Internship Report

Internship report submitted in fulfillment of the requirement for the certificate of the internship

By

Your Name: **ARPAN KUNDU**

College Name: **SANAKA EDUCATIONAL TRUST GROUP OF INSTITUTIONS**

Domain Name: **JAVA DEVELOPER**

Batch (JUNE-JULY)

InternJunction

EMPOWERING MINDS

Date of Submission: 30th JULY

2024

DECLARATION

I hereby declare that the projects entitled "**TIC-TAC-TOE GAME**" and submitted for the completion of my internship at InternJunction, are my original work. These projects have not been submitted for any degree, diploma, or other internship programs at any other organizations.

Name:	Arpan Kundu
Place:	Durgapur
Date:	20th july

ACKNOWLEDGEMENT

It is my proud privilege to express my heartfelt gratitude to several individuals who have assisted me directly or indirectly in completing this project. First and foremost, I would like to extend my deepest appreciation to my internship guides at InternJunction. Their sincere guidance, unwavering support, and invaluable inspiration have been instrumental in the successful completion of this internship.

The insights and knowledge I gained during this period have been profound and enlightening, significantly broadening my understanding of the topics at hand. This experience has not only deepened my academic and professional knowledge but also prepared me for future challenges and opportunities.

I am genuinely grateful for the opportunities provided by InternJunction, which have allowed me to explore and engage with new and exciting avenues of knowledge. This internship has been an enriching journey, and I am confident that the skills and insights I have gained will be of immense benefit in my future endeavours.

To all those who have contributed to this project, whether directly or indirectly, I extend my sincere thanks. Your support and encouragement have been invaluable, and I am truly indebted to you all.

TABLE OF CONTENTS

INTRODUCTION

Description:

Tic-Tac-Toe, also known as Noughts and Crosses, is a classic game that has been enjoyed by people of all ages for generations. This project aims to recreate this simple yet engaging game in a console-based format using basic programming principles. By developing this game, we aim to provide a fun and interactive way to practice and enhance programming skills, particularly in the areas of game logic, user input handling, and state management.

Objective:

The primary objective of this project is to develop a console-based Tic-Tac-Toe game that provides an engaging and interactive experience for users. The game will facilitate a two-player mode where users can play against each other on the same console. This project aims to enhance programming skills, specifically in understanding game logic, implementing user input, and handling game states.

Scope:

<u>This project covers the following areas:</u>

Game Board Implementation: Creating a 3x3 grid that represents the Tic-Tac-Toe board.

Player Input: Allowing two players to enter their moves alternately.

Game Logic: Implementing the logic to check for win conditions, draw conditions, and invalid moves.

User Interface: Designing a simple text-based user interface to display the game board and status messages.

Game Flow Control: Ensuring the game progresses smoothly from start to end, including restarting or ending the game based on player choices.

The game will display the updated board after each move and notify the players of the game's status. Once a win or draw condition is reached, the game will announce the result and offer the players the option to restart or exit.

Result:

Upon completion, the project will result in a fully functional console-based Tic-Tac-Toe game. Players will be able to:

1.Enter their moves alternately.

2.See real-time updates of the game board after each move.

3.Receive feedback on the game status (win, lose, or draw).

The successful implementation of this project will demonstrate a foundational understanding of game development principles in a console environment and provide a basis for more complex game development projects in the future. This project not only serves as a fun exercise in programming but also as a stepping stone for learning more advanced concepts in game development.

METHODOLOGY

The TicTacToe class and the GameAlgorithms class are designed to showcase the principles of modularity in programming. The GameAlgorithms class is particularly structured with six distinct methods, each serving a specific function:

1. showBoard(): Displays the current state of the game board.

2. operationOnTable(): Handles operations on the game board, such as updating it based on user moves.

3. decideUser(): Determines which user's turn it is to play.

4. winChecker(): Checks the game board for a winning condition.

5. takeInput(): Manages the input from the users.

6. moveOfUsers(): Executes the moves of the users based on the input received.

