A

Learning Project-I Report

On

**“TIC TAC TOE”**

Submitted in partial fulfillment of

The requirements for the 3rd Semester Sessional Examination of

BACHELOR OF TECHNOLOGY

IN

**COMPUTER SCIENCE & ENGINEERING**

By

Sakshi Kumari Sinha(22UG010098)

Ekata Kumari(22UG010135)

Sarmistha Parija

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

****

GIET UNIVERSITY, Gunupur

**2023 - 24**

****

**GIET UNIVERSITY,GUNUPUR**

***Dist. - Rayagada, Odisha-765022,* *Contact:- +91 7735745535, 06857-250170,172, Visit us:-* www.giet.edu**

**Department of Computer Science & Engineering**

**CERTIFICATE**



**This is to certify that the project work entitled “TIC TAC TOE” is done by Name- Sakshi Kumari Sinha(22UG010098), Ekata Kumari(22UG010135), Sarmistha Parija (22UG010, in partial fulfillment of the requirements for the 3rd Semester Sessional Examination of Bachelor of Technology in Computer Science and Engineering during the academic year 2023-24. This work is submitted to the department as a part of evaluation of 3rd Semester Learning Project-I.**

Miss Sandhya Rani Das Mr. Santosh Kumar Panda

Proctor Class Teacher

Mr. Bhavani Sankar Panda Dr. Preamanshu Shekhar Rath

And Dr. Sachikanta Dash

Project Coordinator, 2nd Year HoD, CSE, 2nd Year

**ACKNOWLEDGEMENT**

We express our sincere gratitude to Miss Sandhya Rani Das of Computer science and engineering for giving us an opportunity to accomplish the project. Without his/her active support and guidance, this project report has not been successfully completed.

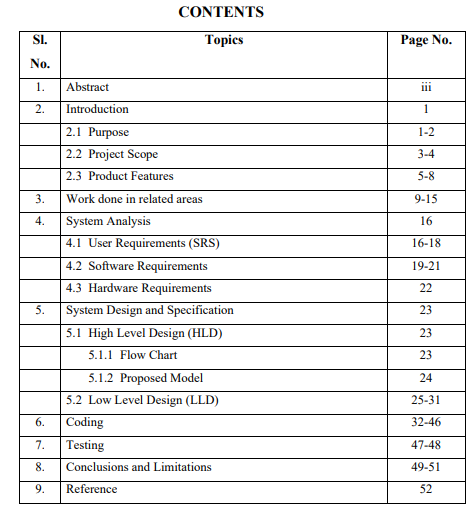
We also thank our class teacher Mr. Santosh Kumar Panda for his/her constant support and guidance provided during this project period.

We also thank Mr. Bhavani Sankar Panda, Project Coordinator, Dr. Preamanshu Shekhar Rath and Dr. Sachikanta Dash, Head of the Department of Computer Science and Engineering, Prof. (Dr.) Sanjay Kumar Kuanar, Dy. Dean, Computational Science, SOET for their consistent support, guidance and help.

Sakshi Kumari Sinha (22UG01098)

Ekata Kumari (22UG010135)

Sarmistha Parija



**PROJECT ABSTRACT**

The TIC TAC TOE presented here is a very simple application developed HTML, CSS, PHP and MYSQL. This portal fulfills the thirst of gaming which can be deliver for the player at any where anytime and any age through a range of gaming solution while compared with traditional gaming system. We have also used MYSQL database. The Tic Tac Toe project implemented in C language for a website offers a classic yet engaging gaming experience. The abstract begins by highlighting the enduring popularity of Tic Tac Toe, known for its simplicity and strategic depth. The project's primary objective is to bring this timeless game to a digital platform through web development, utilizing the versatile capabilities of the C programming language. The website seamlessly integrates user-friendly interfaces, allowing players to enjoy the game intuitively. To enhance the gaming experience, the project incorporates features such as responsive design for various devices and a visually appealing user interface. The website leverages HTML, CSS alongside C, fostering a holistic understanding of web development principles. The project explores the incorporation of duo-player functionality, allowing users to engage with friends in real-time.

Tic Tac Toe, a deceptively simple but highly engaging game, traces its origins to ancient civilizations, with varieties played in different cultures under names such as ropes, crosses, and more. The fundamental premise remains un-changed—a battle of wits on a 3x3 grid where two opponents each carrying a unique symbol wield weapons against each other in a devastating fashion; To achieve the success they crave.

