DTP Final Presentation

Accomplishments, struggles, and what still needs to be done

By: Michael Liguori and Christopher Chen

Back End Components

Web Server and Database Server

Web Server

Dependencies: Node js, Express js, mysql js, socket io, pug js

Node js- for server functionality. Handles connection requests from client.

Express js- Streamlines routing and web server functionality

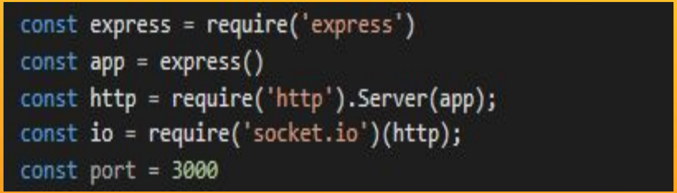
Mysql js- Streamlines connection to a MySQL server, which was manipulated through MySQL workbench interface and through Web server calls to the MySQL Server

Socket io- streamlines open connections and rooms for each connection to provide chat abilities and handles asynchronous callback emit and listen events between client and web server.

Pug js- allows to dynamically create html files with data retrieved from DB server to send to client upon request.

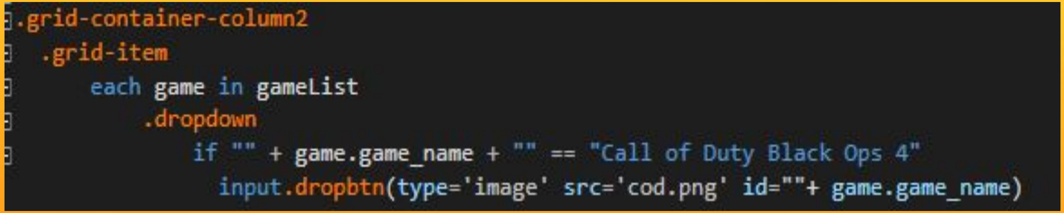
Web Server

Dependencies: Node js, Express js, mysql js, socket io, pug js



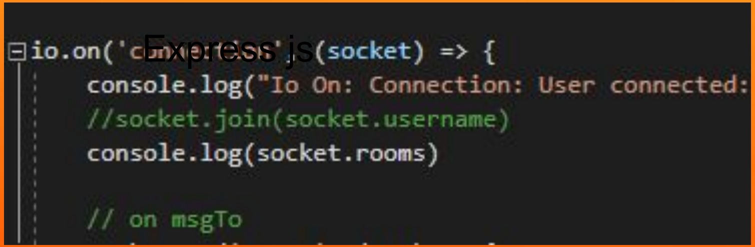
```
const express = require('express')
const app = express()
const http = require('http').Server(app);
const io = require('socket.io')(http);
const port = 3000
```

Express js



```
.grid-container-column2
  .grid-item
      each game in gameList
          .dropdown
              if "" + game.game_name + "" == "Call of Duty Black Ops 4"
                input.dropbtn(type='image' src='cod.png' id=""+ game.game_name)
```

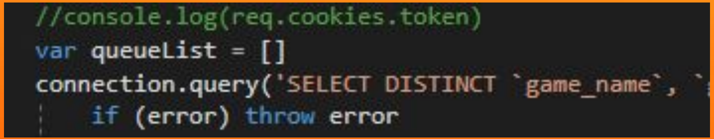
pug js



```
io.on('connection', (socket) => {
    console.log("Io On: Connection: User connected:
    //socket.join(socket.username)
    console.log(socket.rooms)

    // on msgTo
```

