Study_Room

Tyler Cako, Alan Kerdman, Jaron Rothbaum, Jake Tucker, and Josh Wright

Project Description

Our motivation for this project comes from the difficult experiences many of us have had being in huge classes where we don't know any of our peers. Study Room was designed to make it easier to meet other CU students and get information about classes under these circumstances. The app does this by creating group chats where all of the students in a class can communicate and get to know each other in an informal setting. Any student can join by registering with their email, signing in, and adding their class list (only undergrad CSCI classes are supported at this time). After just those simple steps, students can instantly message all of their classmates in any of their classes if they have a question, an announcement, or just want to make friends. This functionality is the core of our project. All messages are stored persistently in our database and are all sent to each user in the class when they load the page. However, we didn't want users to have to refresh their chats to see new messages, so we also used websockets to automatically propagate new messages to all users who currently have the chat open.

Project Tracker - https://github.com/users/Tyler-Cako/projects/3/views/1

Video - video1660185575.mp4
VCS - https://github.com/Tyler-Cako/Study_Room

Contributions

Tyler

- Wrote the initial project structure with Typescript, created the database, created a CI/CD pipeline, and deployed the application.

Jaron

- I worked on a lot of the frontend stuff, like wireframes and HTML. Also I did the tests and a little backend like the add class route.

assignee:jaronr11 10 Discard Sav

	Title	Linked pull requests
1	Write Register Class Tests #77	
2	Write tests for login API Endpoint #35	
3	Login Page HTML #13	
4	Register Page HTML #15	#28
5	Create Class Registration Node Endpoint #38	
6	Add Class HTML #18	
7	Create Login Wireframe #19	
8	Create Group Chat Wireframe #20	
9	Create Register Wireframe #21	
10	Create Class Registration Wireframe #22	

You can use Control + Space to add an item

Jake

- I wrote a majority of the websocket functionality. I used a library called “socket.io” to handle new message events, user join and disconnect events, and errors. I also wrote an endpoint to save message data to the database.

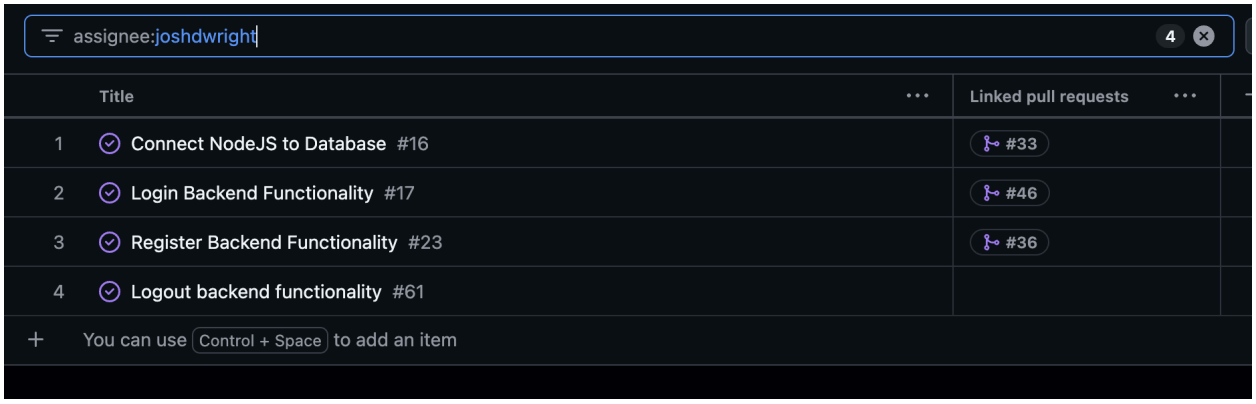
assignee:jTucker583 13

	Title	Linked pull requests
1	Deploy Database #78	
2	Create NodeJS docker container in dockerfile #12	#29
3	Build WebSocket Functionality on Frontend #40	
4	write API endpoint to get user class list #67	
5	Connect NodeJS to Database #16	#33
6	Build WebSocket Functionality on Backend #25	#44
7	Write API to save message data to DB #63	
8	Write API to get room message data from DB #64	
9	Modify chatRoom.js to work with new endpoints #65	
10	Write API endpoint to get user session data #66	
11	fix requirement to build upon code changes #31	#32
12	fixed nodemon error #32	
13	Db ws devel #68	