This modular design demonstrates the usefulness of modularity in programming languages, allowing for better organization, readability, maintainability, and scalability of the code. Each method encapsulates a specific functionality, making the overall system easier to understand and modify.

Main Class: TicTacToe

```java
public class TicTacToe {
    public static void main(String[] args) {
        System.out.println("........LET'S PLAY THE ONE AND ONLY TIC-TAC-TOE 
GAME.......");
        GameAlgorithm drawTable=new GameAlgorithm();
        drawTable.showBoard();
        drawTable.decideUser();
    }
}
```

The main class, `TicTacToe`, serves as the entry point for the game. Within the `main` method of this class, several important steps are taken to set up and start the game:

1. Create an Object of GameAlgorithms: An instance of the `GameAlgorithms` class is created. This object will be used to access various game-related methods.

2. Display the Initial Game Board: The `showBoard()` method is called using the `GameAlgorithms` object. This method displays the initial state of the TicTacToe table, allowing players to see the starting configuration.

3. Conduct a Toss to Decide the First Player: The `decideUser()` method is invoked next. This method performs a toss to determine which player will make the first move and assigns the symbols X and O to the players.

These steps ensure that the game is properly initialized, the board is displayed, and the first player and their symbols are decided, providing a clear and organized start to the TicTacToe game.

GameAlgorithm class:

```java
class GameAlgorithm{
    char board[][]={{'A','B','C'},
                    {'D','E','F'},
                    {'G','H','I'}};
    static int conut=0;
    String Xuser;
    String Ouser;
    public void showBoard()
    {
        System.out.print(board[0][0]+" | "+board[0][1] +" | "+ board[0][2]);
        System.out.println();
        System.err.print(board[1][0]+" | "+board[1][1] +" | "+ board[1][2]);
        System.out.println();
        System.out.print(board[2][0]+" | "+board[2][1] +" | "+ board[2][2]);
        System.out.println();
    }
    public void operationOnTable(char pos, char symbol)
    {
        conut++;
        char table[][]=board;
            for(int i=0;i<table.length;i++)
            {
                for(int j=0;j<table[i].length;j++)
                {
                    if(table[i][j]==pos)
                        table[i][j]=symbol;
                }
            }
        board=table;
        showBoard();
        if(conut==9)
        {
            System.out.println("So, the final TIC-TAC-TOE board is: ");
            showBoard();
            winChecker();
        }
    }
    public void takeInput(char input,String name)
    {
        char symbol=input;
        Scanner sc=new Scanner(System.in);
        System.out.println(name+" Please enter your position: ");
        char pos=sc.next().toUpperCase().charAt(0);
        if(pos=='X' || pos=='O')
        {
            System.out.println("This are not applicable... Please enter the
provided named position to continue the game!!!!");
            takeInput(input, name);
        }
        if(pos=='A' || pos== 'B' || pos=='C' || pos=='D' || pos=='E' ||
pos=='F' || pos=='G' || pos=='H' || pos=='I')
        {
            operationOnTable(pos,symbol);
        }
        else
        {
             System.out.println("Please enter the valid position not anything
else....");
            takeInput(input, name);
        }
        sc.close();
    }
    public void decideUser()
    {
        String Xuser;
        String Ouser;
        Scanner sc=new Scanner(System.in);
        System.out.println("Please enter your name: ");
        System.out.println("Who will choose 'X' as symbol: ");
        Xuser=sc.next();
        System.out.println("Who will choose 'O' as symbol: ");
        Ouser=sc.next();
        Random random=new Random();
        int toss=2;
        int win=random.nextInt(toss);
        if(win==0)
            System.out.println(Xuser+" is Win the toss.........");
        else
            System.out.println(Ouser+" is win the toss.........");
        GameAlgorithm drawTable=new GameAlgorithm();
```

```java
        drawTable.moveOfUsers(Xuser,Ouser,win);
        sc.close();
    }
    public void moveOfUsers(String user1, String user2, int win)
    {
        Xuser=user1;
        Ouser=user2;
        int firstMove=win;
        if (firstMove==0){
            takeInput('X',Xuser);
            firstMove=1;
            moveOfUsers(Xuser, Ouser, firstMove);
        }
        else{
            takeInput('O',Ouser);
            firstMove=0;
            moveOfUsers(Xuser, Ouser, firstMove);
        }

    }
    public void winChecker()
    {
        if((board[0][0]=='X' && board[1][0]=='X' && board[2][0]=='X') ||
(board[0][0]=='X' && board[0][1]=='X' && board[0][2]=='X') ||
(board[0][1]=='X' && board[1][1]=='X' && board[2][1]=='X') ||
(board[1][0]=='X' && board[1][1]=='X' && board[1][2]=='X') ||
(board[0][2]=='X' && board[1][2]=='X' && board[2][2]=='X') ||
(board[2][0]=='X' && board[2][1]=='X' && board[2][2]=='X')){
            System.out.println(Xuser+" won the match");
            System.out.println(Ouser+" loss the match");
            System.exit(0);
        }
        else if((board[0][0]=='O' && board[1][0]=='O' && board[2][0]=='O') ||
(board[0][0]=='O' && board[0][1]=='O' && board[0][2]=='O') ||
(board[0][1]=='O' && board[1][1]=='O' && board[2][1]=='O') ||
(board[1][0]=='O' && board[1][1]=='O' && board[1][2]=='O') ||
(board[0][2]=='O' && board[1][2]=='O' && board[2][2]=='O') ||
(board[2][0]=='O' && board[2][1]=='O' && board[2][2]=='O')){
            System.out.println(Ouser+" won the match.");
            System.out.println(Xuser+" loss the match");
            System.exit(0);
        }
        else{
            System.out.println("draw-match");
            System.exit(0);
        }
    }
}
```

