**TIC-TAC-TOE GAME DESIGN**

**Submitted by**

**Name of the Students:** Pradyumna Bhattacharya

**Enrolment Number:** 12022002011051

**Section:** I

**Class Roll Number:** 52  
**Stream:**  EE

**Subject:** Programming for Problem Solving

**Subject Code:** ESC103

**Department:**Basic Science and Humanities

Under the supervision of

Swarrnendu Ghosh

**Academic Year: 2022-26**

PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE FIRST SEMESTER



**DEPARTMENT OF BASIC SCIENCE AND HUMANITITES**

**INSTITUTE OF ENGINEERING AND MANAGEMENT, KOLKATA**



**CERTIFICATE OF RECOMMENDATION**

We hereby recommend that the project prepared under our supervision by **Pradyumna Bhattacharya,** entitled TIC-TAC-TOE GAME DESIGN PROJECT REPORT be accepted in partial fulfillment of the requirements for the degree of partial fulfillment of the first semester.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Head of the Department Project Supervisor

Basic Sciences and Humanities

IEM, Kolkata

# INTRODUCTION :

# This project is assigned to me for developing a Library Management System with

# the help of basic C programming language.

# The basic aim of the project is to create a library management system where we

# need to put up basic book and reader details and thereby with the help of c

# programming, we have to create a portal (.exe file) for adding new books,

# searching books,adding new reader, searching readers, issuing books, submitting

# books, deleting books, deleting readers and finally seeing all the books in the

## VARIABLE DESCRIPTION :

## The different variables used in this project are listed under:-

## 1. int- To store integer datatypes.

## 2. char- To store character datatypes

## FUNCTION DESCRIPTION :

The different functions (structures) used in this project are listed under:-

1. reader- For storing the required reader details vis. name, id and due date.

2. book- For storing the required book details viz. name, author, availability

and reader info

# Programs

#include <stdio.h>

// Function prototypes

void print\_board(char board[3][3]);

int check\_win(char board[3][3], char player);

int main() {

char board[3][3] = { // Initialize the game board

{'-', '-', '-'},

{'-', '-', '-'},

{'-', '-', '-'}

};

int turn = 0; // Player 1's turn is 0, Player 2's turn is 1

int row, col; // To hold the row and column input by the player

printf("Welcome to Tic-Tac-Toe!\n\n");

print\_board(board);

while (1) { // Loop until the game is over

char player = (turn % 2 == 0) ? 'X' : 'O'; // Determine the current player

printf("Player %c's turn.\n", player);

printf("Enter row (0-2): ");

scanf("%d", &row);

printf("Enter column (0-2): ");

scanf("%d", &col);

if (board[row][col] == '-') { // Make sure the chosen spot is empty

board[row][col] = player; // Place the player's symbol on the board

print\_board(board);

if (check\_win(board, player)) { // Check if the player has won

printf("Player %c wins!\n", player);

break; // End the game

}

turn++; // Switch to the other player's turn

}

else {

printf("That spot is already taken. Try again.\n");

}

if (turn == 9) { // If all spots are taken and no one has won, it's a tie

printf("It's a tie!\n");

break; // End the game

}

}

return 0;

}

// Prints the current state of the game board

void print\_board(char board[3][3]) {

printf("\n");

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

printf("%c ", board[i][j]);

}

printf("\n");

}

printf("\n");

}

// Returns 1 if the given player has won, 0 otherwise

int check\_win(char board[3][3], char player) {

for (int i = 0; i < 3; i++) {

// Check rows

if (board[i][0] == player && board[i][1] == player && board[i][2] == player) {

return 1;

}

// Check columns

if (board[0][i] == player && board[1][i] == player && board[2][i] == player) {

return 1;

}

}

// Check diagonals

if (board[0][0] == player && board[1][1] == player && board[2][2] == player) {

return 1;

}

if (board[0][2] == player && board[1][1] == player && board[2][0] == player) {

return 1;

}

return 0; // If the player hasn't won yet, return 0

}

# Outputs

Welcome to Tic-Tac-Toe!

- - -

- - -

- - -

Player X's turn