You can use Control + Space to add an item

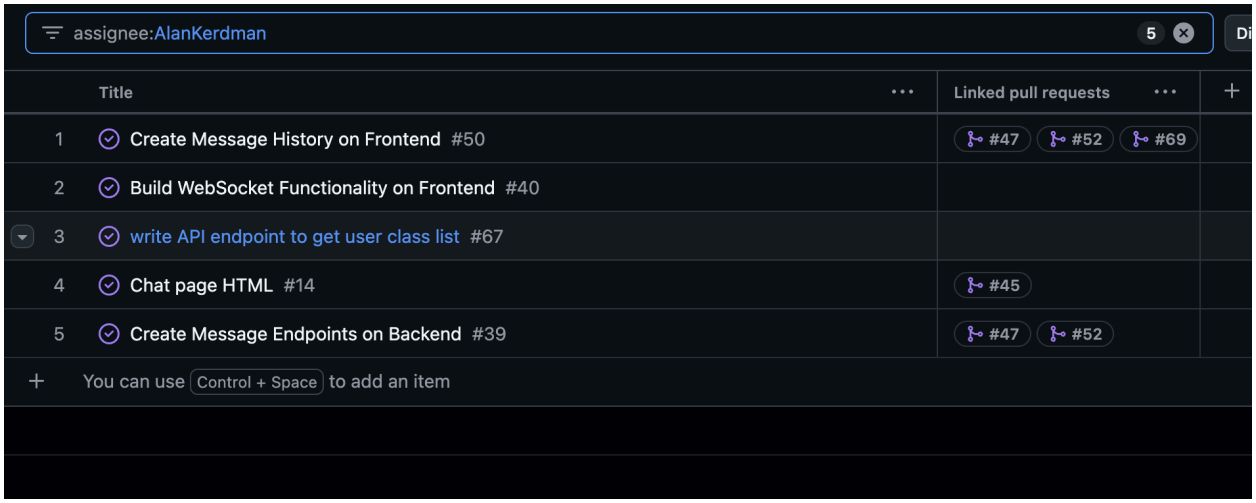
Josh

I worked on the API routes for the register and login pages as well as the logout functionality. I also helped with other general fixes on the backend and made sure the Express Session worked properly.