The GameAlgorithm class is the backbone of the TicTacToe game, managing the core functionalities and ensuring a smooth gameplay experience. Here's an organized and styled breakdown of its key components:

Methods in the `GameAlgorithm` Class:

1. showBoard():

Description:

Displays the current state of the game board.

Functionality:

1.This method iterates through the `board` 2D array and prints each cell, providing a visual representation of the game board to the players after each move.

2. operationOnTable():

Description:

Handles the logic for placing a player's symbol on the board and updates the display.

Functionality:

1.Takes the player's chosen position, updates the `board` array with the player's symbol, and increments the `count` variable.

2.Calls the `showBoard()` method to display the updated board.

3.Checks if the `count` has reached 9, indicating a full board, and then calls `winChecker()` to check for a winner or declare a draw.

3. takeInput():

Description:

Manages and validates the player's input.

Functionality:

1.Receives the player's symbol (X or O) and their name.

2.Prompts the player to choose a position on the board to place their symbol.

3.Validates the chosen position to ensure it is not already occupied and is within the valid range.

4.If the position is valid, updates the board with the player's symbol; otherwise, prompts the player to select a different position.

4. decideUser():

Description:

Decides which player will use X and which will use O, and who makes the first move.

Functionality:

1.Conducts a toss using the `Random()` method to determine which player will make the first move.

2.Assigns symbols (X and O) to the players based on the toss result.

3.Calls the `moveOfUsers()` method with the players' names and the toss result to start the game.

5.moveOfUsers():

Description:

Manages player turns and updates the game state.

Functionality:

1.Alternates between players for their turns, ensuring each player gets to move.

2.Takes the player's name and an integer to store the winning result.

3.Calls `takeInput()` for the current player's move and updates the `winResult` based on the game status.

4.Repeats the process until a win condition is met or the board is full.

6. winChecker():

Description:

Checks for a winning condition or a draw and declares the game outcome.

Functionality:

1.Analyzes the `board` array to check for a winning combination of symbols (three in a row, column, or diagonal).