Express js

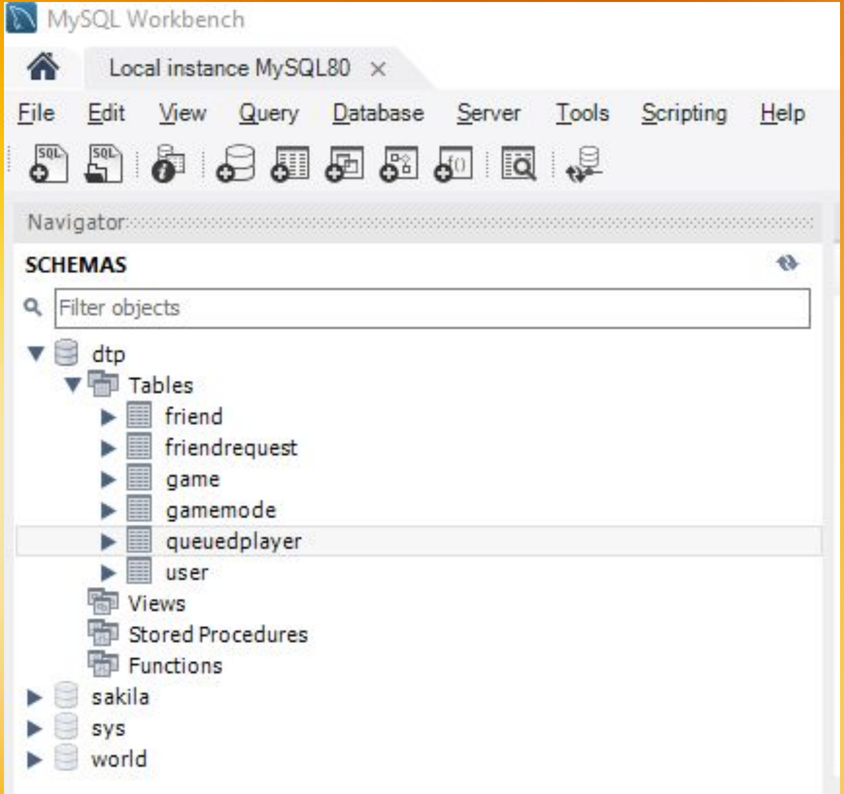
Socket io

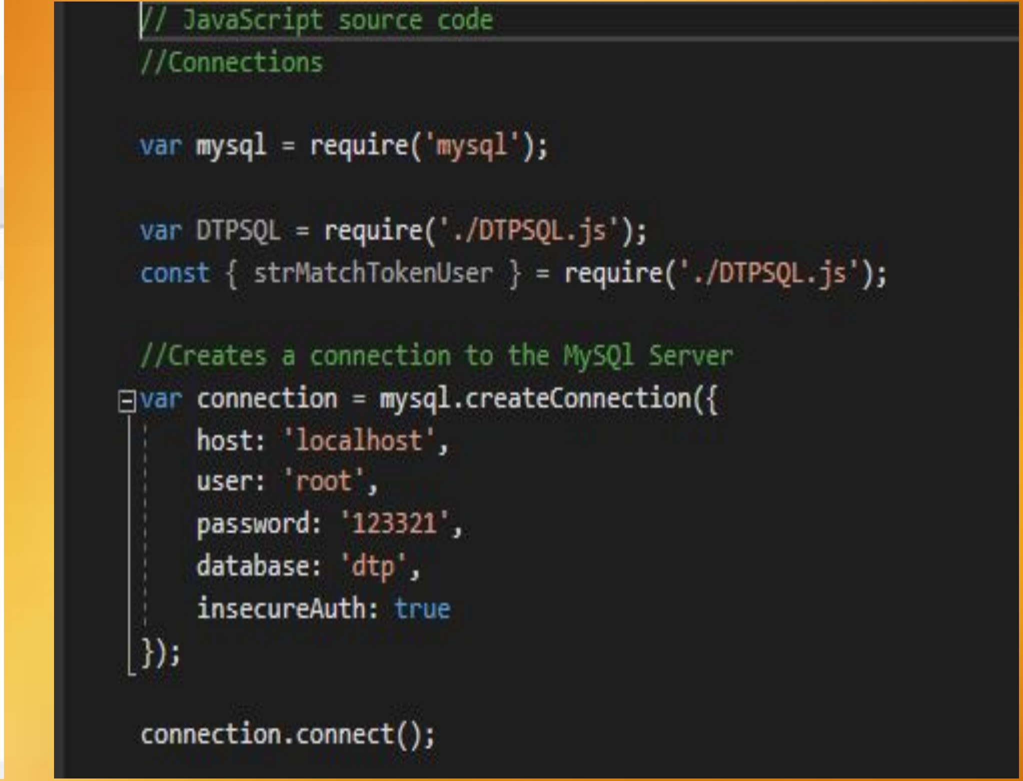


```
//console.log(req.cookies.token)
var queueList = []
connection.query('SELECT DISTINCT `game_name`, `
    if (error) throw error
```

mysql js

DB Server - MySQL WorkBench





```
// JavaScript source code
//Connections

var mysql = require('mysql');

var DTPSQL = require('./DTPSQL.js');
const { strMatchTokenUser } = require('./DTPSQL.js');

//Creates a connection to the MySQl Server
var connection = mysql.createConnection({
    host: 'localhost',
    user: 'root',
    password: '123321',
    database: 'dtp',
    insecureAuth: true
});

connection.connect();
```

