canvas.

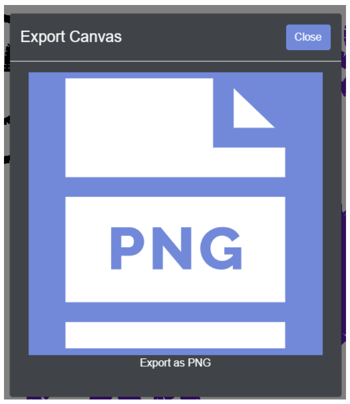




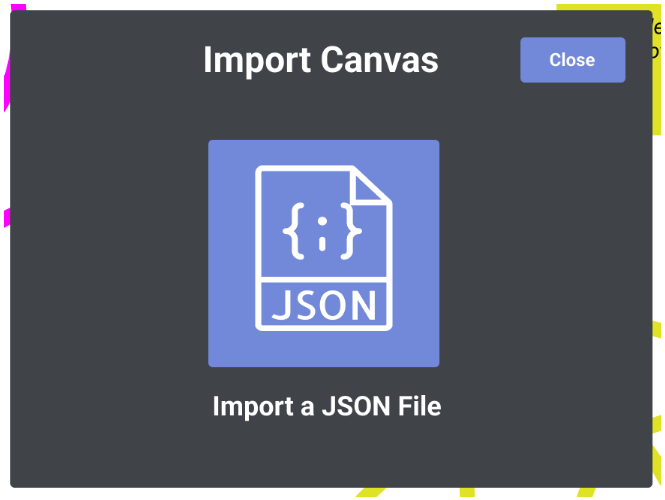
Mockups vs. Final Project

1. Import feature was removed as the saved state of a canvas was changed from a JSON implementation to a PNG implementation and importing did not make sense. *Export as JSON* and *Import* functionality was removed.





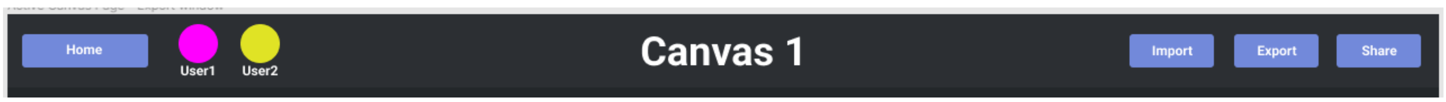
Mockup Actual



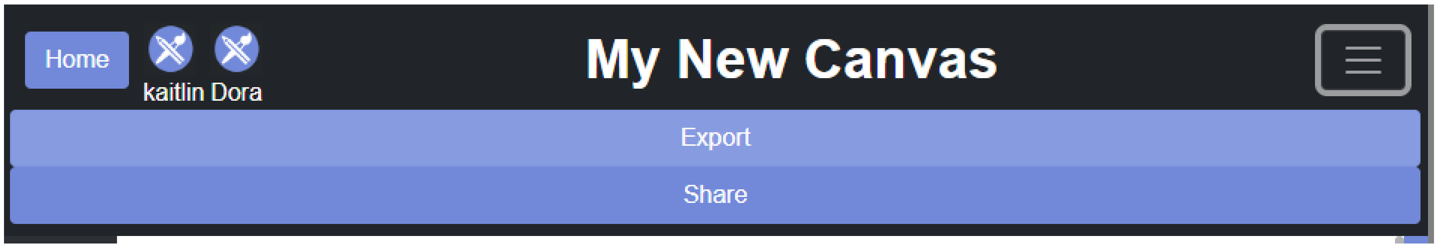
<Removed in Implementation>

Mockup Actual

2. Regarding the header bar, the *Export* and *Share* button were moved to a menu under the top menu bar to be more compatible with the mobile version. Additionally, instead of brush colours, the active users are displayed with our logo.

