You have **3 hours** to build a game of Tic Tac Toe in the browser where the user competes against a computer. Tic Tac Toe is played in on a 3x3 grid where players take turns marking the grid in hopes of getting three of their marks in a row, in a column, or diagonal. For example, here is a sample grid and three samples demonstrating a victory in a row, column and a diagonal:

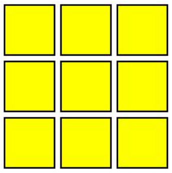	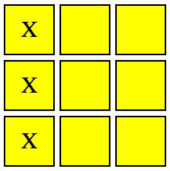	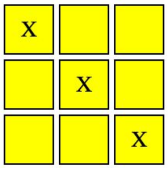	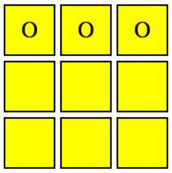
Fig 1: A Sample Tic Tac Toe Grid	Fig 2: A victory for player X, three Xs in a column	Fig 3: A victory for player X, three Xs along a diagonal	Fig 4: A victory for player O, three Os in a row

For your game, you must do the following:

1. **Build and style** your Tic Tac Toe Grid.
2. **Build your functionality.** Assume the user always goes first. When the user clicks on a spot in the grid, the grid should update with a corresponding mark. To get the computer's move, please use the "getmove" call from the attached documentation. The grid should then update with the computer's move.

   The "getmove" call has no knowledge of the current board and returns a random position between 0 and 8. The program will need to check to make sure the computer's move is valid. The program should also throttle the requests.

3. **Publish the board results upon termination of the game.** When either the user wins, the computer wins, or the game ends in a draw, write a json to Arcadia's servers with the "putboard" call from the attached documentation. For example, based upon the numberings, the following board would contain the following JSON:

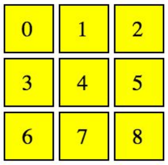	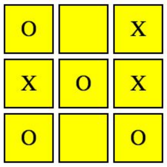
Fig 5: Numberings of the locations on the Tic Tac Toe board	Fig 6: Json to be sent in your data payload: {"final_board": {"0":"O","1":"","2":"X","3":"X","4":"O","5":"X","6":"O","7":"","8":"O"}}

You have a maximum of 3 hours to complete this problem. Upon completion, submit your project as a zip file to the test administrator, preferably via google drive or dropbox rather than email attachment to avoid your email provider rejecting it. This should include your html file, any separate javascript scripts, libraries, stylesheets, etc.

Please do not share your solutions or publish them to any third-party website. Interviewees will be evaluated on **correctness** and **code readability**.

Arcadia API Documentation:

GET /getmove

Resource URL:
https://zpj6onnvm5.execute-api.us-west-2.amazonaws.com/prod/getmove

Response Data Type: JSON
Example Response: {“move_position”: 1}

Rate Limits: 5 calls/second

PUT /putboard

Resource URL:
https://gnswrchqte.execute-api.us-west-2.amazonaws.com/prod/putboard

Data Payload Format: JSON
Example Data Payload: {“final_board”:
{"0":"X","1":"","2":"O","3":"","4":"O","5":"X","6":"O","7":"","8":""}}

Response Data Type: JSON
Example Response: {“success”: “true”}

Rate Limits: 5 calls/second