Entity Overview

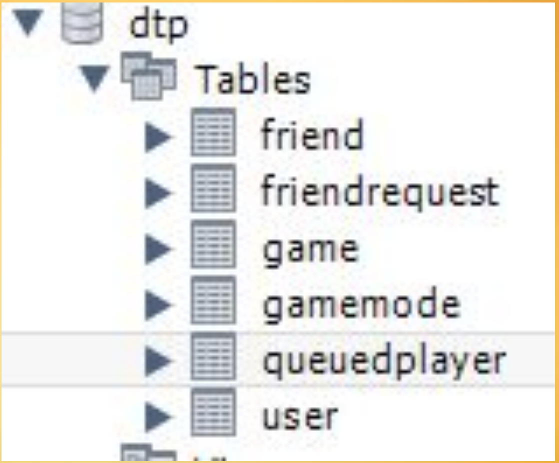
GAME

GAMEMODE

FRIENDS

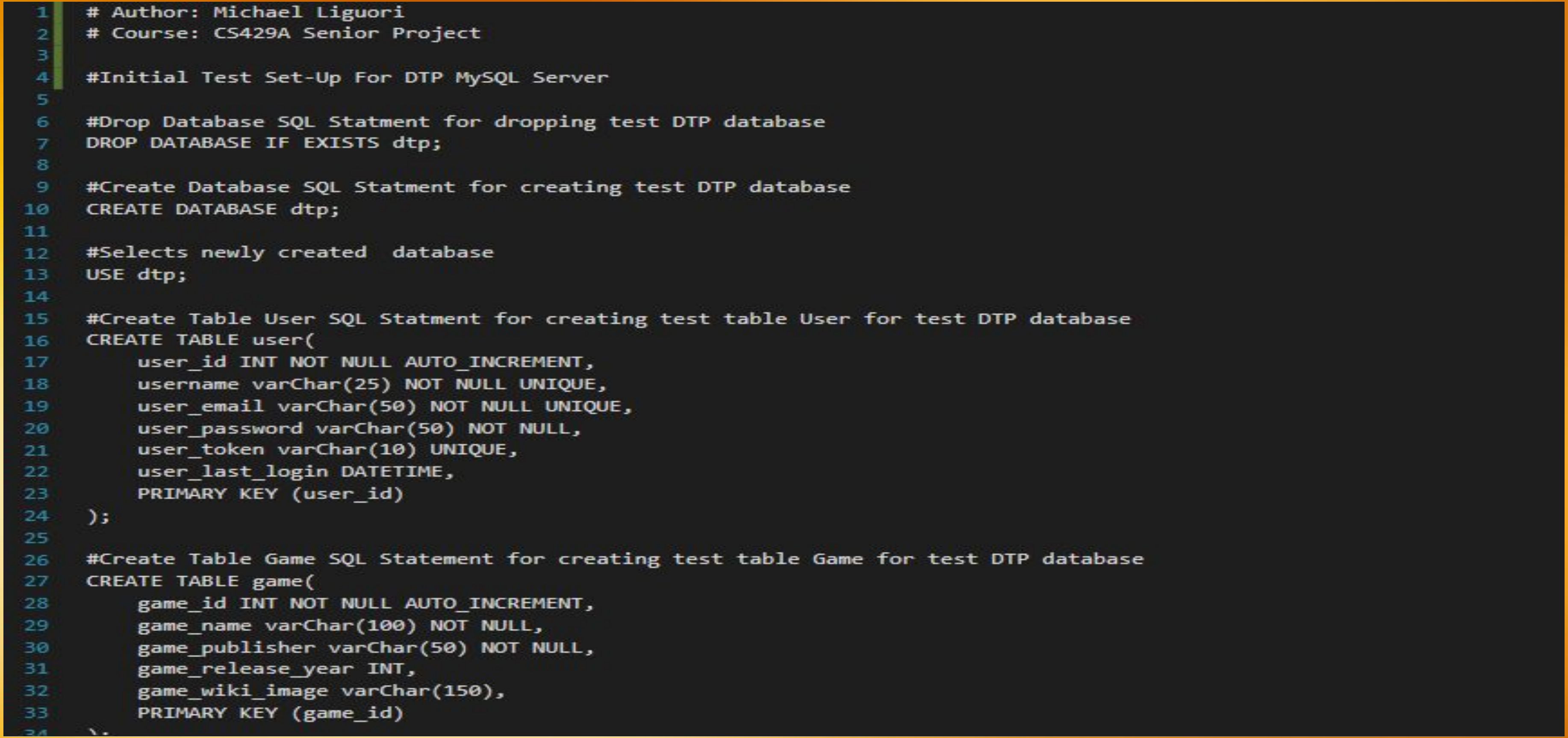
USER

QUEUEPLAYER



