

TASK 10 – TIC TAC TOE!

“Great web design without functionality is like a sports car with no engine.” --Paul Cookson

Preface:

As you already know, in this unit we will deep dive into the Model-View-Controller paradigm and learn how to optimise our business logic with it. To do this, we will jump straight into the making of a series of games. We will start this journey with the simplest of all! Tic Tac Toe!

Objectives:

- 1) Create a new app called games, with an html that holds a list of active Tic Tac Toe games in the database. Create a button to start a new game. A Game is defined by:
 - A room name (the only parameter the user should enter manually, make sure it's unique)
 - Owner (user that created it)
 - A board game
 - Active player (player 1 or player 2)
 - Game state (active or won, and if won, by who)
- 2) We should also create a page that will show the game at its current state. Create a grid to show the game board and a title above it with the room name and a subtitle with the current user. Each square should either show an “empty” square or a token. Tip: Use emojis or icons to represent the tokens, it will be prettier this way.
- 3) Wrap the game board around a form, now each square should become a button so that users are able to click on it. Each button should be a submit button with a different value. The action of the form should be this very page.
- 4) Modify your views.py so that you retrieve the information inside the post request. Here is a list of tips to make the game logic. Feel free to not read these tips if you feel confident, and skip to step 5.
 - a. Retrieve the game from the database.

```
game = Game.objects.get(id=request.POST.get('name'))
```

 - b. Retrieve the square selected by the user.

```
square_id = request.POST.get('submit')
```

 - c. Check if there is a winner, the game should continue or it's a tie.
 - d. Once you have modified the game object. Save it to the database.

```
game.save()
```

- e. At the end of the function render your page passing in argument your game object.
- 5) For the final touch try to modify what you have already written so that only the owner can play, the others should be able to watch. Tip: Try to render a form only if the current user is the owner, if not, display the game as it is.
- 6) Once the game is finished, display who won and the game results. Add a little button that closes the room (deletes the entry in the database).

```
game.delete()
```

Submit method:

Play for a bit and upload a short video of you playing the game. Don't forget to upload the code to github!