CSC 667-01

Team 02

Team Members:

Team Lead: **Jonathan Gurdal**
Front End Lead: Douglas Hebel
Back End Lead: Andrew Sarmiento
Github Master: Ahshil Shah
Back End: James Day
Front End: Tuan Le

https://github.com/csc667/csc667-su19-Team02

August 16th, 2019

Project Introduction

Our team developed a Chess application that allows 2 users to play an online game of chess between one another. They are able to communicate with one another in global and local chat rooms.

Software Stack

Server: AWS

Web Server: AWS

Database Server: MySQL

Server side Language: JavaScript, Node.js

Frameworks: Bootstrap

Additional Packages:

- **Chess.js:** Used for handling the game of chess.
- **Chessboard.js:** Used for drawing the chessboard and move pieces.
- **Bcrypt:** Security for hashing sensitive user information.
- **Connect-flash:** Display messages to users
- **EJS:** Pass variables from front-end to back end and vis versa, help recycle code, simplified pathing.
- **Express:** Everything routing
- **Nodemon:** Simplified updating and running code.
- **Passport:** Used for authentication purposes.
- **Socket:** Real time chat and moves.

Tools Used

Communication: Our group primarily communicated with Discord, where we held group discussions and utilized tagging team members to messages relevant to them. We also used it to help schedule time to meet outside of class to work on the application.

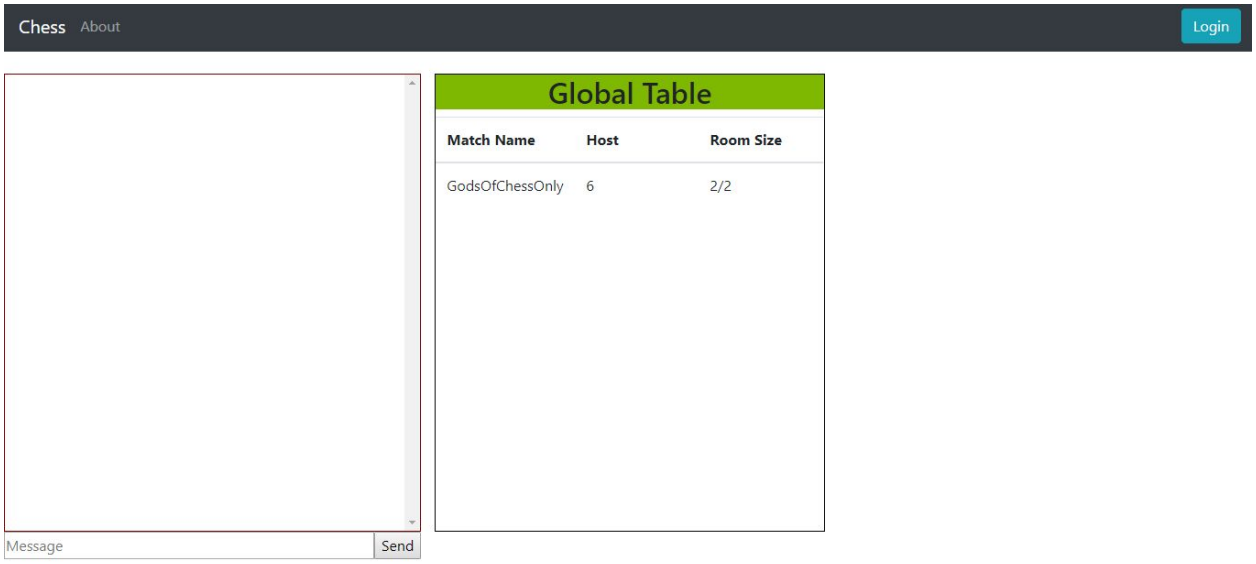
Task Management: Initially, our form of Task Management was through Trello. Eventually, as tasks became more complex and individual work paces differed, weekly meetings and communication through Discord became our main form of Task Management. Our weekly meetings consisted of drawing out plans for the next meeting on a whiteboard.