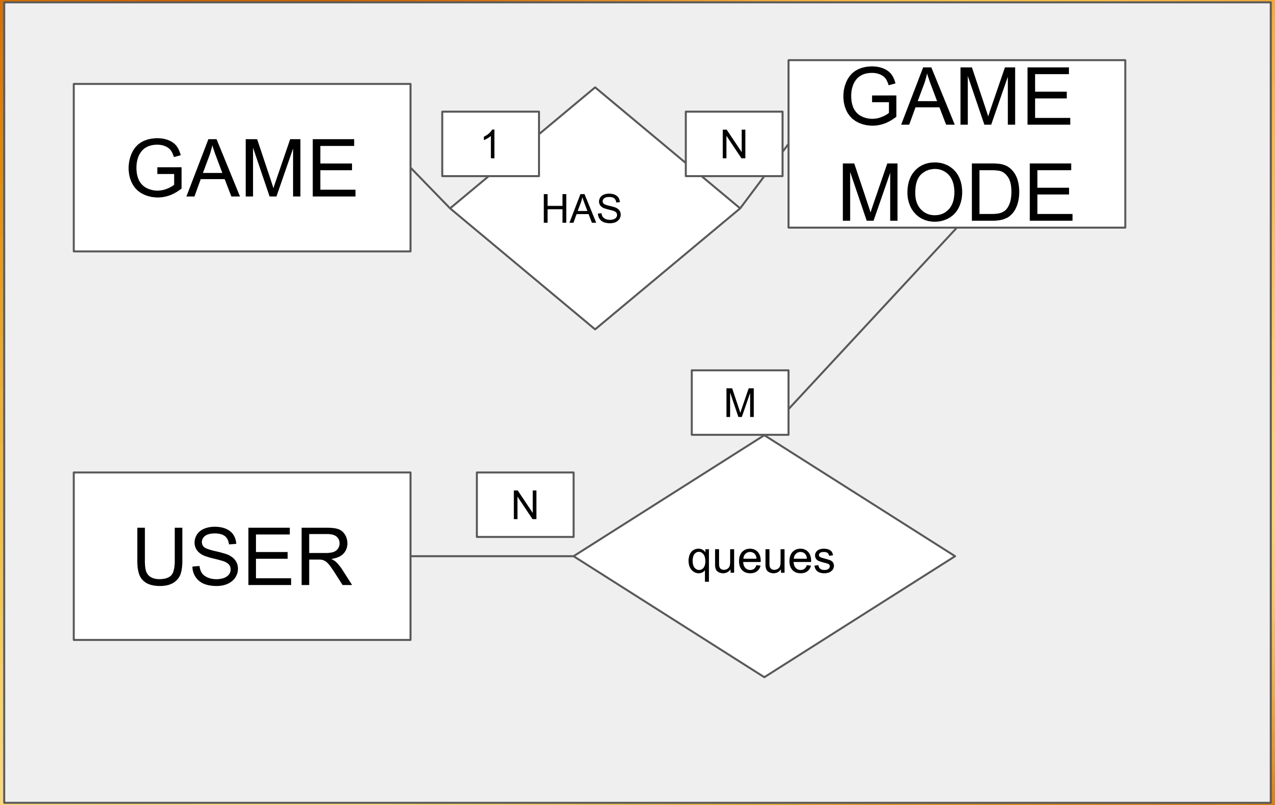
```
▼ dtp
    ▼ Tables
        ▶ friend
        ▶ friendrequest
        ▶ game
        ▶ gamemode
        ▶ queuedplayer
        ▶ user
```