The play begins with a blank canvas, a blank slate meant for the conflict of Xs and Os to unfold. Each player takes turns placing their marker in an empty space, with the goal of creating their chosen marker in a continuous row, column, or line The beauty of tic tac toe depends on its simplicity on; Nine classes, two symbols, and options.

Tic Tac Toe's subtle depth is often reduced. What appears on the surface when children are playing hides a kind of clever gimmick. Players must anticipate opponents’ movements, evaluate possible combinations of victories, and master the art of blocking. Each power is a calculated step, a delicate dance of offense and defense. The space limitations of the 3x3 grid intensify the competition, as each action significantly affects the dynamics of the board.

The gameplay in short is a double-edged sword, inviting players to engage in rapid-fire shootouts, and any tactical victory in a microworld is twofold—be it symbols winning matches or reluctantly accepting to score. But within these two outcomes lies more psychological stimulation, a battle of neural connections where synapses light up with each activation.

In addition to intellectual stimulation, Tic Tac Toe is social perfection. It transcends barriers of age, language and culture and is a universal language of play. Its accessibility in classrooms, cafes and digital forums is its mainstay, fostering friendly competition and camaraderie. The game’s enduring popularity is a testament to its timeless appeal and the basic human nature of strategic engagement.

An ancient board game based on Indian mythology, Snake and Ladder weaves a story of chance, strategy, and the unexpected roller coasters of life The game unfolds on a numbered grid, with multiple no it is 10x10, decorated with snake wings and staircase shortcuts. Players embark on a journey indicated by the roll of the dice, traversing the board with the ultimate goal of leading their opponents to reach the Final Four.

The two-tier snakes and ladders inject an element of dynamic unpredictability into the game. Stairs act as unintentional allies, pushing players forward with bubbles, snakes with social barriers sending them backwards and the interaction of these objects creates a story of highs and lows describing the ups and downs of life’s journey.

Conclusion Tic Tac Toe offer beautiful tapestries of association despite their simplicity and clarity. Tic Tac Toe challenges intelligence in a tense battle of strategy, it share a timeless ways of socializing, a way of thinking and a universal language of sport.

In conclusion, the Tic Tac Toe project in C language for a website serves as a comprehensive learning experience, combining classic game development with modern web technologies, reinforcing programming fundamentals, and providing an interactive platform for users to enjoy this ageless game in a digital context.

**CHAPTER-1 INTRODUCTION**

* 1. **PURPOSE**

A Tic Tac Toe website developed in the C language serves as a practical and educational tool, showcasing fundamental programming concepts and web development skills. This project provides an interactive platform for users to engage in the classic Tic Tac Toe game, fostering entertainment and strategic thinking.

The choice of Tic Tac Toe, a game that requires critical thinking within the constraints of a 3x3 grid, Snake and Ladder, a game of chance and strategy, pays homage to aspects of the traditional game available.

In addition to entertainment, the project is an educational platform. For Tic Tac Toe, it’s an opportunity to provide educational and strategic guidance to help players develop their complex ideas. Access to best practices, understanding systems, and developing decision-making capacity are inherent features of the educational scale.

A key principle of the project is inclusion. By creating a user-friendly interface, ensuring cross-platform compatibility, games can be accessed by a wider audience. Whether played on a desktop, laptop, tablet, or smartphone, the seamless experience caters to different player tastes and devices.

The project is also designed to be professionally inclusive. While the classics are inherently simple, the digital revolution allows for the inclusion of content for novice and experienced players and helps create an inclusive gaming environment with a variety of approaches so are preferences, educational resources, and separate problems.

Ultimately, the ultimate goal is to expand the enjoyment of the game. The project aims to inspire the fun, competition and camaraderie that traditional board games have brought for centuries. The digital realm is a canvas where the simplicity of tic tac toe and the dynamic story of snakes and ladders combine to create an exciting time for players around the world.

The use of web technologies in development plays an important role in the design of the project. HTML, CSS, and JavaScript form the foundation, resulting in a functional and beautiful user interface. This technology is used to ensure cross-browser compatibility, allowing users to access games on different devices without compromising functionality or aesthetics.

The user interface design has been carefully designed to enhance the overall gaming experience. Simple layouts, vibrant graphics and smooth transitions contribute to a visually appealing environment. The design prioritizes simplicity, ensuring that users can easily navigate through the game’s menus and have a hassle-free gaming experience.

