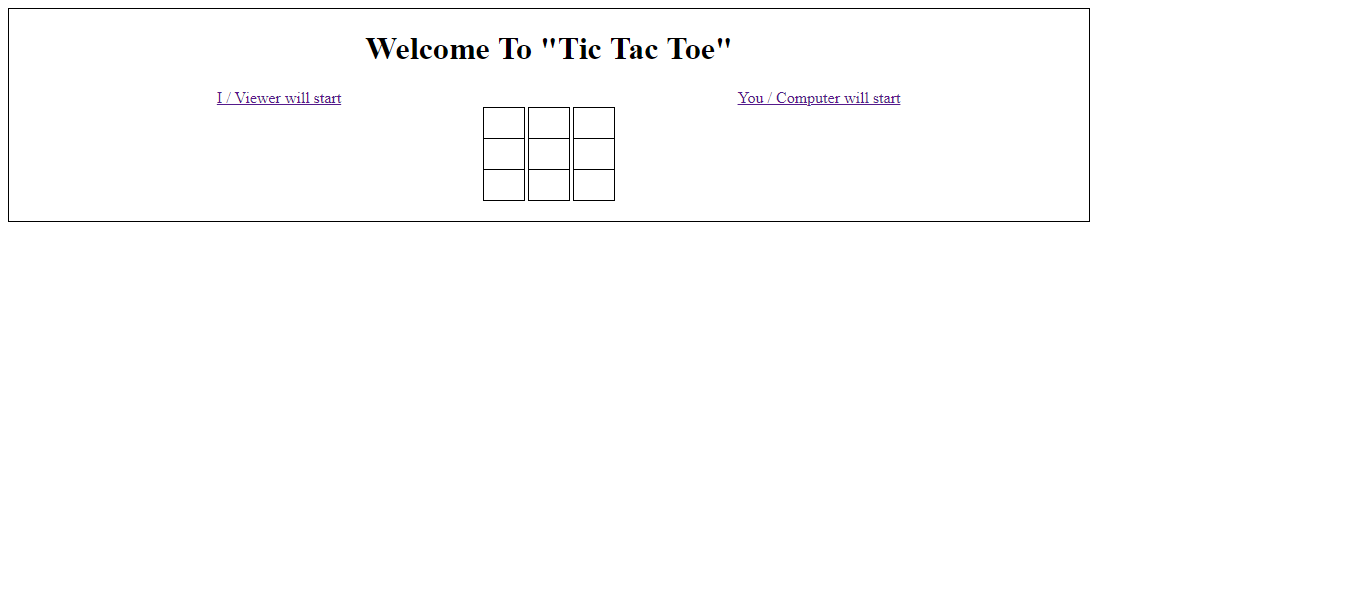
**Q1. How to play the game on web interface?**

Ans: Download code from github and unzip it. Copy code in root directory ie: htdocs/www folder.

**Q2. How it will look like in web interface?**

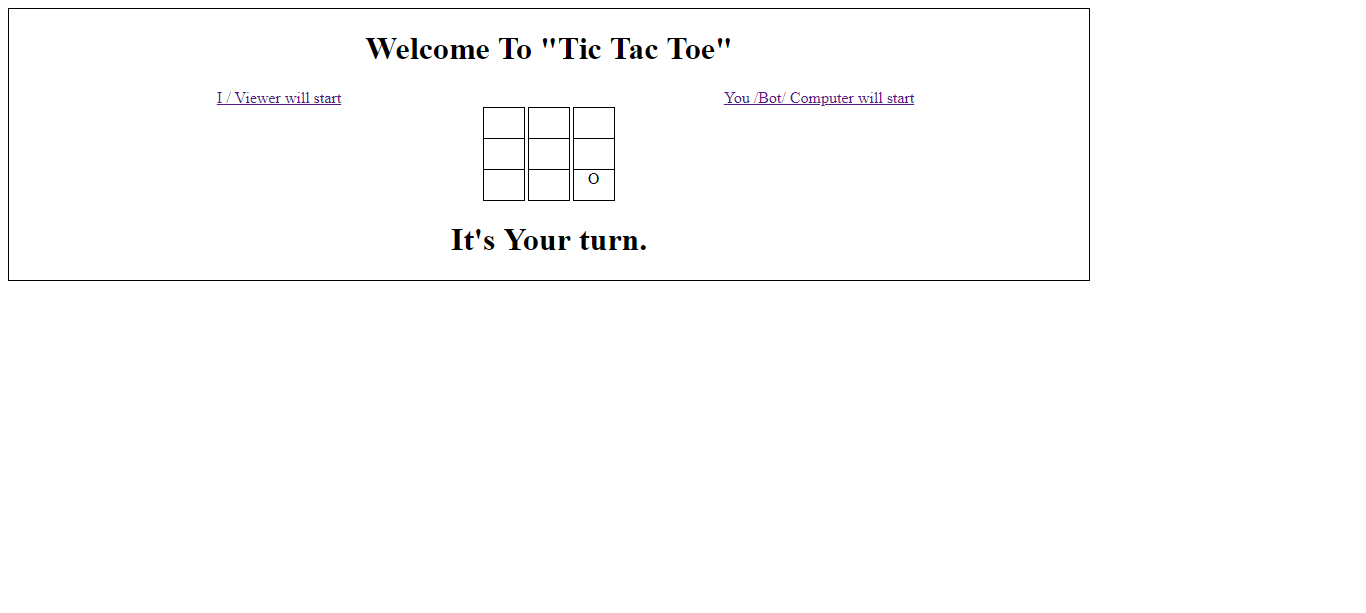
Ans: <http://localhost/Tic-tac-toe/view/>

