Browser tab: Zchess

URL: https://zchess.com/r52f2ck

TheKing21

All Static Logic is received with game_start call.

JS handles chess board input logic, makes moves by posting to the server.

Chessboard responds to chess event stream, moves are posted there as they come in.

All Dynamic Logic is received with game events from the game event stream.

Game Code: r52f2ck7

Challenges | Search | Friends

Chat

TheQueen19: nice move!
TheKing21: Thank you!
TheQueen19: Good game!
TheKing21: ggs

send

Posts new message data to the backend. Chat stream is updated.

TheQueen19 **TheKing21**

White | Black

05:00 | 05:00

Moves

e2e4 -> e7e5
g1f3 -> _____

Resign | Offer Draw

Adds Resign or Draw property to move POST request

Opponent

search

TheQueen19 ●
randomedude21 ●

Friends | None | Challenge

Friends

search

TheQueen19 ●
randomedude21 ●

Remove Friend | Challenge

Search

search

TheQueen19 ●
randomedude21 ●

Friends | Add Friend

Challenges

search

TheQueen19   05:00
randomedude21   05:00

Deny | Accept

Play

Create Game:

Opponent: None

Time: 05:00

Color: White Random Black

Join Game:

Enter Code: _____

Search bars get from the server on change. Friends is filled by a GET call to the backend.

If the data exists in memory it is search there.

Join Game hits POSTS to the join endpoint which takes a

1.) User decides to create or join a game.

2.) If The user presses the Opponent Field they are taken to the Opponent Popup

3.) The user selects their opponent.

3.5) The user presses friends and selects their friend as opponent.

4.) User sets challenge for opponent.

5.) User finishes form and pressed create or join.

This takes all the data posts it to the backend to start the game.

6.) User waits for opponent and plays game

```
                    ┌──────────────────┐
                   /   Client Browser    \
                  /                        \
                 └──────────────────────────┘

                  ┌────────────────────┐
                 <      INTERNET         >
                  └────────────────────┘

         ┌──────────────────┐
        /  HTTP Node.JS server \
       /                        \
      └──────────────────────────┘
```

Client asks server for HTML, CSS, and JavaScript content.

Client interacts with site for Account CRUD.

Client Creates a game with opponent username

Opens a websocket for the event of the game to be communicated.

Custom Chessboard canvas communicates over the socket.

Node server combines express endpoints and Socket.io to create a single server game experience for any user.

Express REST Endpoints

Socket.IO Websockets

CRUD Endpoints

Content Delivery Endpoint

Game Event Handlers

MySQL Zchess Database