Build Instructions

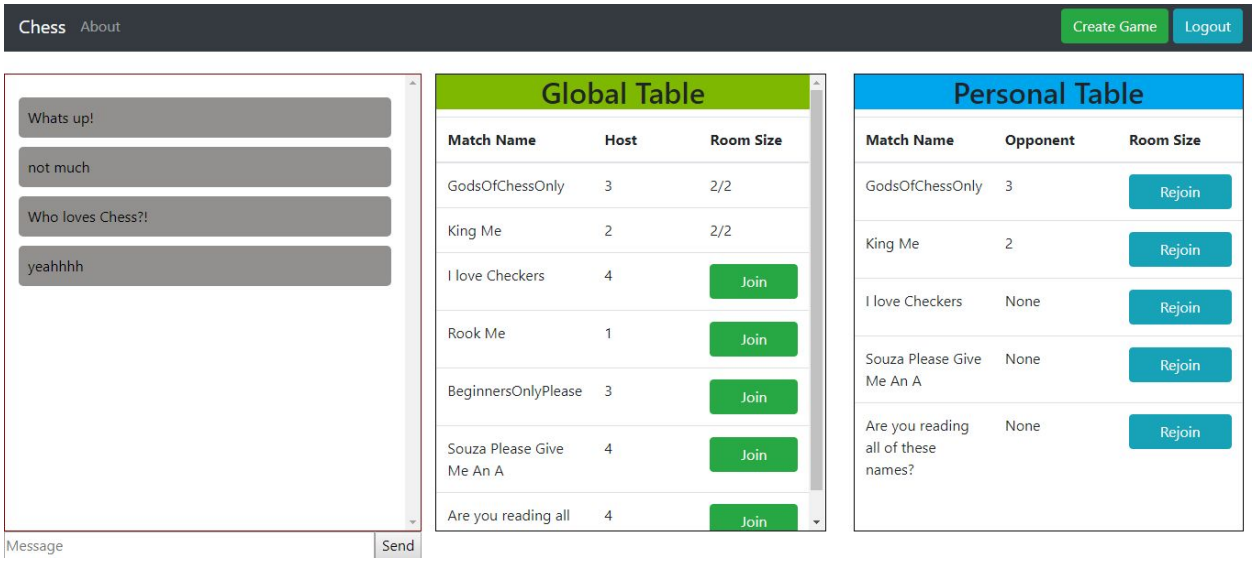
1. cd application
2. npm install
3. node app.js