**Q3: Who can start the game?**

Ans: There are two options that viewer can decide by clicking on any one of the hyperlinks shown above image (**I /Viewer will start** OR **You /(Bot)/Computer will start**).

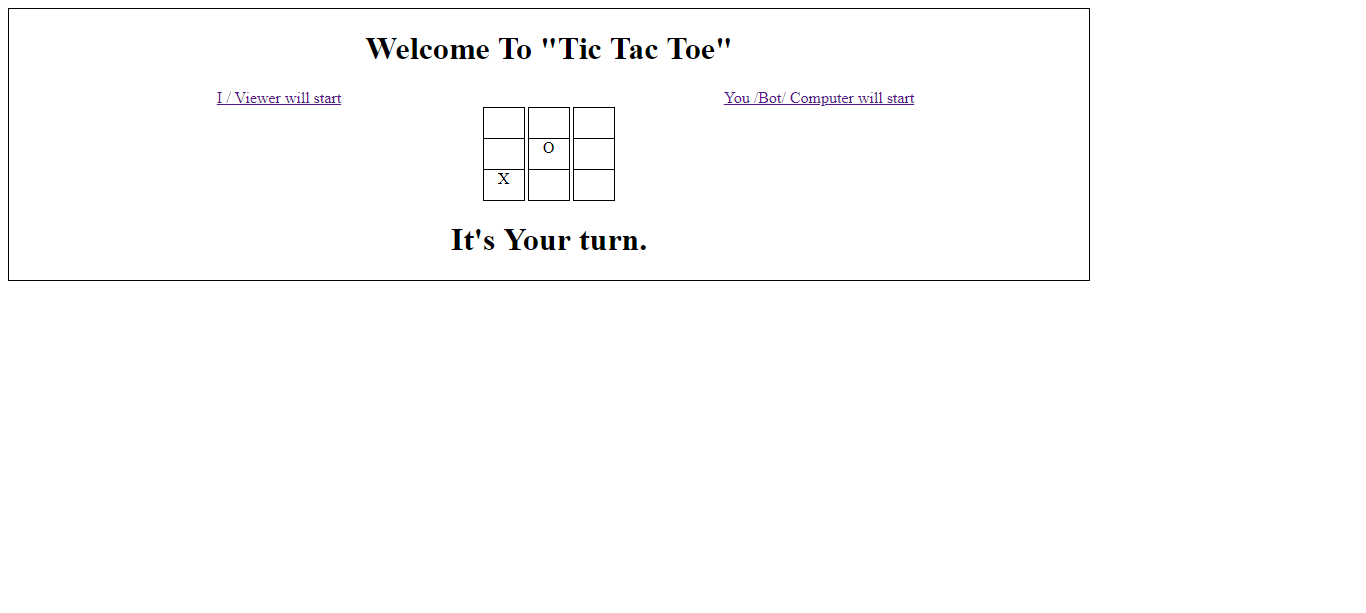
**Q4. What if viewer decides bot will start i.e. (Clicks on the link** [**You /Bot/ Computer will start**](http://localhost/Tic-tac-toe/view/)**) ?**

Ans: Bot /Computer will make its first move and screen will looks like below image ask for your turn. Once viewer will click bot will automatically make his next turn and so on.