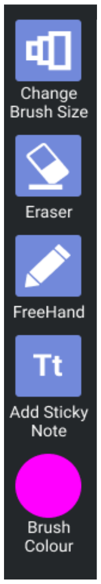


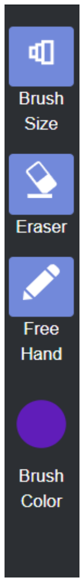
Mockup



Actual

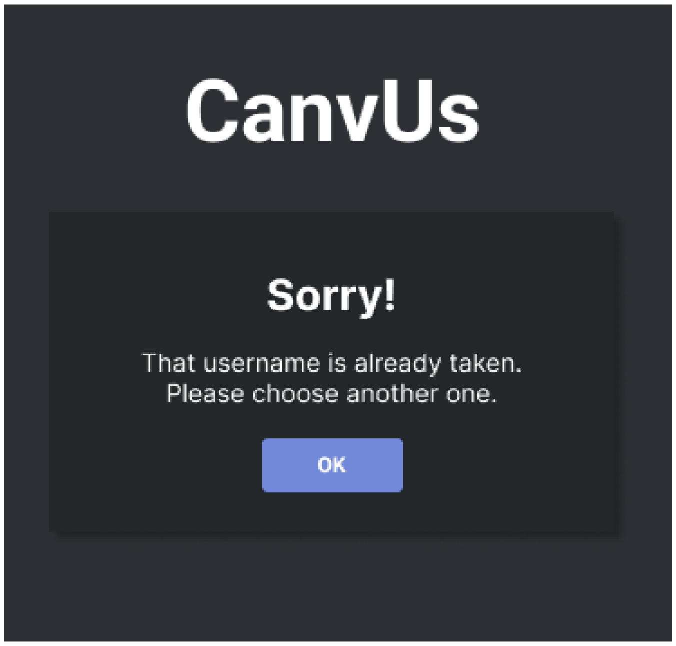
3. *Add Sticky Note* functionality was removed.





Mockup Actual

4. The messages for Sign Up page errors were standardized.

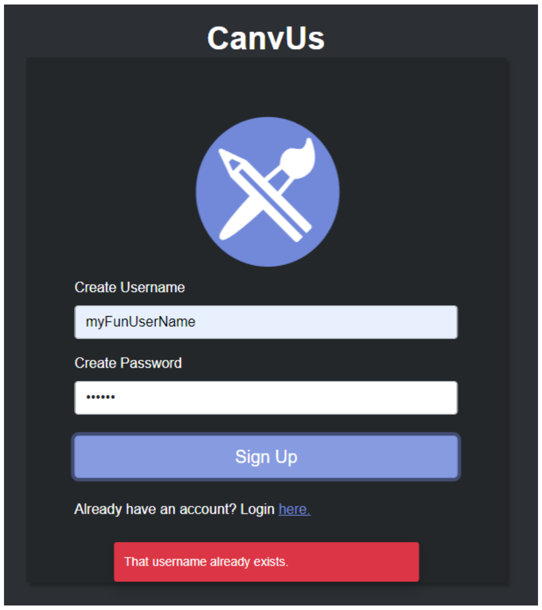


CanvUs

Sorry!

That username is already taken. Please choose another one.

OK



CanvUs

Create Username

myFunUserName

Create Password

••••••

Sign Up

Already have an account? Login here.

That username already exists.

Mockup Actual

Project Requirements

Requirement	**Requirement Met?**	**Explanation**
Client Side	Yes	Our application consists of two main views that a user interacts with, the All Canvases Page and the Active Canvas Page. The All Canvases Page serves as a user point of entry to choose which canvas they want to work on. The Active Canvas Page is where a majority of the functionality in drawing, chatting, and collaborating are done in the application. The client side was implemented with React JS and is compatible on both Chrome and Firefox.
Mobile Support	Yes	Our application is responsive to mobile screens, we used a mobile friendly approach to achieve this. The chat is collapsed into a separate view which can be accessed as needed. Users are able to interact with the same tools and draw and view the canvas as needed.
Server Side	Yes	Our application uses NodeJs for our backend, it utilizes websockets for asynchronous functionality. This allows us to create separate rooms for separate canvases, persist user information to our database, and update multiple users for collaborative work environments.
Multiple User Support	Yes	Our application allows multiple people to be signed in at once. Users are able to set a password to keep their account secure and are authenticated upon login.
Persistence	Yes	Users may leave a canvas and return to it during the same login session or after having logged out and logged back in again. Both user information and canvas information is persisted to our MongoDB.
User Interaction	Yes	Users who have been added to the same canvas can see changes a person has made previously when they load the canvas, as well as see live updates if they are on the canvas at the same time. Additionally, they are able to chat with other online users in the same canvas.