Pictures

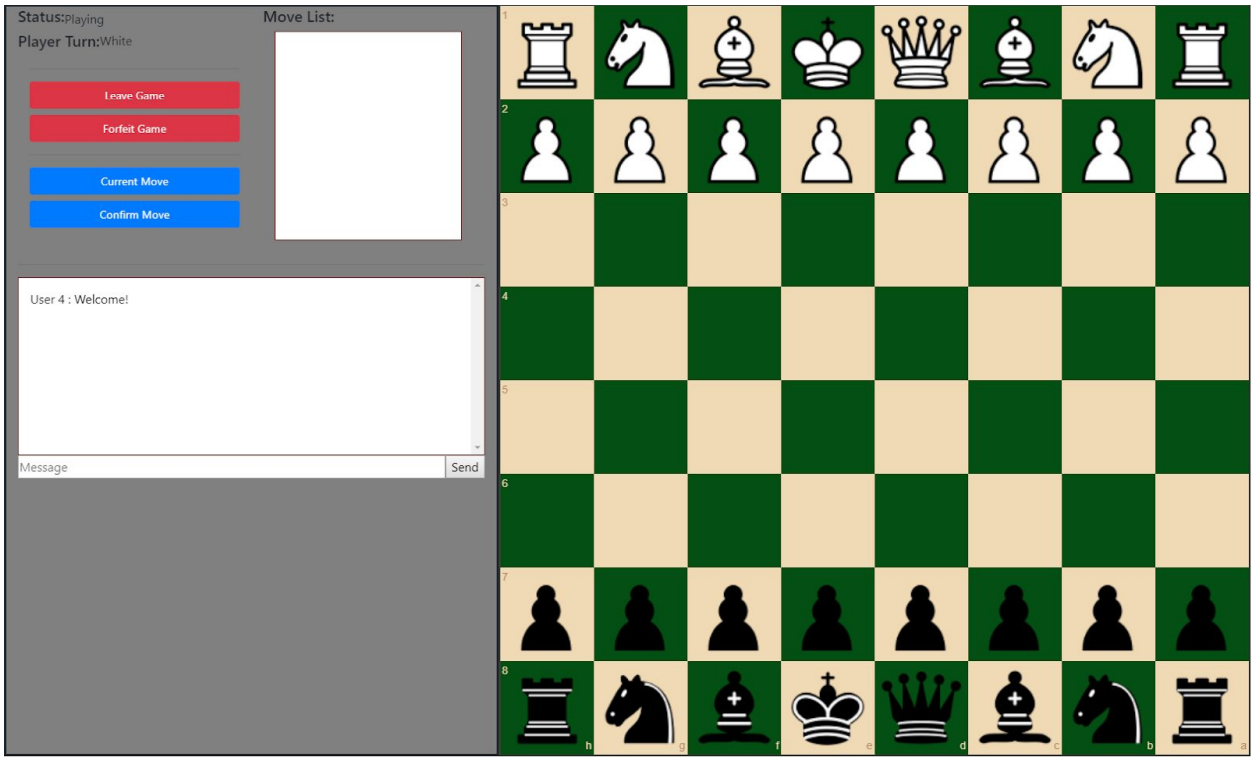
Unregistered Home Screen



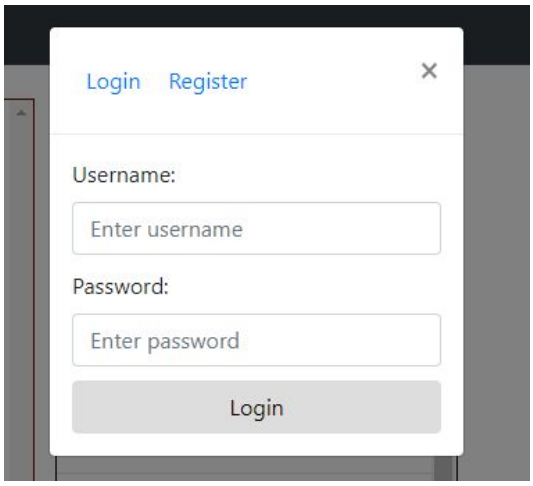
Registered Home Screen



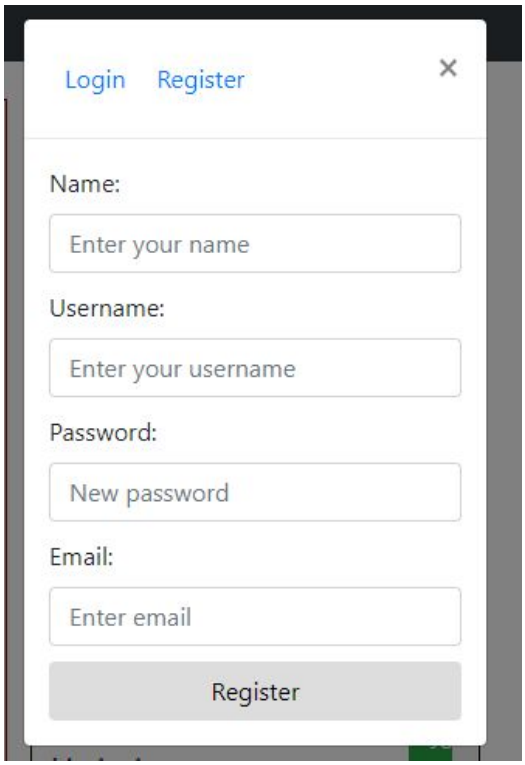
Chess Page



Login



Registration



List of API routes

Route for registering

post /user/:username,:password

req.params: {"username" : "string", "password" : "string"};

Route for logging in

post /user/:email,:username,:password

req.params: {"email" : "string", "username" : "string", "password" : "string"};

Route for forgot password

post /user/:email

req.params: {"email" : "email"};

Route for resetting password

post /user/:password

req.params: {"password" : "string"};

Route for creating new game

post /gameLobby/:game_name

req.params: {"game_name" : "string"};

Route for creating joining a game

post /gameLobby/:game_name

req.params: {"game_name" : "string"};

Route for sending message global

post /home/:message

req.params: {"username" : "string", "message" : "string"};

Route for sending message in game

post /game_name/:message

req.params: {"username" : "string", "message" : "string"};

Route for making a move

post /game_name/:username,:mov

req.params: {"username" : "string","move" : "val"};

Route for forfeiting a game

post /game_name/:username

req.params: {"username" : "string"};

Team Member Contribution

Andrew Sarmiento

- Server Setup (AWS)
- Database Initialization (AWS RDS)
- Backend (user auth, create/join game, socket io, error handling)

Jonathan Gurdal

- Milestone Documentation
- Bootstrap front end framework set up
- CSS styling for pages
- Form Validation
- Frontend

Douglas Hebel

- Wire Frame / Story Board
- Initial html / css implementation
- Front end implementation support
- Misc front end bugs
- Presentation Assistance

Ahshil Shah

- Join Game/Rejoin Game
- Game logic
- Backend

James Day

- Presenter
- Route Maker
- Misc Bug Fixing/Problem Solving

Tuan Le

- Frontend
- Html , js and css styling support

Project Reflection

In the beginning of the summer semester, our group was slow to start in terms of communication and working towards the project. As a group, we had a hard time creating goals to work towards. Before the check points were assigned we underestimated the size of the project and the amount of work needed to be done. Once the check points were assigned, our group became a lot more efficient at getting tasks done. From there, our group started working in pairs to implement and deliver required features. We developed weekly meetings to discuss where we were currently in the assignment and what needed to be done. During that time, everyone was willing to teach and assist one another with questions and issues that came up. If we were to do this project again, we would create a more structured project in order to help streamline and deliver the final project.

Project Conclusion

We built a complete chess application that allows two players to play online and and communicate via local and global chat. Additionally, we also implemented a move list on the game screen. Within the lobby, the implementation of a list of games you are currently in is displayed as well as a list of global games. We would have wished to have added additional features such as 3D visuals, match ratings, iterating through previous moves. If we had more time, we could have increased the scope of our project to include these additional features. Our team worked very well together to deliver a final product, even though there were things we could have improved on, we are happy with our final project.