For Tic Tac Toe, game technology is used to mimic traditional pen-and-paper ones. Players can play one-on-one with friends or AI opponents. Web technology facilitates real-time innovation, ensuring that every move is immediately reflected on the opponent’s screen, creating a more in-game feel

The Tic Tac Toe website in C serves as a tangible portfolio piece, showcasing the developer's ability to create functional and visually appealing web applications. It encourages problem-solving and critical thinking while fostering a deeper appreciation for the synergy between C programming and web development. Overall, this project not only entertains users with a timeless game but also equips developers with a well-rounded skill set, making it a purposeful endeavor in the realm of programming education.

* 1. **PROJECT SCOPE**

The scope of a tic tac toe website using HTML, CSS, C, PHP and MySQL is quite broad. With this combination of technologies, it is possible to create a fully-featured tic tac toe website that can support a gaming activity and develops critical.

The scopes are:

1. Educational tool
2. Community engagement
3. **Adaptation for Mobile Platforms**

At its core, the project aims to provide a simple and enjoyable gaming experience that can be accessed through a web browser. The inclusion of timeless games like Tic Tac Toe and Snake and Ladder adds a nostalgic touch, appealing to a wide range of audiences, from casual gamers to those who want to take a trip down memory lane.

Overall, the Tic Tac Toe website can be a collaborative project, encouraging teamwork and code sharing among developers. It opens avenues for code reviews, discussions, and improvements, creating a collaborative learning environment. Additionally, the website's adaptability allows developers to integrate new features or modify existing ones, making it a dynamic project with continuous scope for innovation and enhancement. Overall, the Tic Tac Toe website developed with C offers a multifaceted scope, bridging the realms of education, practical application, and collaborative programming.

**1.3PRODUCT FEATURES**

A tic tac toe website built using HTML, CSS, PHP and MySQL can have various Features. Tic Tac Toe, the web-based gaming service boasts a range of features designed to enhance the user experience and increase the overall enjoyment of the games

Here are some common features that such a portal might have:

1. **User Authentication:** Allow users to create accounts and log in, providing a personalized experience and the ability to track their game statistics.
2. **Responsive Design:** The project prioritized a functional and user-friendly interface. The design seamlessly adapts to different screen sizes and resolutions, ensuring that players can enjoy the game on a variety of devices including desktop, laptop, tablet, smartphone etc. The layout is intuitive, with easy navigable menus, visually appealing graphics and immersive games It helps to navigate the environment.
3. **Interactive Game Board:** Create a visually appealing and interactive game board with clickable cells, providing a seamless gaming experience.
4. **Game Statistics:** Track and display statistics such as the number of games played, wins, losses, and ties for each player.

**CHAPTER-2 WORKS DONE IN RELATED AREA**

1. Project Planning and Configuration:

The basis of the program is to create a structured file system. HTML files define the layout of web pages, CSS files handle the design, C contains the game and MYSQL contain the user data who has registered in the website. Typical layouts contain directories for images, style sheets. The functional design ensures modularity and maintainability.

2. HTML for layout:

In an HTML file you define the layout of web pages. Tic Tac Toe includes a grid that represents the game board in HTML. Each cell on the grid has a clickable element that players can place their icons on.

3. CSS for styling:

CSS is used to design HTML elements, creating an aesthetically pleasing interface. Design elements include describing the shape of the icons, buttons, and many. The CSS ensures responsiveness, making the website appealing on different devices.

4. C for Game Logic:

Cis the heart of the project, managing game logic, constructing table, asking the index of grid where the user wants to print his or her mark. For Tic Tac Toe, C monitors player turns, monitors victory status, and updates Winner and Looser or Draw status.

5. Event Management:

C captures user interaction through event processing. In Tic Tac Toe, listening events are used to determine in which Tic Tac Toe grid the mark is going to filled. This triggers activities to handle relocations and create new game environments to use duo-player functionality, the project requires two player and both enter their marks in desired place and out of them one is going to be the winner and other is looser, there also be a chance of draw between them. The seamless integration of this technology provides users with an engaging and enjoyable gaming experience.