SQL Statements



```sql
# Author: Michael Liguori
# Course: CS429A Senior Project

#Initial Test Set-Up For DTP MySQL Server

#Drop Database SQL Statment for dropping test DTP database
DROP DATABASE IF EXISTS dtp;

#Create Database SQL Statment for creating test DTP database
CREATE DATABASE dtp;

#Selects newly created  database
USE dtp;

#Create Table User SQL Statment for creating test table User for test DTP database
CREATE TABLE user(
    user_id INT NOT NULL AUTO_INCREMENT,
    username varChar(25) NOT NULL UNIQUE,
    user_email varChar(50) NOT NULL UNIQUE,
    user_password varChar(50) NOT NULL,
    user_token varChar(10) UNIQUE,
    user_last_login DATETIME,
    PRIMARY KEY (user_id)
);

#Create Table Game SQL Statement for creating test table Game for test DTP database
CREATE TABLE game(
    game_id INT NOT NULL AUTO_INCREMENT,
    game_name varChar(100) NOT NULL,
    game_publisher varChar(50) NOT NULL,
    game_release_year INT,
    game_wiki_image varChar(150),
    PRIMARY KEY (game_id)
);
```

ER Diagram -Relationships For Queues



Challenges and Bugs

- Authentication enforcing re-login for every connection made to server.
    - At times a user would no longer have a cookie, yet not be directed to re-login.
- Resetting all data if a player does not respond or rejects a full queue.
    - Sometimes, a player would have accepted, someone else fails to accept, yet their values remain unchanged after supposed attempt to reset them through Web server commands
- If a player is on page and is not active, at times the accept and reject buttons received do not work for their intended purposes.
    - Due to fact that after connection has been made, if the player was not in a room, they were not joined into it, which happens n load for the My queues page.

Places for/of Improvement

- Organization of code
  - The code can be reorganized to be easier to follow
- First attempt at JS and Servers.
  - Never used this language or the other dependencies mentioned
  - Use this project as an experience to code in JS and deal with server routing and structure.
- MySQL Database field renaming
  - Streamline table names and field names for easier readability and code
- Played around with promises, and callback hells
  - More confident in maneuvering around callback hell
  - Can refactor code and MySQL commands to grab vital information needed for nested queries to improve readability and efficiency.

Goals and their status

Connecting all three major components - STATUS: COMPLETE

DTP has all three major components working together and connected.

This means a connection has been made between the DB server and the web server, and from web server to clients.

A user may affect the data in the DB, such as user table information on sign up, sign in, and authentication and queuedplayer tables when queuing up by making requests to the web server which handles those requests to brain desired results.

Goals and their status

Create a queuing service for users - STATUS: COMPLETE

Using dummy data and two active logged in users, DTP project can create a unique room token when a user hosts a room, search for other players who do not have a room for that game and gamemode.

DTP attaches the unique room token to their queues, essentially adding them to the room through the use of socket io emits and listen events.

Goals and their status

Deliver Dynamic HTML through Pug - STATUS: COMPLETE

When routes and pages require it, DTP uses pug to render a html file that is adjusted based on the data received by the Web server from the DB server.

This helped streamline pages, allowed for quicker production of what feels like dynamic pages which is vital to a user’s ability to queue up quickly and effectively.

Goals and their status

Chat for Users in Same Room - STATUS: COMPLETE

Once a User has a room token attached to one of their queues, socket io will join that user to the same room token, allowing messages to be sent and received based on this identifier.

This provides for the ability for users to chat before a room is full, and after everyone has accepted the full room to exchange host information to begin setup for a match.

Goals and their status

Accept/Reject/Timeout Actions - STATUS: MOSTLY COMPLETE

There is a bug where a user's queue has_accepted boolean does not reset back to 0 or false. This will create error conditions for next round of accepts if a full room was to occur afterwards.

DTP requires to reset certain attributes if the user accepts/rejects/timeouts after a check against a full room is found to be true. Due to the many circumstances in which a user may be part of that reset, the attribute is not correctly changed on one or two of those circumstances.

Goals and their status

Checking Authentication - STATUS: MOSTLY COMPLETE

There has been only one occurrence where the socket io fails to redirect the user to re-login either after their unique token identifier cookie expires or inactivity longer than 30 mins is detected.

Authentication is performed twice, once through the initial connection to the Node Web server and once again for the socket io connection.

Nodes authentication will trigger a re-login for client requested gets and posts, and pass vital information if authentication is successful to all other routes.

