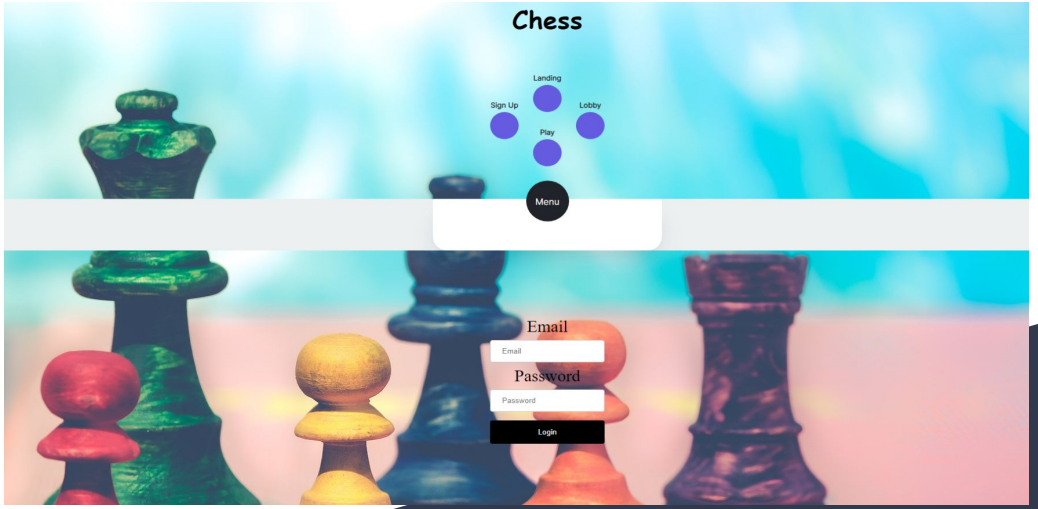
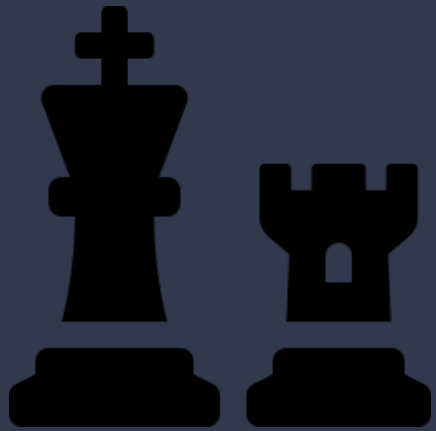
By: Babak Milani, Daniel Moorhatch,
Christian Montalvo and Karl Moreno

Our game

Our Chess game follows the standard rules and layout of traditional chess on an online platform. Users will have the options to create an account, create/join a game and sign in.

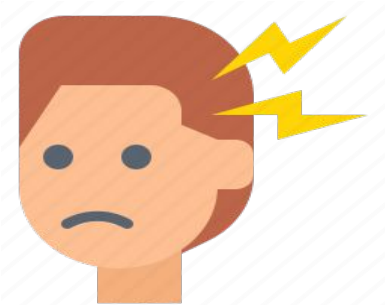




Difficulties of the project.

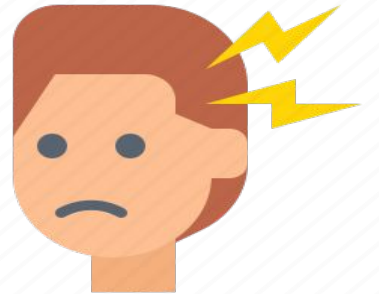
Working with new technology is always a difficulty to overcome at first. Understanding how things work and watching videos helps us understand these new tools that we worked with on this project such as Postgres and Render which luckily we did not have to implement.

Implementing the actual game such as having the game board and pieces appear was also a difficulty we ran into. I believe we spent a lot of time focusing on making sure our user sign up and sessions were working properly that we ran out of time to refine the game itself and have it fully implemented.



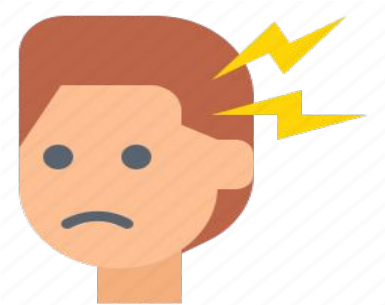
Difficulties of the project.

