***Background of the tools and techniques used.***

Tools used are

1. Socket.io used for multiway communication between the server and the client.
2. Express used for setting up the server.
3. HTML used for the basic user interface.
4. CSS used to style the user interface.
5. JavaScript used to dynamically change document elements on the page for user interaction.
6. jQuery used for easy manipulation of document elements.

Techniques Used

1. Event emitting system using Socket.io for communication between client and server.

***Main Components of the app.***

******

The above flowchart is just a basic explanation of how a player can play the game.

Basically my game is based on rooms. There are multiple rooms available at the home page. Each room has a maximum space of 2. If a room has space available, a player can join the room. If two players have joined a specific room, then the game starts. After the game finishes, the players have two options, they can either reset the game to play another match or a player can leave the current room and go back to the home page. If the player wants they can join any other rooms and play or disconnect from the server by closing the tab.

If a game is currently going on in some room, other players can also play in other rooms according to the availability of the rooms. So my system supports multiple sessions of tic toe at the same time.

So to summarize, the main components are

1. Rooms where user can play.
2. Home Page showing availability of rooms

***Special requirements and steps to run the app***

1. Go to my assignment folder and start terminal
2. Use “cd” and get inside my folder
3. Type “node app.js” to start the server
4. Open a browser and type “localhost:3000”
5. The Home page will appear.

***Features I have added***

1. Room system so that multiple users can play at the same time.
2. Update Rooms when space is available so that other players waiting can join.
3. Leave Room option