**CHAPTER 3 SYSTEM ANALYSIS**

**3.1 Hardware Requirements**

Hardware requirements for web-based Tic Tac Toe game are minimal, as web development relies primarily on the computing power of the developer's machine for coding, testing, and debugging Here's a detailed breakdown:

1. Personal Computer:

Regular use of a personal computer or laptop is the basic requirement. It should have a modern processor with at least a dual core configuration and enough clock speed for a smooth development process. A multi-core processor can greatly improve performance in running different development tools simultaneously.

2. RAM (Random Access Memory):

At least 8 GB of RAM is recommended to ensure that the development environment, including web browsers, code editors, and other tools, can run smoothly. Having enough RAM to meet today’s web development workflow and testing requirements is important.

3. Storage:

The right kind of storage, preferably a solid-state drive (SSD), is beneficial for faster data throughput and improved overall system performance. Storage requirements depend on the project and other software tools installed on the device.

4. Internet connection:

A stable and reliable Internet connection is required to access online resources, libraries, and version control systems. It makes it easy to set up dependencies, updates, and collaboration with remote repositories.

5. Remote objects:

Standard peripherals such as keyboards, mice and displays are required for coding and testing. Also, having a second monitor can improve productivity by providing more screen real estate for code and preview panels.

6. Browser Development Tools:

Modern web browsers (such as Chrome, Firefox, Safari, etc.) come with built-in developer tools that are essential for developing and testing web applications. Making sure these browsers are up to date is important for a seamless development experience.

The hardware requirements for web-based games like Tic Tac Toe and Snake and Ladder are very minimal, focusing on standard personal computers with adequate processing power, RAM, and resources preservation Strong Internet connectivity and standard external infrastructure Comfortable and productive development environment supports.