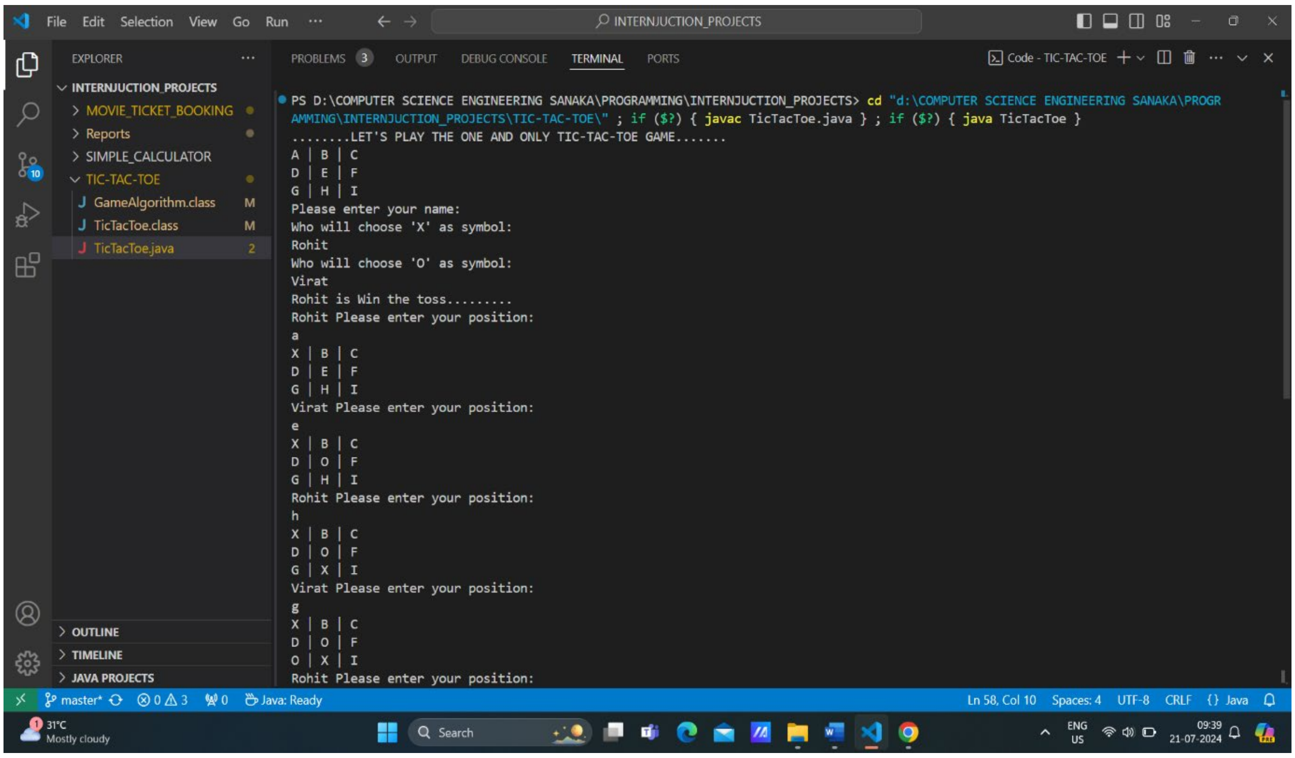
2.Declares the winner if a winning condition is met.

3.If no winning condition is met and the board is full (`count` is 9), declares the game as a draw.

