**Introduction:**

Hello and Welcome to Ping Pong Tracker

Brought to you by Team 7

Project Members: Diyor Gafiatullin, Kobe Uytiepo

Link to our code repository: <https://github.com/diyorzakirov07/serverNodejs>

Website URL: <http://ec2-18-191-187-53.us-east-2.compute.amazonaws.com:3000/games.html>

Project Description: Website that tracks ping pong games played between teams of players. Teams can have 3 players, and players can only be on one team at a time. The win/loss stats of teams are displayed on the team’s page. Player information including what team they are on can be viewed on the players page. The home page shows the 5 most recent games and the top 5 teams in the league. The games page lists all the games played in the league and the scores of each team.

**Architecture:**

Application Architecture:

* NodeJS Server and Postgres Database
* AJAX server/client communication
* New Tech: Deployment to AWS

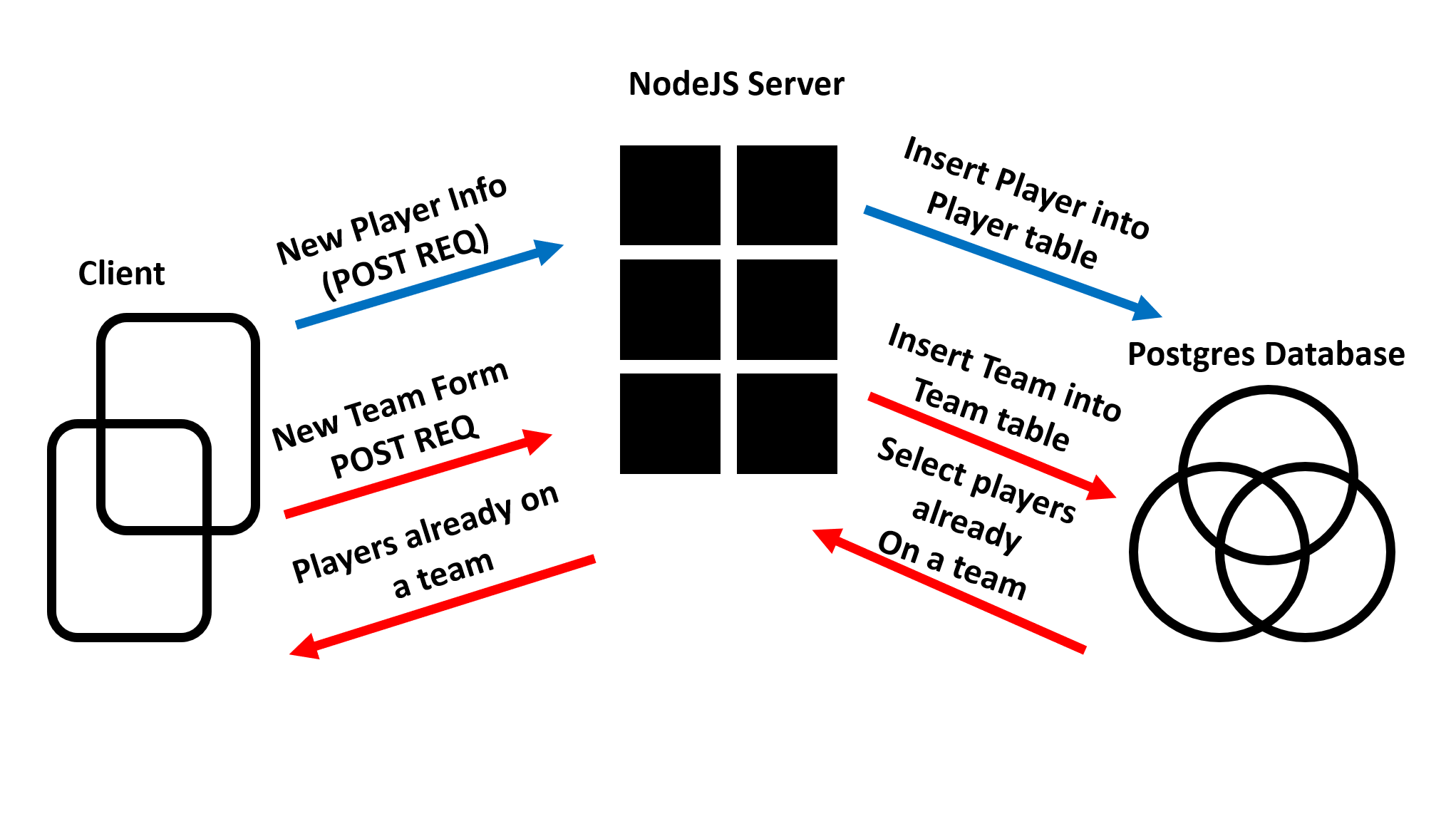
Overview:

The files in this project are hosted by AWS. A Javascript file stored on the site handles the user interaction with the HTML client. Depending on the page that the user is on, the client will send the associated fetch request to the server to query for new data. For example, the Players page will send a POST request to our NodeJS server when a new player’s information is submitted. This is sent with JSON which is parsed on the server side and inserted into our Postgres database. The database stores information about players, teams, and games in three separate tables.

**Add a Player**: In order to add a player, select the Add New Player button on the Players page. Enter the requested information and hit submit. This will send a POST request with a Player object to the server using the Fetch API, and the player’s data will be stored in the Postgres Database.

**Add a Team**: In order to create a team, select the Add a New Team button on the Teams page. Select 3 players from the drop down menus and click the submit button. This will send a POST request with a Team object to our Postgres database using the Fetch API, and the team’s data will be stored in the Postgres database. The client also uses a GET request to list all the Players that are not on a team prior to populating the drop down menus, since a player is only allowed to be on one team at a time.

**Add a Game**: In order to create a new game entry, select the Add New Game button on the Games page. Select 2 teams from the drop down menus and click the submit button. This will send a POST request with a game object to our Postgres database using the Fetch API, and the game’s data will be stored in the Postgres database. The client also uses a GET request to get the list of Teams and to see which Teams have not been selected so that there are not erroneous records of a team playing itself.



**Reflection:**

Originally, we planned on having 5 players per team, but since Ping Pong can at most be played 2v2 so a team of 3 made much more sense since you can have 2 plus 1 substitute. We wanted to track other stats but realized that there was no feasible way to accurately report and quantify statistics such as shot percentage, volley average, errors, and defensive skill. The score of the two teams was a much simpler method of tracking team skill, but in this case, we believe it to be the most effective. We struggled the most with dynamic updating of the pages, specifically removing a player from the list of possible selections in Player 2/3 on the Teams page once they have already been selected as Player 1/2. If we did this all again we would probably choose a different sport, ping pong wasn’t as exciting as we thought it was.