Task 1

Aim: Build a Tic Tac Toe Web Application
Theory:

1. Project Structure

- **HTML**: Defines the structure and layout of the game interface.
- **CSS**: Styles the game to make it visually appealing.
- **JavaScript**: Implements the game logic, interactions, and dynamic updates.
- **Optional**: Use a framework (e.g., React, Vue) for a modern approach.

2. Core Components

1. **Game Board**
   - A 3x3 grid (HTML `<div>` elements styled with CSS).
   - Each cell acts as a clickable area.
2. **Game State**
   - Tracks the current state of the board (e.g., a 2D array or a 1D array of 9 elements).
   - Keeps track of the current player (X or O).
3. **Game Logic**
   - Handles player moves, alternating turns.
   - Detects winning conditions:
     - Three matching symbols in a row, column, or diagonal.
   - Checks for a draw (no moves left).
   - Resets the game when necessary.
4. **User Interface**
   - Updates the board dynamically based on player moves.
   - Displays messages for wins, draws, or errors (e.g., invalid moves).
5. **Interactivity**
   - Use event listeners (e.g., `onclick`) for player interaction.
   - Provide a reset button for starting a new game.

3. Steps to Implement

1. **HTML Structure**
   - Create a container for the grid.
   - Include placeholders for game status and reset button.
2. **Styling with CSS**
   - Style the grid with borders and alignments.
   - Add hover effects to enhance user experience.
3. **JavaScript Logic**
   - Initialize the board and players.
   - Create functions for:
     - Handling player moves.
     - Checking the game status (win/draw).
     - Resetting the game.
   - Update the UI based on the game state.

4. **Testing**
   ○ Verify edge cases (e.g., multiple clicks on the same cell).
   ○ Test all winning and draw scenarios.

4. Optional Enhancements

● Add a score tracker.
● Implement AI for single-player mode.
● Style the application for responsive design.
● Use animations for cell transitions.
● Deploy the app online using GitHub Pages or hosting platforms like Netlify

Code:

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, init
    <title>Tic Tac Toe</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            text-align: center;
            margin-top: 50px;
            background-color: #f4f4f4;
        }
        h1 {
            color: #333;
        }
        #game {
            display: grid;
            grid-template-columns: repeat(3, 100px);
            grid-gap: 5px;
            justify-content: center;
        }
        .cell {
            width: 100px;
            height: 100px;
            display: flex;
            align-items: center;
            justify-content: center;
            border: 2px solid #333;
            font-size: 24px;
            font-weight: bold;
            cursor: pointer;
            background-color: #fff;
            transition: background-color 0.3s;
        }
```

Output:

Tic Tac Toe

Play with Friend   Play with Computer

X	O	X
X	O	X
O	X	O

It's a draw!