Socket io authentication joins the user into their current rooms for chat and emit/listen functions to work properly. May also lead to re-login.

Front-End Development

Accomplishments

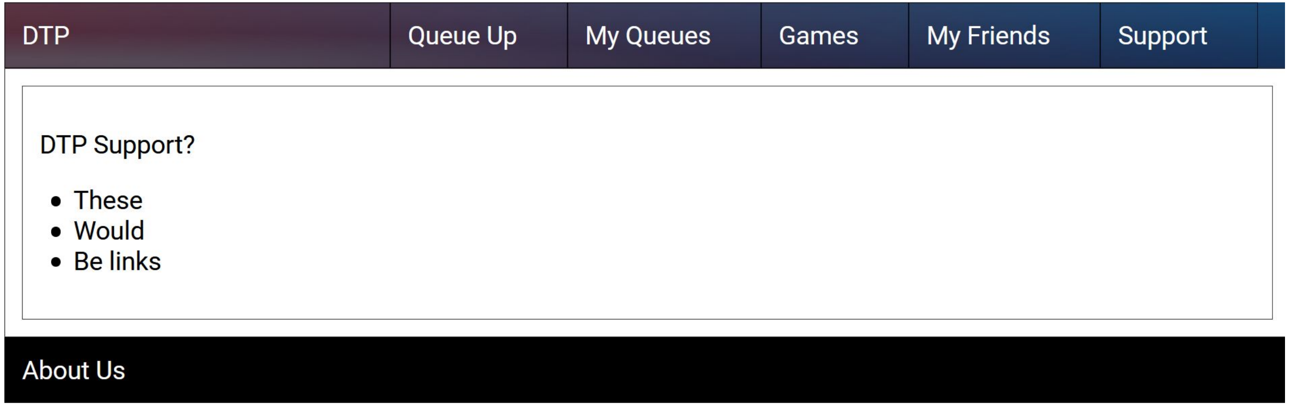
- Practiced more front-end coding
- Learned how to implement CSS and the structure of HTML
- Learned how to implement PUG and the basis of Bootstrap

Struggles

- Struggled learning new languages like Bootstrap and PUG
- Failed to implement Bootstrap
- Struggled with the connecting variables in the web page with the java script

