## **Name: Javeria Manzoor**

**Roll no: BSAI-3C-201**

**Subject:Artificial Intelligence(LAB)**

**Task no:02**

**Submitted to : Sir Rasikh**

## **Mini Project – Tic Tac Toe (Dynamic)**

## **How It Works**

1. The **board** is a list of 9 spaces (representing 3×3 grid).
2. The **players ("X" and "O")** take turns choosing a position (1–9).
3. After each move:  
   * The board is updated and printed.
   * The game checks for a **winner** or **draw**.
4. The game ends when someone wins or all positions are filled.

## **WHY THIS CODE WORKS**

### **1. The Board Representation**

board = [" " for \_ in range(9)]

The board is just a **list of 9 elements**, representing the 9 positions (1–9).  
Example layout:  
  
 index → 0 | 1 | 2

3 | 4 | 5

6 | 7 |

Each element starts as " " (empty).  
a player makes a move, "X" or "O" replaces that space.

### **2. Displaying the Board**

def print\_board():

print(f"{board[0]} | {board[1]} | {board[2]}")

print("--+---+--")

print(f"{board[3]} | {board[4]} | {board[5]}")

print("--+---+--")

print(f"{board[6]} | {board[7]} | {board[8]}")

* This function simply **prints the 3×3 grid** using values from the list.
* It makes the game dynamic — after each move, it shows updated positions.

### **3. Switching Turns Dynamically**

current\_player = "X"

current\_player = "O" if current\_player == "X" else "X"

The game always starts with "X".  
After every valid move, it switches to the **other player** automatically using this one-line condition.

If it’s "X", it changes to "O".  
If it’s "O", it changes to "X".

### **4. Checking for a Winner**

### win\_conditions = [

[0, 1, 2], [3, 4, 5], [6, 7, 8], # rows

[0, 3, 6], [1, 4, 7], [2, 5, 8], # columns

[0, 4, 8], [2, 4, 6] # diagonals

]

* These are all **possible winning combinations** (row, column, diagonal).
* The function checks if **all 3 positions** in any of these lists contain the same player's mark ("X" or "O").  
   If yes → that player **wins**.

### **5. Checking for a Draw**

def is\_full():

return " " not in board

If no empty space is left and no winner is found, the game is declared a **draw**.

**6. Why the While Loop Works**

while True:

# take move, check winner or draw, switch players

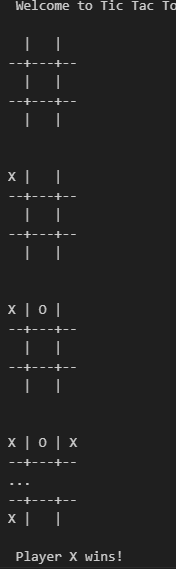
The while True keeps the game running until someone **wins** or the board is **full**.  
break is used to stop the loop when the game ends.

### **7. Input Validation**

if move < 0 or move > 8 or board[move] != " ":

print(" Invalid move! Try again.")

This ensures players can’t overwrite already filled spots or enter invalid numbers.



.