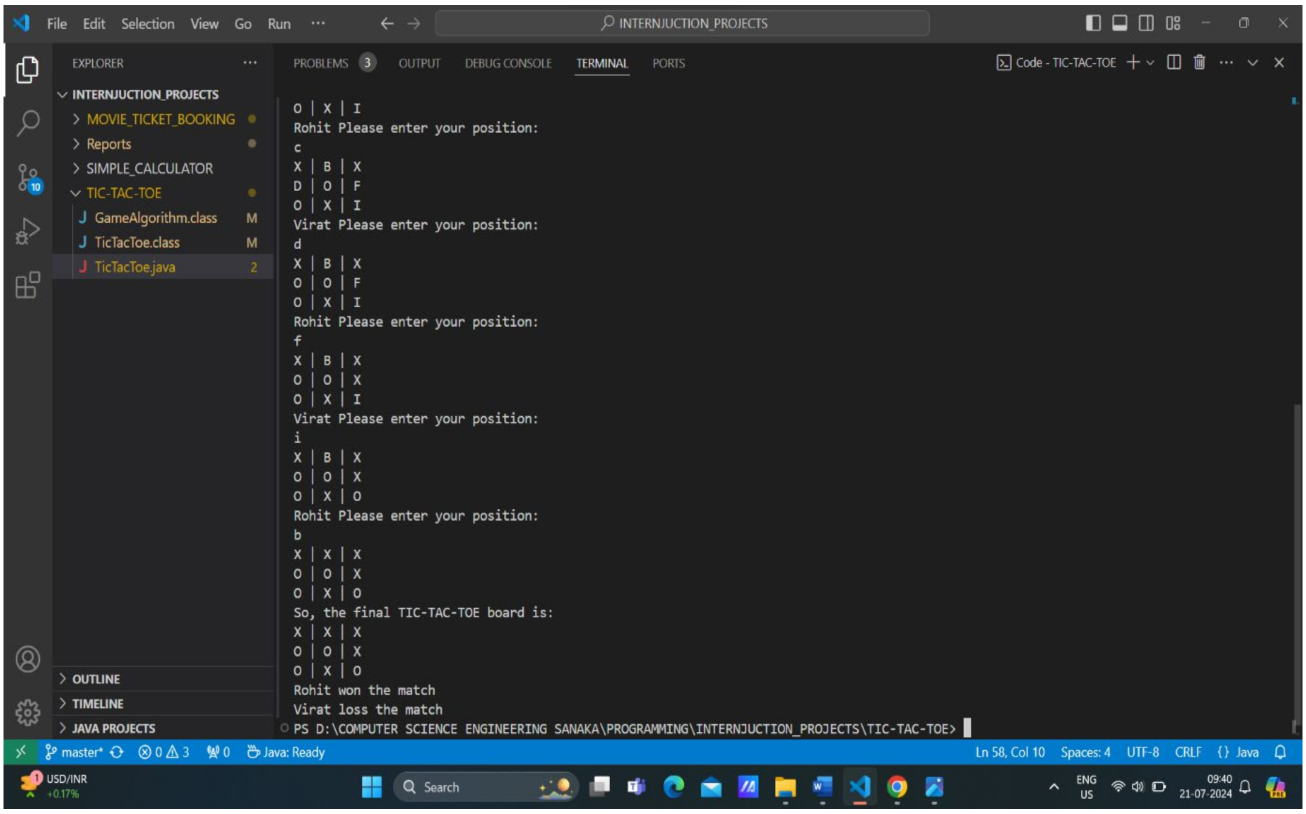
4.Terminates the game and displays the final result to the players.

These methods collectively ensure the smooth execution of the TicTacToe game, from displaying the board and handling player input to determining the winner and managing game turns. The modular design enhances the clarity and maintainability of the code.