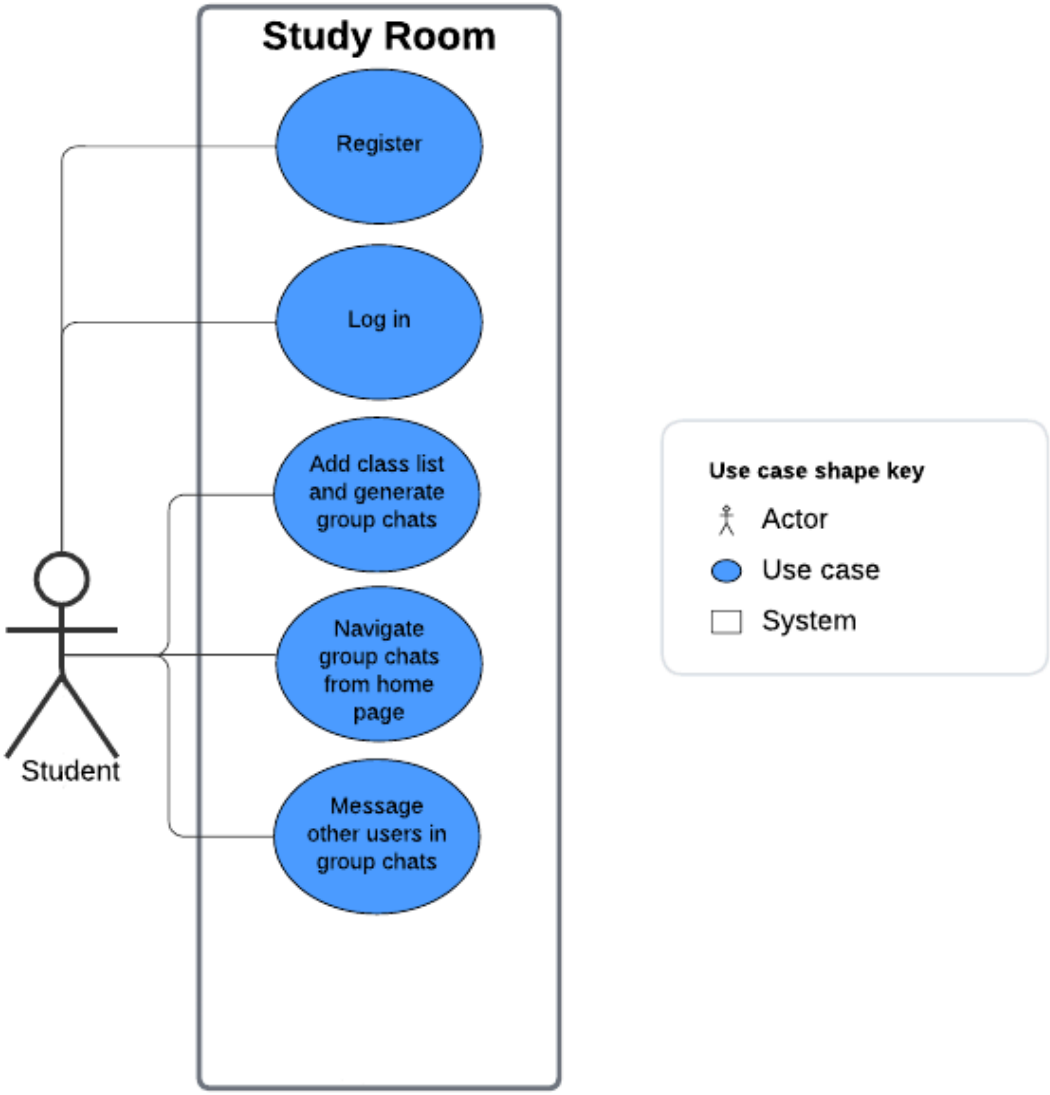


Alan

I made the chat page with HTML (including dynamic rendering with handlebars). I also wrote the Javascript for the chat page, including working with Jake to implement websockets. In addition, I wrote multiple API routes, most of which were associated with the chat page.