As I followed along the lecture recordings I would get in a flow and hit a wall where I had to retract and find my way back to point in time where I could continue from the best working code which actually turned out to be inefficient and very time consuming in itself. so , I decided to backup every big milestone and name it appropriately so that I could find my way back and start over. The problem i had this semester was managing three group projects. So when I had to switch gears. I would forget all the work I had done on the other projects. This caused some confusion at times. Also, as I would follow the video lectures to the tee, I would still get errors where I had to debug.



Difficulties of the project.

When I reached the Socket.io milestone, I really started to get frustrated. I had to spend a half a day just to figure it out. So eventually gave up and just tried to get the most functionality and aim at getting a chess board and the ability to move pieces. I made two different model of the web app where one actually had a chessboard on it but no web socket connection. Then I started a new model and I was able to get a web socket connection which is the final code I submitted.



Main problems

We believe a main overlying problem with this project and projects in general is finding a common ground with teammates and time commitments to work together and complete the task. With other courses work, outside life commitments like jobs, and busy schedules it can be very hard to make steady progress. We were able to combat this by using tracking apps such as Trello and weekly meetings to stay on track and updated with each of our members.

What we learned from this project

This project has taught us a lot of valuable skills not only in development but also working in a group setting. Taking into account fellow members time constraints and using tools such as Trello to manage and delegate work is a key part of working on projects as a team as stated in the slide before.

We also learned how to properly hold a session in our application using cookies and tables to store our session information. This was the first time for our group members to use this kind of technology and it was very helpful and informative as sessions are a key aspect of software development not only in games such as ours, but websites of all types.

Backend/Technical Information

These screenshots show our backend of the project that isn't highlighted in our demo video.

Cookies table

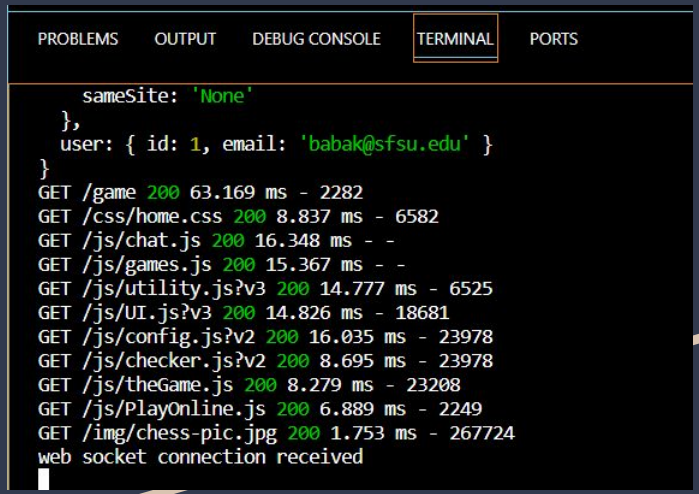
```
chess=# SELECT * FROM session;
          sid                 |                                                  sess                                                                                  |       expire
------------------------------+----------------------------------------------------------------------------------------------------------------------------------------+---------------------
 cuDRxtGgvT71jJ91i3yPRduuwKmYShJZ | {"cookie":{"originalMaxAge":null,"expires":null,"secure":false,"httpOnly":true,"path":"/","sameSite":"None"},"user":{"id":2,"email":"bmilani@sfsu.edu"}} | 2023-12-15 06:43:48
(1 row)
```