**Q5. What if viewer decides I/ Viewer will start i.e. (Clicks on the link** [**I / Viewer will start**](http://localhost/Tic-tac-toe/view/)**) ?**

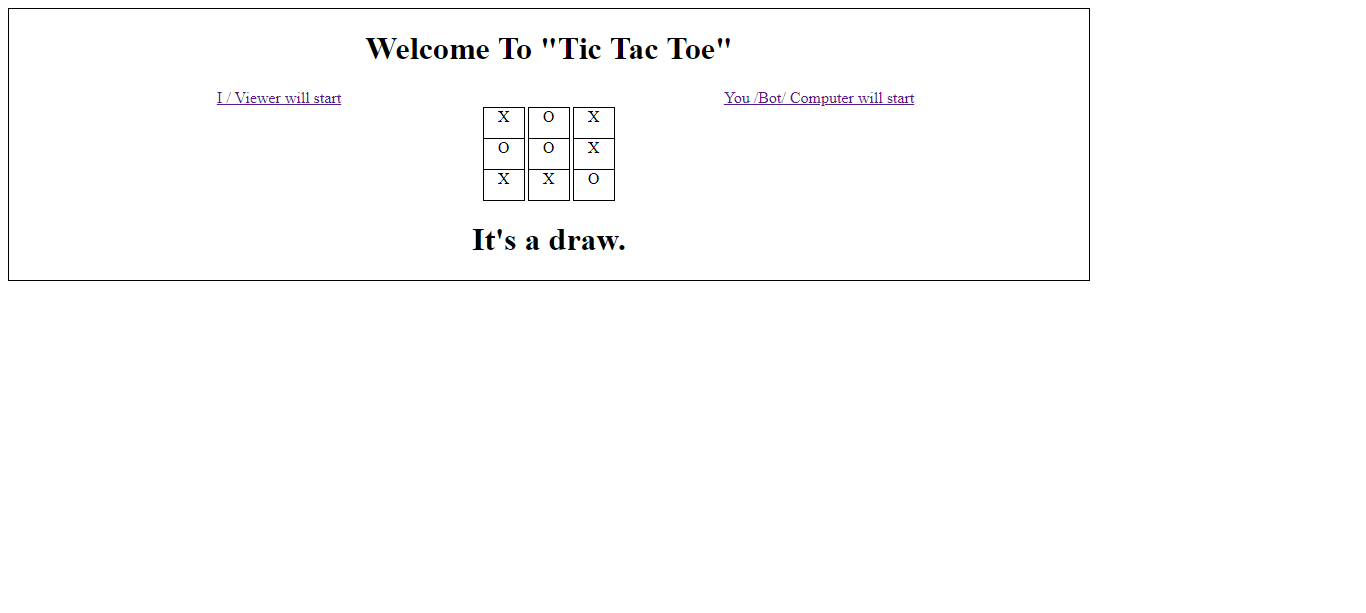
Ans: Viewer needs to click on any block. Screen will look like below image and bot will make his next move one viewer will select any block from game board.