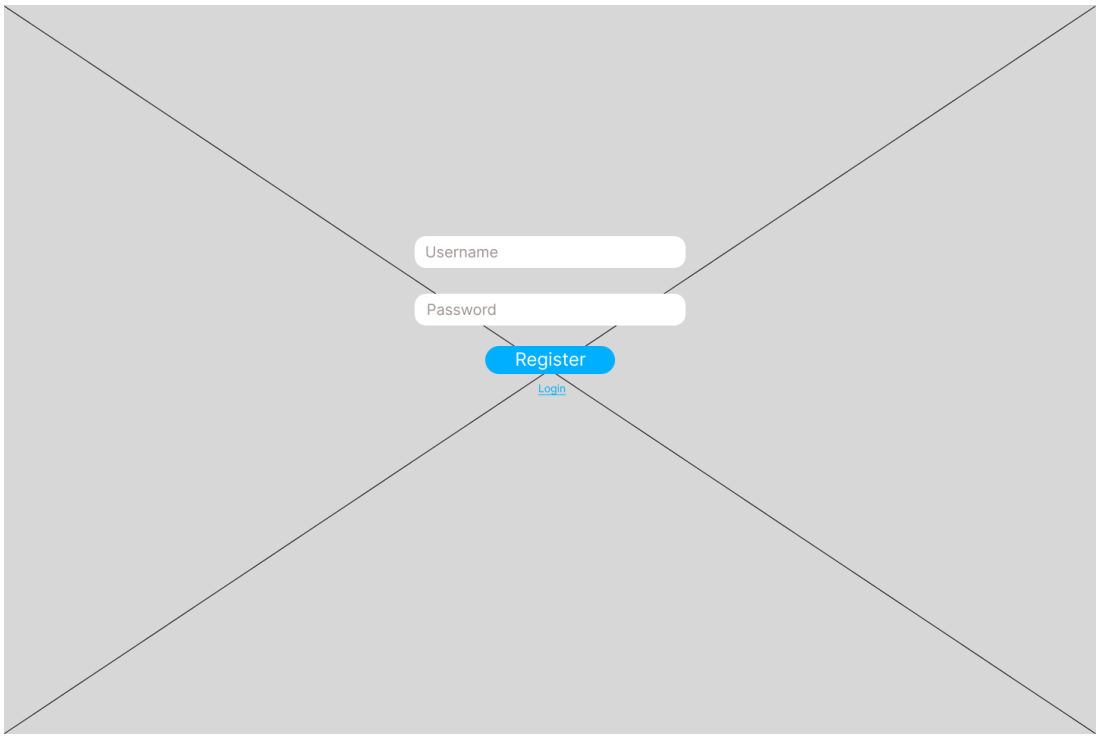


Use Case Diagram

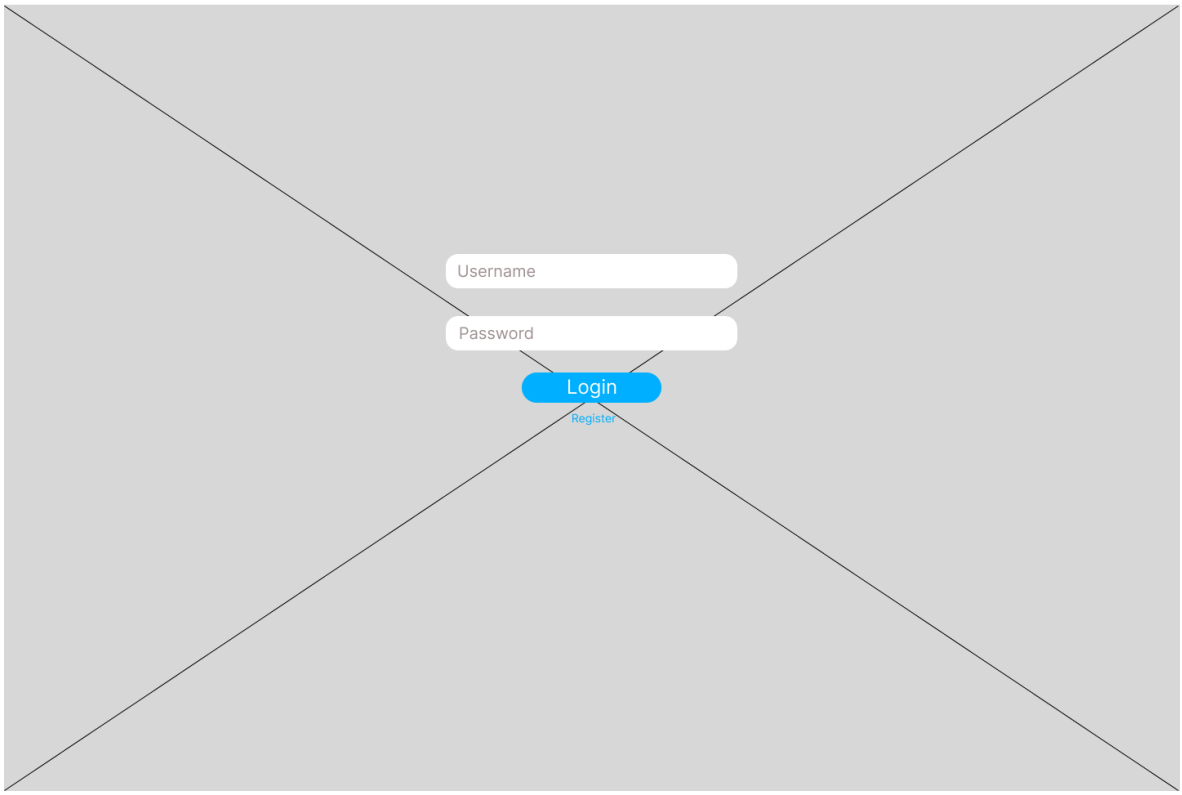


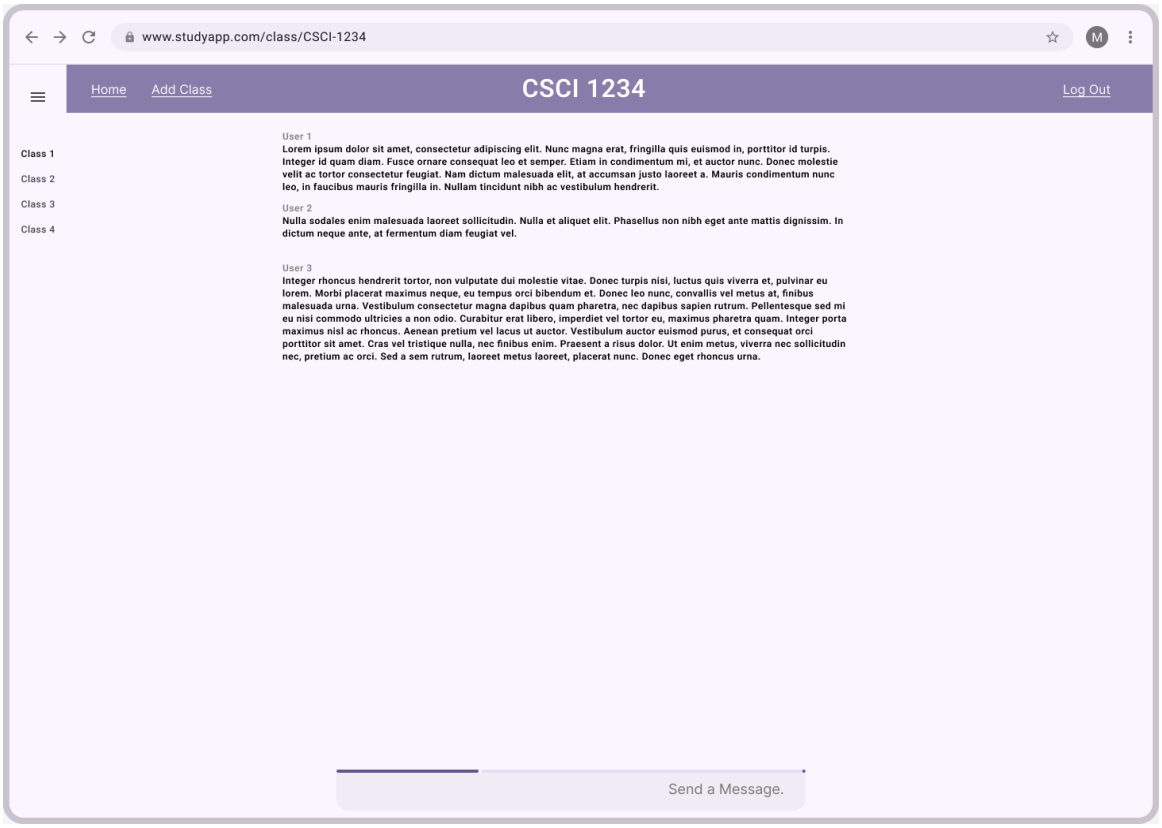
Wireframes

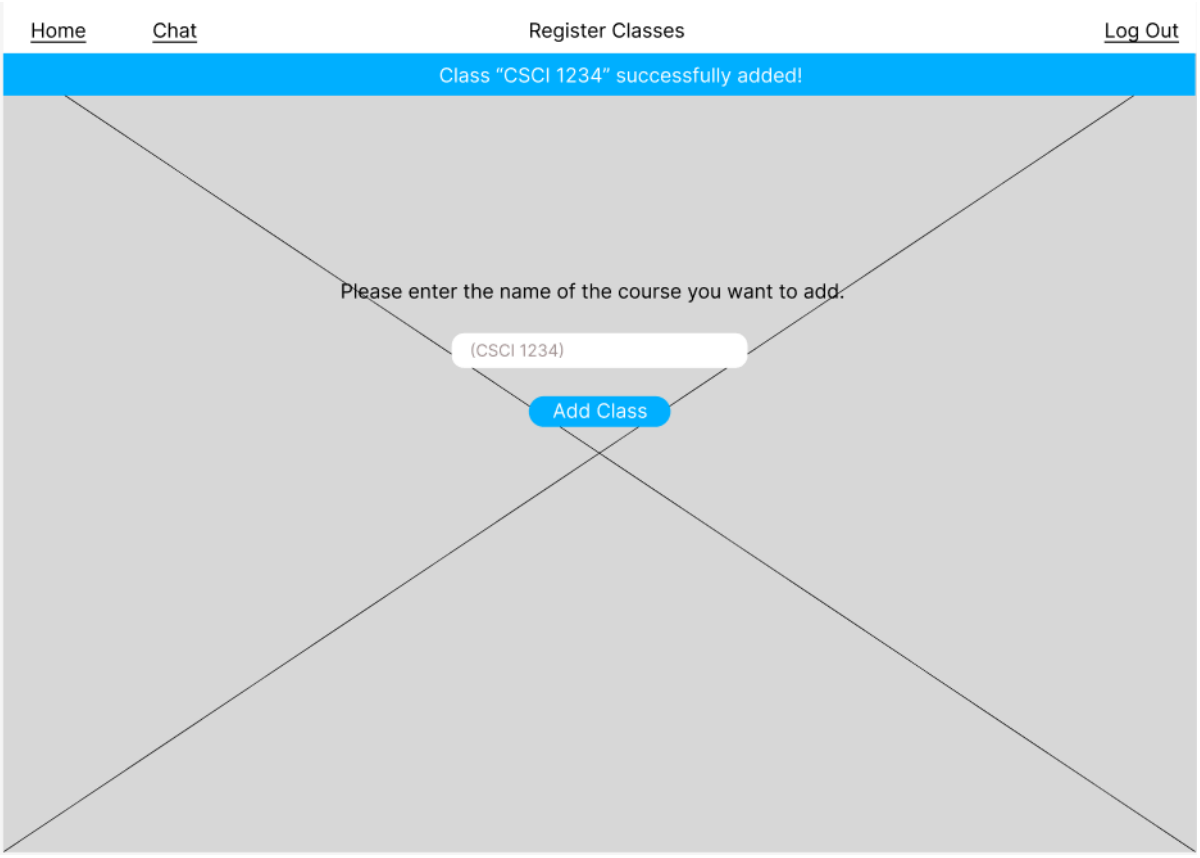
Home Login Welcome!



Home Register Welcome!




www.studyapp.com/class/CSCI-1234
Home
Add Class
CSCI 1234
Log Out
Class 1
Class 2
Class 3
Class 4
User 1
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc magna erat, fringilla quis euismod in, porttitor id turpis. Integer id quam diam. Fusce ornare consequat leo et semper. Etiam in condimentum mi, et auctor nunc. Donec molestie velit ac tortor consectetur feugiat. Nam dictum malesuada elit, at accumsan justo laoreet a. Mauris condimentum nunc leo, in faucibus mauris fringilla in. Nullam tincidunt nibh ac vestibulum hendrerit.
User 2
Nulla sodales enim malesuada laoreet sollicitudin. Nulla et aliquet elit. Phasellus non nibh eget ante mattis dignissim. In dictum neque ante, at fermentum diam feugiat vel.