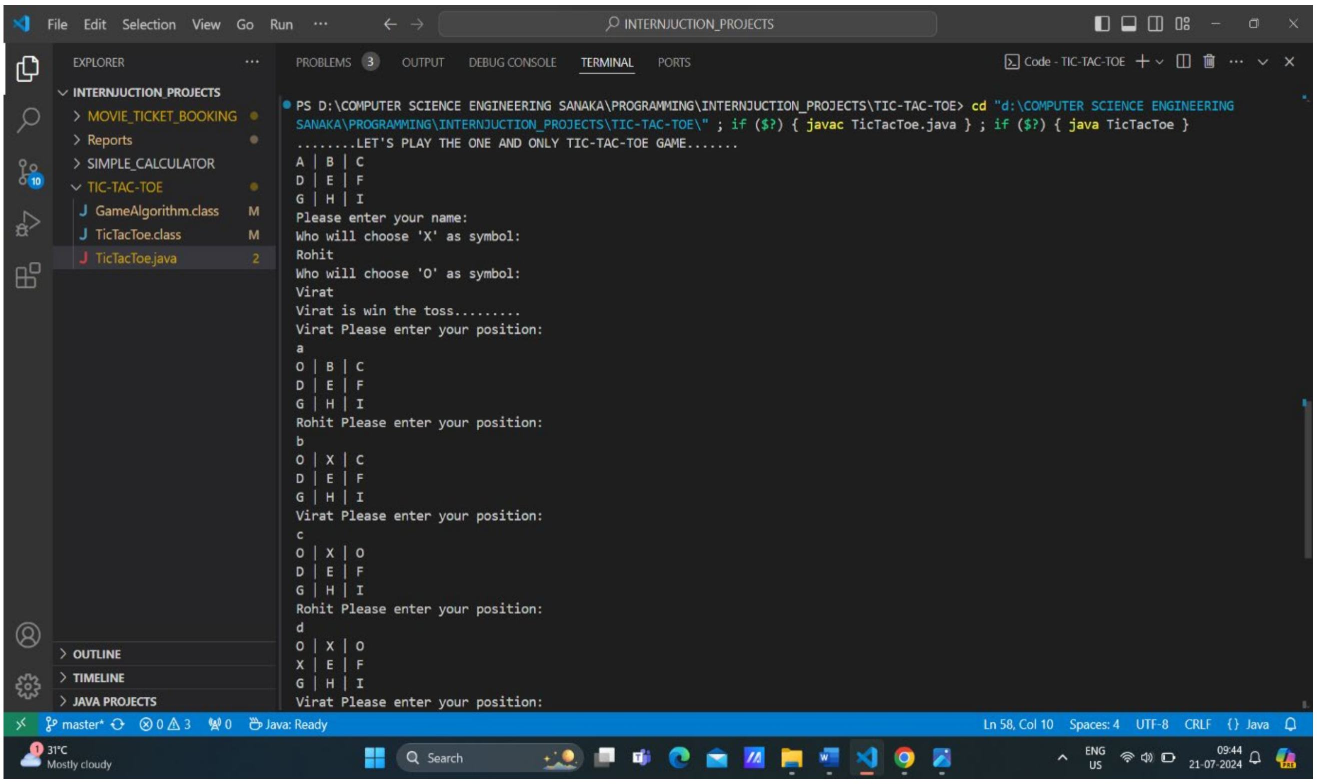
RESULTS



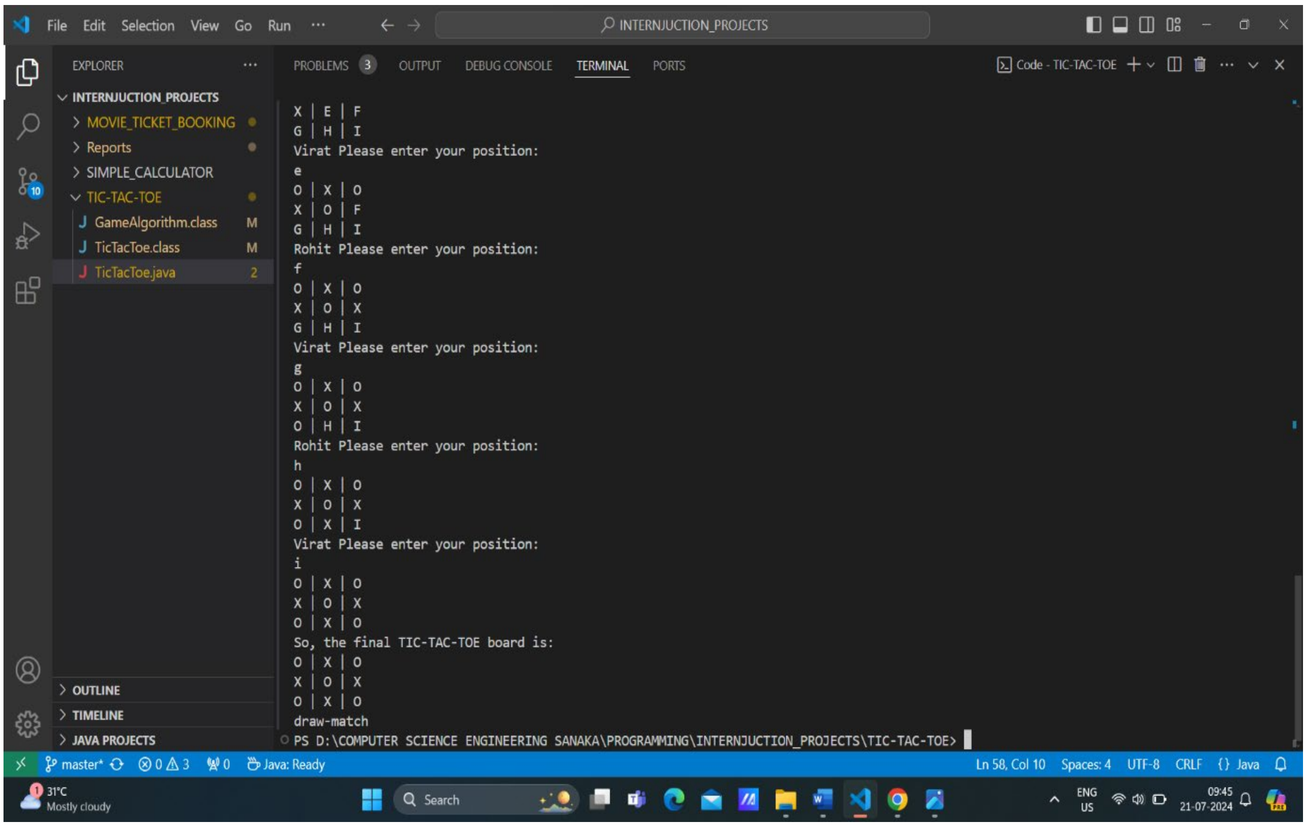
```
PS D:\COMPUTER SCIENCE ENGINEERING SANAKA\PROGRAMMING\INTERNJUCTION_PROJECTS> cd "d:\COMPUTER SCIENCE ENGINEERING SANAKA\PROGR
AMMING\INTERNJUCTION_PROJECTS\TIC-TAC-TOE\" ; if ($?) { javac TicTacToe.java } ; if ($?) { java TicTacToe }
........LET'S PLAY THE ONE AND ONLY TIC-TAC-TOE GAME.......
A | B | C
D | E | F
G | H | I
Please enter your name:
Who will choose 'X' as symbol:
Rohit
Who will choose 'O' as symbol:
Virat
Rohit is Win the toss.........
Rohit Please enter your position:
a
X | B | C
D | E | F
G | H | I
Virat Please enter your position:
e
X | B | C
D | O | F
G | H | I
Rohit Please enter your position:
h
X | B | C
D | O | F
G | X | I
Virat Please enter your position:
g
X | B | C
D | O | F
O | X | I
Rohit Please enter your position:
```



```
O | X | I
Rohit Please enter your position:
c
X | B | X
D | O | F
O | X | I
Virat Please enter your position:
d
X | B | X
O | O | F
O | X | I
Rohit Please enter your position:
f
X | B | X
O | O | X
O | X | I
Virat Please enter your position:
i
X | B | X
O | O | X
O | X | O
Rohit Please enter your position:
b
X | X | X
O | O | X
O | X | O
So, the final TIC-TAC-TOE board is:
X | X | X
O | O | X
O | X | O
Rohit won the match
Virat loss the match