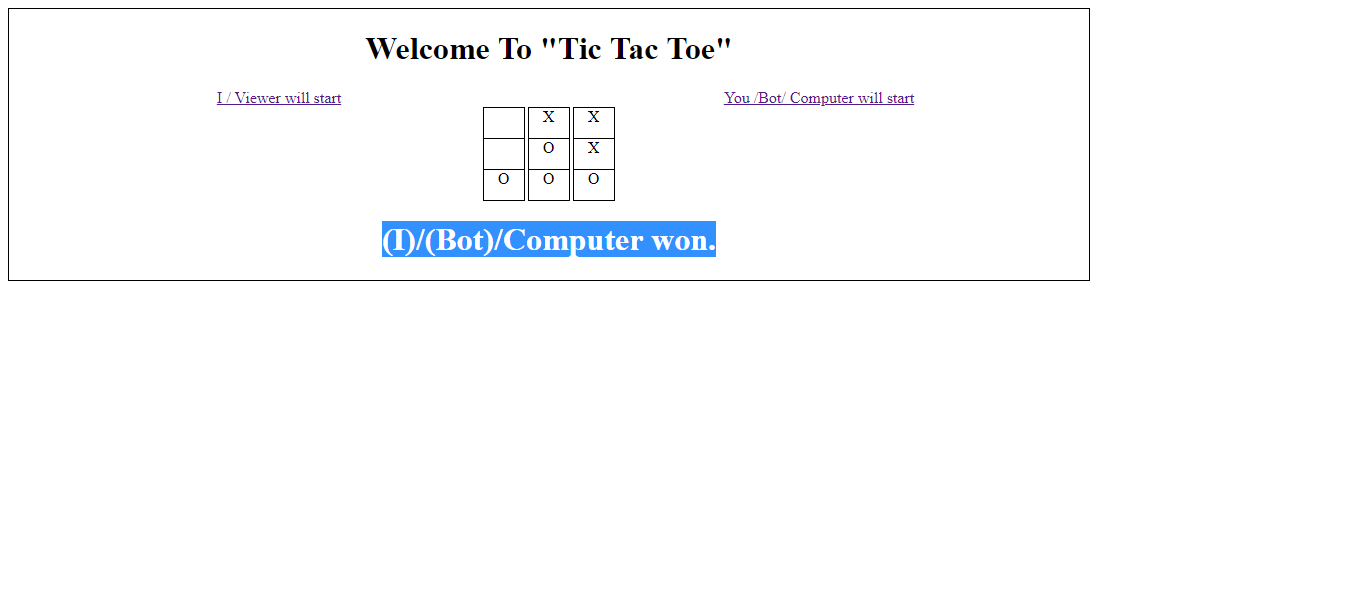
**Q6. How we will come to know game result?**

Ans: It will get displayed below the game board and possible results are.

A) It’s a draw. And screen will look like.



B) (I)/(Bot)/Computer won.



**Q7. How to restart game?**

Ans: By clicking on any one of the hyperlinks shown above image (**I /Viewer will start** OR **You /(Bot)/Computer will start**).

**API**

**Q1: How one can start or restart game ?**

Ans:

A) If user decides how will make first move

Request URL: <http://localhost/Tic-tac-toe/index.php?type=refresh&player=viewer>

Response :

{

* **status**: 0,
* **numMovesLeft**: 9,
* **currentState**:

[

* + [
    - "\_",
    - "\_",
    - "\_"

],

* + [
    - "\_",
    - "\_",
    - "\_"

],

[

* + - "\_",
    - "\_",
    - "\_"

]

],

* **errorStatus**: 0,
* **errorDescription**: "OK"

}

**B)** If user will decide bot will play first

Request URL: <http://localhost/Tic-tac-toe/index.php?type=refresh&player=computer>

Response : {

* **status**: 0,
* **pos**: "2x2",
* **numMovesLeft**: 8,
* **currentState**:

[

* + [
    - "\_",
    - "\_",
    - "\_"

],

* + [
    - "\_",
    - "\_",
    - "\_"

],

* + [
    - "\_",
    - "\_",
    - "O"

]

],

* **errorStatus**: 0,
* **errorDescription**: "OK"

}

**Q1: How viewer will make moves in game?**

Ans: If viewer decides make move at 1x0 position of board

Request: [http://localhost/Tic-tac-toe/index.php?type=makeMove**&pos=1x0**](http://localhost/Tic-tac-toe/index.php?type=makeMove&pos=1x0)

Response:

 {"status":0,

"pos":"0x0",

"numMovesLeft":7,

"currentState":[["O","\_","\_"],["X","\_","\_"],["\_","\_","\_"]],

"errorCode":0,

"errorDescription":"OK",

"botsRecentMove":["0","0","O"]

}

Here status filed will decide winner

1. status = 0 [Game is On]

2. status = “O” [Computer Won]

3 status = “X” [Viewer Won]

Field pos is same as botsRecentMove , it being in UI.

Field numMovesLeft the decision-making if numMovesLeft=0 and status =0

Means game is draw.

currentState Represents the current state of board with all values.

errorCode and errorDescription for used during validation process and return result.

errorCode 0 means there is no error.

Ps: There is also scope of improvement api response itself display the result which currently handled in UI.