**3.2 Software Requirements**

|  |
| --- |
| 1.Code Editor:  A code editor is necessary for writing and maintaining HTML, CSS, and JavaScript code. Popular features include Visual Studio Code, Sublime Text, Atom, or any other code editor of choice with features like syntax highlighting, code completion, and version control integration  2. Website:  Multiple web browsers are needed to test the compatibility of web-based games. Developers typically use Google Chrome, Mozilla Firefox, Safari, Microsoft Edge to ensure cross-browser compatibility and use their developer tools for debugging  3. Translation User System:  It’s important to use a version control system like Git to track changes, collaborate with team members, and manage different versions of a project. Platforms such as GitHub or GitLab can be used for remote repositories.  4. Web server:  For local development, a web server needs to be tested for web-based games. This can be accomplished using Node.js using tools like Express.js to create a simple server or other methods like Python’s SimpleHTTPServer.8. Text Editor (Optional):  You may need a text editor separate from the code editor to annotate the project, write README files, or handle other textual content. Software like Notepad++, Sublime Text, or Visual Studio Code can serve this purpose.  5. Browser Developer Tools:  Knowing and using browser developer tools is essential for debugging, profiling, and managing web pages as they are developed. Each major browser offers its own set of developer tools, including elements, consoles, networks, and more.  6. Test Setup (Optional):  Depending on the complexity of the project, adding a testing framework like Zest, Mocha, or Jasmine can help ensure code quality through automated testing.  In conclusion, the software requirements for web-based Tic Tac Toe game include code editor, web browser, version control system, web server, images and MYSQL database. Management software, and tools for documentation and testing. All these tools together provide the perfect environment for efficient and effective web development.  Essentially this software requirement provides a solid foundation for the development, testing and maintenance of the web-based Tic Tac Toe and Snake and Ladder games The combination of these tools ensures a smoother development process and a better end result.  **CHAPTER 4 SYSTEM DESIGN AND SPECIFICATIONS**  System design and specification for Tic Tac Toe website.  I. Introduction  The "Game-Zone" website aims to provide users with a platform to enjoy timeless games: Tic Tac Toe. The website will be user-friendly, attractive and accessible, catering to a diverse audience.  II. planning process  User Interface (UI):  The UI will have a clean and simple layout, with a homepage with options for Tic Tac Toe  Each game will have a dedicated interface, keeping the design consistent and maintaining a consistent user experience.  Game boards:  Tic Tac Toe: 3x3 grid of responsive cells for users to move themselves.  Considerations of Responsibility:  Make sure the website is accessible on different devices such as desktops, tablets and smartphones.  Server-side logic**:**  Server-side logic must be developed to manage game sessions, player movements, and game state.  Consider factors such as skill levels and availability and use a coherent system for multiplayer games.  Database Usage:  Store user information, play statistics, and preferences in a secure database.  Keep records of game completions and progress to enhance the overall user experience.  Security measures:  Use encryption protocols to protect user data and prevent unauthorized access.  Regularly update security measures to protect against potential vulnerabilities.  III. Game mechanics  Playing games:  Users can play against computers or other players.  Use algorithms to run the computer, making sure they vary in complexity.  Wins and accomplishments:  Track wins, losses and ties.  Install achievements and badges to incentivize players.  IV. User accounts and profiles  Creating Accounts:  Users can create accounts using email and phone number.  Use email verification for added security.  V. Economic strategies  Advertising revenue:  Combine non-intrusive ads to make money.  Offer a premium version without ads for a subscription fee.  In-app purchases:  Apply cosmetic upgrades to game pieces, backgrounds, and themes.  Introduce virtual currencies for in-game purchases.  VI. testing and quality assurance  Functional testing:  Thoroughly test out the game's functionality, including single and multiplayer modes.  Identify and correct errors to ensure a smooth user experience.  Performance Testing:  Optimize website performance to handle multiple concurrent users.  Test the site on different devices and browsers to ensure compatibility.  VII. Agriculture of the future  Other games:  Plan the integration of new games to keep the platform fresh and engaging.  Local Features:  Conversational features have been implemented to allow players to interact while playing.  Introduce events and community councils to create a sense of community.  The "Game-Zone" website, which is a combination of Tic Tac Toe, is designed to provide users with an engaging, interactive gaming experience. Focusing on intuitive design, robust backend design, and engaging game mechanics, the site aims to be a destination for classic game enthusiasts and regular updates and the future improvements will ensure the platform’s longevity and relevance in the dynamic world of online gaming.  TIC-TAC-TOE:  How to play one:  In single player mode, users have the option to play against computer opponents. The movements of the computer are determined by an algorithm tailored to the player’s skill level, creating a challenging  and enjoyable experience. Players can test their tactical skills against an opponent with artificial intelligence, increasing the replay value of the game.  Multiplayer mode:  The multiplayer mode allows users to challenge friends or random opponents in real time. The system ensures fast movement and immediately reflects the opponent’s move, creating a sense of competition and communication. The social aspect of multiplayer games adds some fun as players try to outdo each other.  