PS D:\COMPUTER SCIENCE ENGINEERING SANAKA\PROGRAMMING\INTERNJUCTION_PROJECTS\TIC-TAC-TOE>
```



```
PS D:\COMPUTER SCIENCE ENGINEERING SANAKA\PROGRAMMING\INTERNJUCTION_PROJECTS\TIC-TAC-TOE> cd "d:\COMPUTER SCIENCE ENGINEERING
SANAKA\PROGRAMMING\INTERNJUCTION_PROJECTS\TIC-TAC-TOE\" ; if ($?) { javac TicTacToe.java } ; if ($?) { java TicTacToe }
........LET'S PLAY THE ONE AND ONLY TIC-TAC-TOE GAME.......
A | B | C
D | E | F
G | H | I
Please enter your name:
Who will choose 'X' as symbol:
Rohit
Who will choose 'O' as symbol:
Virat
Virat is win the toss.........
Virat Please enter your position:
a
O | B | C
D | E | F
G | H | I
Rohit Please enter your position:
b
O | X | C
D | E | F
G | H | I
Virat Please enter your position:
c
O | X | O
D | E | F
G | H | I
Rohit Please enter your position:
d
O | X | O
X | E | F
G | H | I
Virat Please enter your position:
```



```
X | E | F
G | H | I
Virat Please enter your position:
e
O | X | O
X | O | F
G | H | I
Rohit Please enter your position:
f
O | X | O
X | O | X
G | H | I
Virat Please enter your position:
g
O | X | O
X | O | X
O | H | I
Rohit Please enter your position:
h
O | X | O
X | O | X
O | X | I
Virat Please enter your position:
i
O | X | O
X | O | X
O | X | O
So, the final TIC-TAC-TOE board is:
O | X | O
X | O | X
O | X | O
draw-match
PS D:\COMPUTER SCIENCE ENGINEERING SANAKA\PROGRAMMING\INTERNJUCTION_PROJECTS\TIC-TAC-TOE>
```