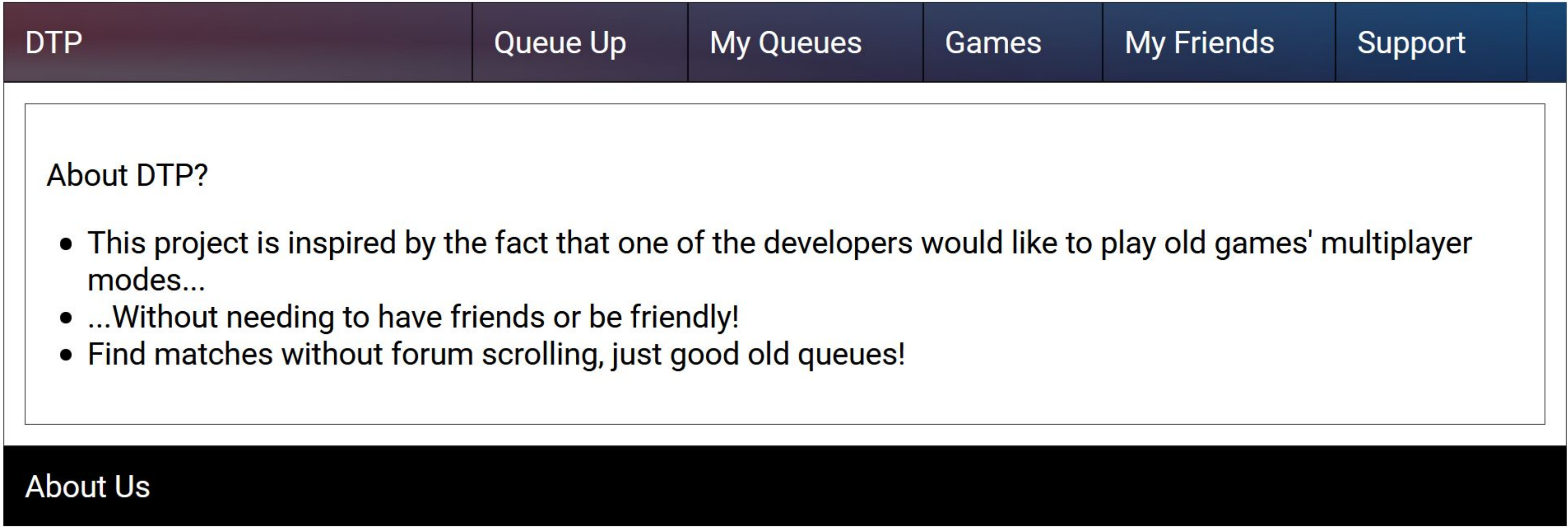
What Still Needs To Be Done

- Implementation of Bootstrap
- Streamlining My Queues for easier user interfaces
- More time to work on adding more games and corresponding gamemodes



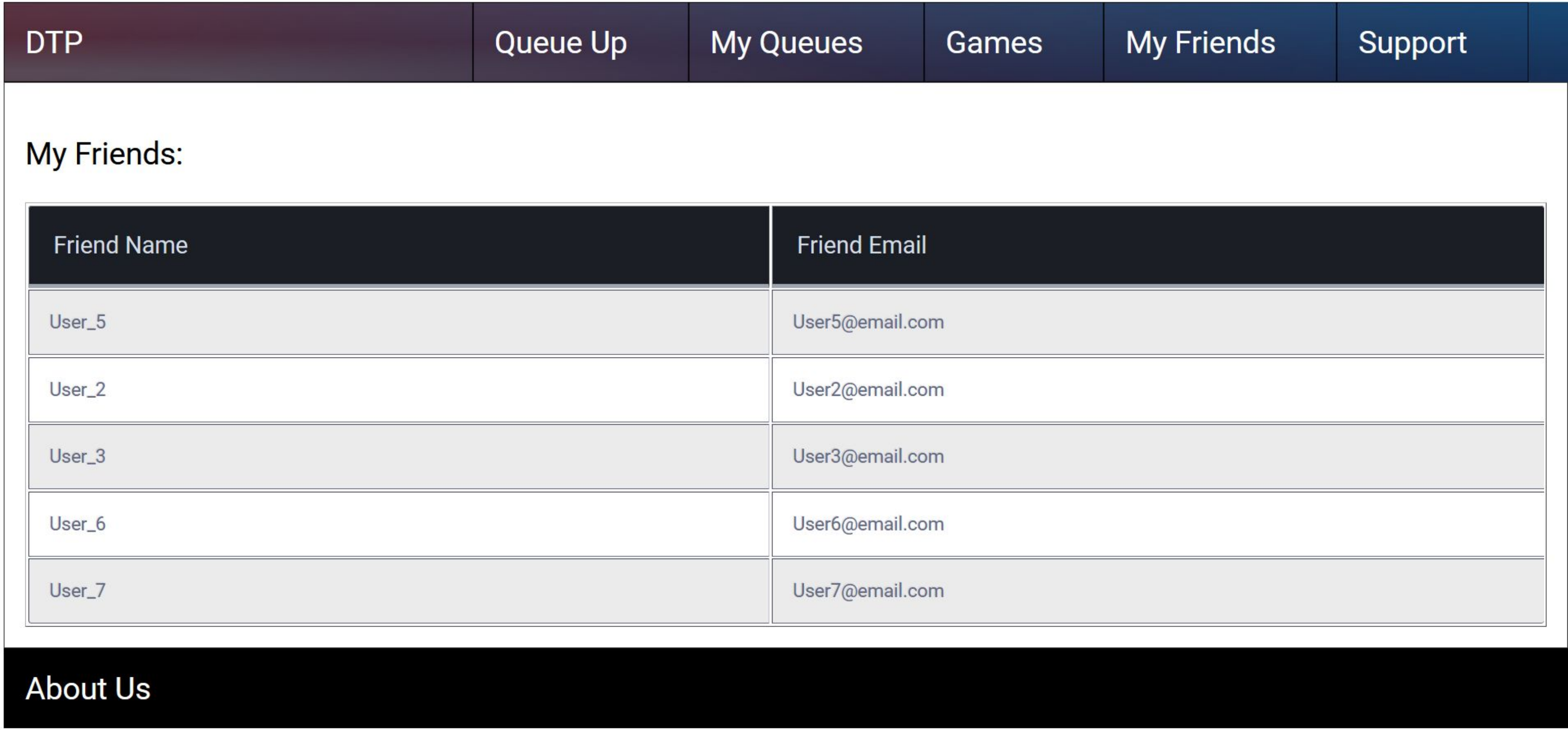
DTP Support?