User 3
Integer rhoncus hendrerit tortor, non vulputate dui molestie vitae. Donec turpis nisi, luctus quis viverra et, pulvinar eu lorem. Morbi placerat maximus neque, eu tempus orci bibendum et. Donec leo nunc, convallis vel metus at, finibus malesuada urna. Vestibulum consectetur magna dapibus quam pharetra, nec dapibus sapien rutrum. Pellentesque sed mi eu nisi commodo ultricies a non odio. Curabitur erat libero, imperdiet vel tortor eu, maximus pharetra quam. Integer porta maximus nisl ac rhoncus. Aenean pretium vel lacus ut auctor. Vestibulum auctor euismod purus, et consequat orci porttitor sit amet. Cras vel tristique nulla, nec finibus enim. Praesent a risus dolor. Ut enim metus, viverra nec sollicitudin nec, pretium ac orci. Sed a sem rutrum, laoreet metus laoreet, placerat nunc. Donec eget rhoncus urna.
Send a Message.


Home
Chat
Register Classes
Log Out
Class "CSCI 1234" successfully added!
Please enter the name of the course you want to add.
(CSCI 1234)
Add Class

Test Results

For our test cases, we tested the following routes: /register, /add, /login with positive and negative test cases for all of them. We observed users trying different things such as registering with invalid credentials (i.e, no @ in email, duplicate accounts) and added checks to make sure people couldn't register with faulty data. Similarly, for the login route we had users log in with their email instead of a username to ensure uniqueness. Additionally, we added middleware to ensure no user was accessing certain pages, like the add class page, without being logged in. Furthermore, users were trying to add duplicate classes, so we created a check that makes sure users don't register for the same class twice.

Deployment - https://study-room-4n5l.onrender.com/