CONCLUTION

Successful Completion of the Simple Calculator Project

The Tic-Tac-Toe Game (Console-Based) project provided by InternJunction has been successfully completed, marking a significant achievement in practical programming.

Practical Application of Knowledge

We applied theoretical knowledge effectively, implementing modularity and object-oriented programming concepts to develop a functional and user-friendly application.

Achievement of Objectives

The project met its primary objective of creating a user-friendly tic-tac-toe game and its educational goal of serving as a valuable learning resource for beginners in coding.

Skill Enhancement

Our programming skills were significantly enhanced through hands-on experience with Java, user input handling, error management, and modularity programming.

Positive Impact on Professional Growth

The project positively impacted our professional growth by improving problem-solving abilities, strengthening our understanding of OOP principles, and providing insights into industry-level software development.

Overall Experience

The project was enriching and fulfilling, solidifying our technical skills and boosting our confidence in real-world programming tasks. The knowledge and experience gained will contribute to our future success in software development.

We are grateful to InternJunction for this valuable opportunity and look forward to applying these skills in future endeavors.

FUTURE WORK

1. Graphical User Interface (GUI):

- Develop a graphical interface to replace the console-based output, enhancing user interaction and visual appeal. Tools like Swing (Java) or Tkinter (Python) can be used to create an intuitive and engaging game window.

2. AI Opponent:

- Introduce an artificial intelligence opponent for single-player mode. Implement algorithms such as Minimax to enable the AI to make strategic moves and provide a challenging gameplay experience.

3. Network Play:

- Enable multiplayer functionality over a network, allowing players to compete with others remotely. This involves setting up server-client architecture to facilitate online gameplay.

4. Score Tracking:

- Add a feature to track and display scores across multiple games. This could include maintaining statistics for wins, losses, and draws, and presenting them in a leaderboard format.

5. Enhanced Game Modes:

- Expand the game with additional modes, such as tournament play or various difficulty levels. This would provide players with a range of options and increase the game's replay value.

These enhancements aim to improve the overall gaming experience, adding complexity and interactivity to the TicTacToe project.