Async

Tic-Tac-Toe

Game

**Developed By:**

**Umair Shaikh**

1. **Goal of Game:**

Players must arrange their marks so that they form a continuous line of three cells that runs either vertically, horizontally, or diagonally. By blocking the opponent's line from being completed, one opponent can keep the other from winning.

1. **Features:**

* Asynchronous game
* Multiplayer
* Attractive Aesthetic
* User Authentication and Management

1. **Languages and Framework Used:**

|  |  |
| --- | --- |
| Language | JavaScript, CSS, HTML |
| Front-end | ReactJs |
| Back-end | Nodejs, Express |
| Database | getStream(Cloud API) |
| Deployment | Netlify |

1. **Logical explanation of Architectural choices in the project:**

* **Using getStream API:** 
  + I have used getStream API, because I had used it before in my college project (Chatting App using ReactJS). So, I had pleasant experience in this API. I have also worked with MongoDB, but because of short period of time, I used this Cloud based API.
  + This API handles all the Back-end activities such as Authentication, storing and retrieval of user data.
* **Every player will not see their piece as ‘X’:**
  + Whoever gets the first turn will see their piece as ‘X’ and the opponent will see their piece as ‘O’.
  + Because player don't always have to be ‘X’, and the player wants to be ‘O’ sometimes.

1. **Assumptions made:**

* **How to start a game:**
  + I have assumed that the user should enter opponent’s email id for starting the game.
  + As soon as user will enter opponent’s email id and will remain in waiting state until the opponent enters the user's email id.
  + Basically, both users who wants to play should enter each other’s email id to start the game.
* **Layout:**
  + Can work on Desktop as well as on Mobile phone.
  + I used iPhone XE's resolution to build this layout, which is 375 x 667.

1. **Problem Faced:**

* **Starting an asynchronous game with anyone via their email id:**
  + **Problem:** The problem was that when a player goes offline and comes back, how he will still get the game in the same state as before.
  + **Solution:** Recorded every move of both players in the database, so when the offline player comes back to the game, he can see all the moves made by him
* **Creating cookies:**
  + **Problem:** The problem was how to make a cookie for client because I had no experience before.
  + **Solution:** I used universal-cookie library for maintain sessions and creating cookies. The cookie value can be set to any value one wants, however, here I am setting it to the access token which is generated by the server so that I can access it throughout the application.
* **Homepage(Game History):**
  + **Problem:** Can’t get the access of Game lobbies that have been played before.
* **Displaying status – ‘Waiting for you to play’:**
  + **Problem:** Got problem in displaying the status of ‘Your move’ and ‘Opponent’s move’. On both side it is displaying same status.