Technologies Used

React

The react framework consists of the majority of the frontend UI and was used in order to efficiently separate the project into reusable components which allowed for easily organisable and recyclable code.

How was This Used: React was the underlying framework used to create the frontend UI by allowing us to build separate components that were easily utilized and put together in order to achieve the resulting layers of pages that the frontend consists of.

How to Install React: React provides a tool called *npm init react-app my-app* which creates the initial structure needed for a react project. This can be installed within the components having the import statement *import React from "react";* will allow you to utilize the react framework.

React Bootstrap

React Bootstrap was utilized to simplify a majority of the components. Rather than making them from scratch, react bootstrap components were utilized to simplify the implementation of frontend components and add accessibility for users and scalability to suit different window and mobile screen sizes.

How was This Used: React Bootstrap was used in the following components:
Grid supporting the All Canvases Page, Header Bar, Error Messages, Modals, Login & Sign Up Page, Buttons

How to Install: React Bootstrap can be added through npm: *npm install react-bootstrap*
See here for further documentation: https://react-bootstrap.github.io/

Socket IO

Socket IO was used to allow users to connect to the application and the backend logic of the application. It allowed multiple people to connect to the application, work simultaneously on the same canvas or different canvases and login as needed.

How was This Used: A client socket connection is used on the frontend, and a listening socket connection on the server is also used to handle multiple connecting clients.

How to Install: Socket IO can be installed through npm: *npm install socket-io*
See here for further documentation: https://socket.io/

BCrypt

BCrypt was used to support encryption for passwords in the database, it ensures that sensitive information is not persisted for others to see.

How was This Used: This is used in the backend system to authenticate the user.

How to Install: BCrypt can be installed through npm: *npm install bcryptjs*
See here for further documentation: https://www.npmjs.com/package/bcryptjs

Node JS

Node JS provides the backend server and allows it to run and service our clients.

How was This Used: This is used as the main framework that allows the backend application to work. It runs the server, enabling asynchronous connection to our real time system for all clients.

How to Install: Node JS can be installed from: https://nodejs.org/en/download/

MongoDB

MongoDB was used as our projects database for persisting user login information and canvas information. MongoDB was chosen as it allowed us to create a flexible remote database that we could use for our project. MongoDB provided us with a flexible schema which allowed us to pivot our design midway through our project.

How was This Used: This was used on the backend server, the server made calls to insert, update, and get users from the system. Specifically, Creating users, checking if a username was taken already, retrieving users information for login authentication, getting all canvases per user, adding a user to a canvas.

How to Install: A free tier remote cluster can be created on https://www.mongodb.com/. To connect to the database installing the mongodb module with *npm install mongodb* and then adding the client *var MongoClient = require('mongodb').MongoClient;* allows node js to interact with your database.

Future Work

Given more time to work on this project we would implement the following additions to our project:

1. Implement Sticky notes: this would allow text to appear on the whiteboard.
2. Implement Different shapes: have predetermined shapes users may utilize.
3. Support changing the background of the canvas: This would allow users to set the background colour when using the database.
4. Supporting session id's to allow for maintaining login upon refresh, this could include something akin to this tutorial: https://socket.io/get-started/private-messaging-part-2/
5. We would fix the known bugs.
6. Add notices to the chat when someone enters or leaves the canvas.

Lesson Learned

1. When creating a canvas, you can store the canvas as an image and not as a list of separate components
2. When rendering canvases as PNG's, we had to save the image on every change which was creating more load on the database. A better method would have been to implement auto-saving every 5 seconds or a manual save button for the user.

Known Bugs

1. Session does not persist on reload of the window.
2. Refreshing page duplicates socket connections.
3. Chat messages double up when a user leaves and re enters a canvas.
4. Share button success and fail notifications persist upon reopening the button if the user does not wait for notifications to hide before closing modal.

Conclusion

Overall we were able to create a functional system that allows users to create an account, sign into already created accounts, collaboratively draw, chat, save their work and revisit it upon logging in again. We were able to complete our project with minimal changes to our mockups. Our project meets all requirements as outlined in the project requirements document. We accomplished this by using React, Node, BCrypt, MongoDB, and SocketIO. We've identified areas of improvement for future work, including sticky notes, custom shapes and custom backgrounds.