Sessions table

```
PS C:\Users\ctrc\downloads\SFSU Fall 2023\team-term-project-nocoding> psql -h localhost -U postgres -d chess
Password for user postgres:
psql (16.1)
WARNING: Console code page (437) differs from Windows code page (1252)
         8-bit characters might not work correctly. See psql reference
         page "Notes for Windows users" for details.
Type "help" for help.

chess=# \dt
            List of relations
 Schema |     Name     | Type  |  Owner
--------+--------------+-------+----------
 public | pgmigrations | table | postgres
 public | test_table   | table | postgres
 public | users        | table | postgres
(3 rows)


chess=# \d users
                                         Table "public.users"
   Column   |            Type             | Collation | Nullable |              Default
------------+-----------------------------+-----------+----------+-----------------------------------
 id         | integer                     |           | not null | nextval('users_id_seq'::regclass)
 email      | character varying(256)      |           | not null |
 password   | character(60)               |           | not null |
 created_at | timestamp without time zone |           | not null | CURRENT_TIMESTAMP
Indexes:
    "users_pkey" PRIMARY KEY, btree (id)
    "users_email_key" UNIQUE CONSTRAINT, btree (email)


chess=# SELECT * FROM users;
 id |       email       |                           password                           |         created_at
----+-------------------+--------------------------------------------------------------+----------------------------
  1 | bmilani@mail.sfsu | $2b$10$DlpZBMkK0yYG4T52M5FcZeWGEVkH38L06GqQGgvDW9.A1q8DNq1ny | 2023-12-10 22:14:49.993034
(1 row)


chess=# \dt
            List of relations
 Schema |     Name     | Type  |  Owner
--------+--------------+-------+----------
 public | pgmigrations | table | postgres
 public | test_table   | table | postgres
 public | users        | table | postgres
(3 rows)


chess=# \dt
            List of relations
 Schema |     Name     | Type  |  Owner
--------+--------------+-------+----------
 public | pgmigrations | table | postgres
 public | session      | table | postgres
 public | test_table   | table | postgres
 public | users        | table | postgres
(4 rows)


chess=#
```

Backend/Technical Information

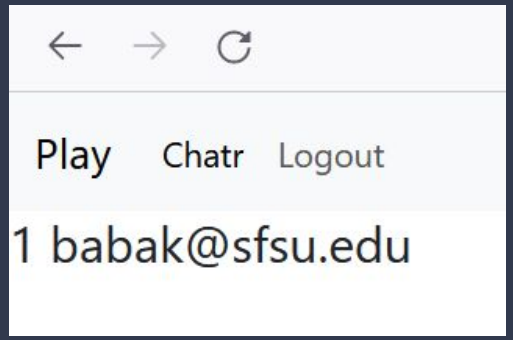
Console log

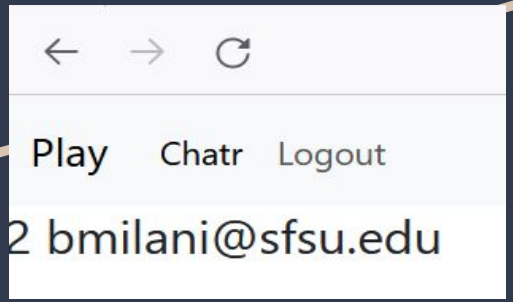


I was able to make web socket connection but unable to render my chessboard.

Backend/Technical Information

Stored session cookie and user info

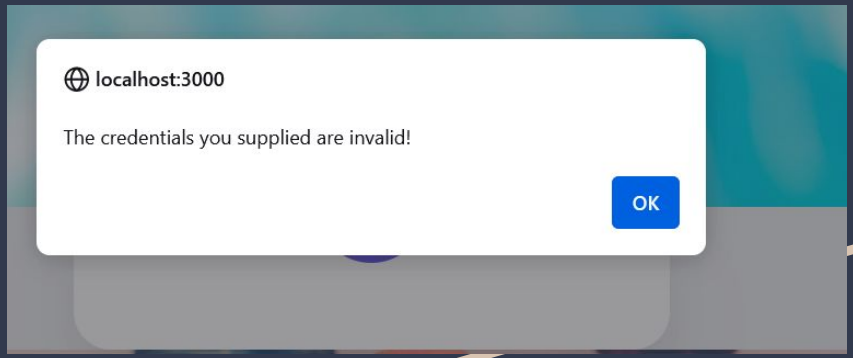




I was able to properly implement the session code to display user id and email per the lecture recordings. I admit that I had to watch every recording twice to backtrack. Sometimes I had to spend 6 hours on the 2 hour video.

Backend/Technical Information

Error popup message when user enters incorrect credentials to login



I was also able to implement a small code in the form section when the user is prompted to enter his/her credentials.

Backend/Technical Information

Things that didnt get implemented

I was unable to get the chat feature and the chessboard and pieces implemented. Also, the sign up form page link was broken last minute. But it worked as i was able to save a session and two users.

We were unable to implement the game itself due to running out of time and spending too much effort on the log in sessions and the sign up account features.

Demo of our application

Video of our game demonstration

https://drive.google.com/file/d/1VCwqtP6Db13kg0upRsUFjfvK2UBpP2N0/view?usp=sharing