- These
- Would
- Be links



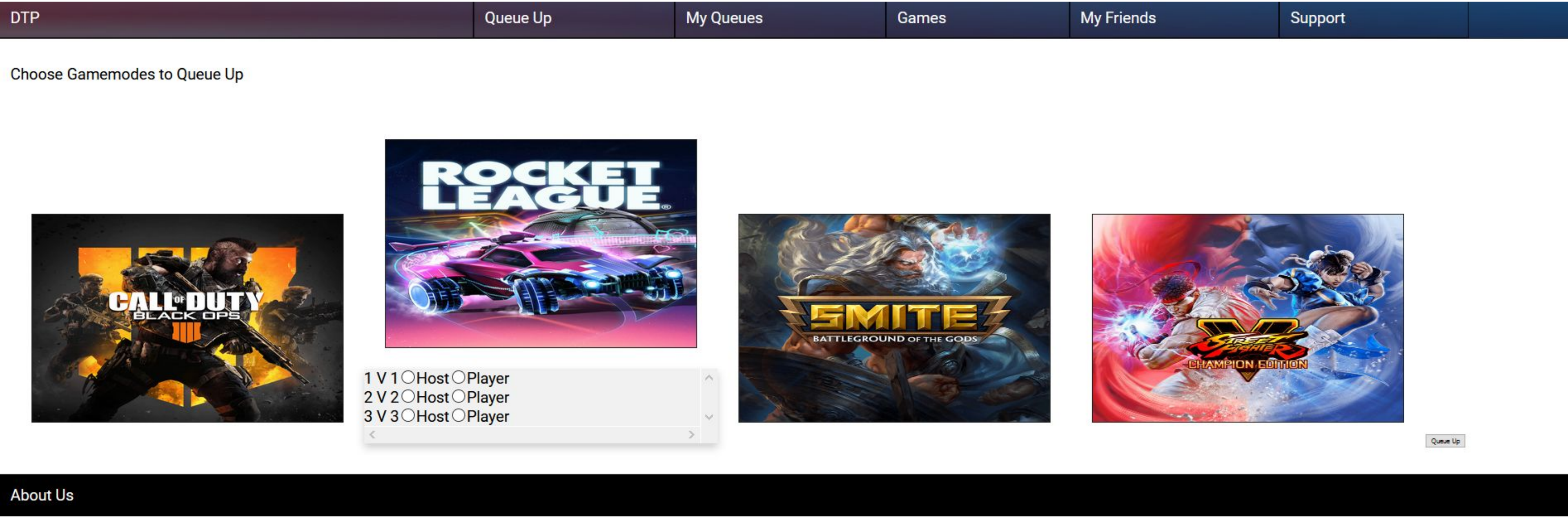
About DTP?

- This project is inspired by the fact that one of the developers would like to play old games' multiplayer modes...
- ...Without needing to have friends or be friendly!
- Find matches without forum scrolling, just good old queues!



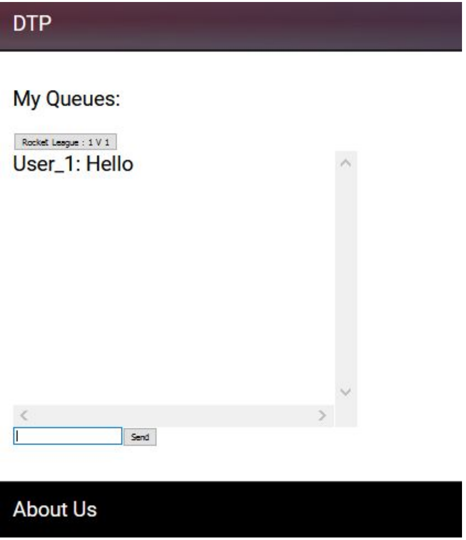
My Friends:

Friend Name	Friend Email
User_5	User5@email.com
User_2	User2@email.com
User_3	User3@email.com
User_6	User6@email.com
User_7	User7@email.com



Choose Gamemodes to Queue Up

1 V 1 ○Host ○Player
2 V 2 ○Host ○Player
3 V 3 ○Host ○Player



DTP

My Queues:

User_1: Hello

About Us