Wins and accomplishments:  Wins, losses and relationships are tracked to provide a comprehensive record of player performance. Additionally, the introduction of achievements and badges adds some motivation, encouraging players to explore different paths to reach specific milestones This point system enhances the overall gameplay experience and gives players a sense of accomplishment work.  Conclusion, the detailed descriptions of Tic Tac Toe demonstrate the thoughtfulness of creating fun and engaging gaming experiences for users. Whether playing tactically on tic tac toe or embracing the elusiveness of snakes and ladders, the games are designed to be multiplayer, available to casual and competitive players fun in the "Game-Zone" web page.  **4.1 High Level Design (HLD)**  **4.1.1 Flow Chart**    **4.2.2 ER Diagram**    **4.2 Low Level Design (LLD)**  4.2.1 Process Specification  Algorithm: Tic Tac Toe Game  1. Initialize the board:  - Create a 3x3 grid (matrix) to represent the Tic Tac Toe board.  - Initialize each cell of the grid to be empty.  2. Display the empty board.  3. Loop until the game is over:  a. Prompt the current player for their move (row and column).  b. Check if the chosen cell is empty:  - If yes, mark the cell with the symbol of the current player.  - If no, inform the player that the cell is already occupied, and return to step 3a.  c. Display the updated board.  d. Check for a win:  - Check rows, columns, and diagonals for three symbols in a row for the current player.  - If found, declare the current player as the winner and end the game.  e. Check for a draw:  - If all cells are filled and no winner is found, declare the game as a draw and end the game.  f. Switch to the next player.  4. End the game.  Pseudocode:  initializeBoard():  // Create a 3x3 matrix and initialize each cell to be empty.  board = [[' ', ' ', ' '], [' ', ' ', ' '], [' ', ' ', ' ']]  return board  displayBoard(board):  // Display the current state of the Tic Tac Toe board.  for row in board:  print(row)  getPlayerMove():  // Prompt the current player for their move.  inputRow = getValidInput("Enter row (0, 1, or 2): ")  inputColumn = getValidInput("Enter column (0, 1, or 2): ")  return inputRow, inputColumn  getValidInput(prompt):  // Helper function to get valid input from the user.  while True:  userInput = input(prompt)  if userInput is a valid input:  return userInput  else:  print("Invalid input. Please try again.")  checkWin(board, player):  // Check if the current player has won.  // Check rows, columns, and diagonals.  // Return true if a win is found, false otherwise.  checkDraw(board):  // Check if the game is a draw.  // Return true if the board is full and no winner is found, false otherwise.  main():  // Main program logic.  board = initializeBoard()  currentPlayer = 'X'  while True:  displayBoard(board)  row, column = getPlayerMove()  if board[row][column] is empty:  board[row][column] = currentPlayer  if checkWin(board, currentPlayer):  displayBoard(board)  print("Player " + currentPlayer + " wins!")  break  if checkDraw(board):  displayBoard(board)  print("The game is a draw!")  break  currentPlayer = switchPlayer(currentPlayer)  else:  print("Cell already occupied. Try again.")  // End of the game.  **3.2.2 Screen-Shot Diagram**  Home Page      **Registration Page**    **Login Page**    **Explore/Help**    **Game code:**      **CHAPTER 7 CODING**  In coding we have used c language for game and HTML and CSS for the website along with PHP and MYSQL for database.  Let me show you all the section, as follows;  **C CODE:**  #include <stdio.h>  #include <conio.h>  void printBoard();  int checkWin();  void system();  char board[]={'0','1','2','3','4','5','6','7','8','9'};  void main(){  int player=1,input,status=-1;  printBoard();    while (status==-1)  {  player=(player%2==0) ? 2 : 1;  char mark=(player==1) ? 'X' :'O';  printf("\n\nPlease enter Number For Player %d\n",player);  scanf("%d",&input);  if(input<1 || input>9)  {  printf("invalid input");  }  board[input]=mark;  printBoard();  int result=checkWin();  if(result==1){  printf("\nPlayer %d is the Winner",player);  return;  }else if(result==0){  printf("\ndraw");  return;  }  player++;  }      }  void printBoard(){  system("cls");  printf("\n\n");  printf("\t\t\t\t\*\*\*=== TIC TAC TOE ===\*\*\*\n\n");  /\*printf(" | | \n");  printf(" %c | %c | %c \n",board[1],board[2],board[3]);  printf("\_\_|\_|\_\_\n");  printf(" | | \n");  printf(" %c | %c | %c \n",board[4],board[5],board[6]);  printf("\_\_|\_|\_\_\n");  printf(" | | \n");  printf(" %c | %c | %c \n",board[7],board[8],board[9]);  printf(" | | \n");  printf("\n\n");\*/  printf("\n\t\t\t\t | | ");  printf("\n\t\t\t\t %c | %c | %c ",board[1],board[2],board[3]);  printf("\n\t\t\t\t\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_");  printf("\n\t\t\t\t | | ");  printf("\n\t\t\t\t %c | %c | %c ",board[4],board[5],board[6]);  printf("\n\t\t\t\t\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_");  printf("\n\t\t\t\t | | ");  printf("\n\t\t\t\t %c | %c | %c ",board[7],board[8],board[9]);  printf("\n\t\t\t\t | | ");  }  int checkWin(){  if(board[1]==board[2] && board[2]==board[3]){  return 1;  }  if(board[1]==board[4] && board[4]==board[7]){  return 1;  }  if(board[7]==board[8] && board[8]==board[9]){  return 1;  }  if(board[3]==board[6] && board[6]==board[9]){  return 1;  }  if(board[1]==board[5] && board[5]==board[9]){  return 1;  }  if(board[3]==board[5] && board[5]==board[7]){  return 1;  }  if(board[2]==board[5] && board[5]==board[8]){  return 1;  }  if(board[4]==board[5] && board[5]==board[6]){  return 1;  }  int i,count=0;  for (i = 1; i <=9; i++)  {  if(board[i]=='X' || board[i]=='O'){  count++;  }  }    if(count==9){  return 0;  }  return -1;  }  **CSS CODE:**  \*{      padding: 0;      margin: 0;      box-sizing: border-box;  }  /\* -- Header section -- \*/  header{      width: 100%;      height: 100vh;      background: linear-gradient(rgba(0,0,0,0.8), rgba(0, 0, 0, 0.2)), url("assets/back1.jpeg");      background-size: cover;      font-family: 'Lucida Sans', 'Lucida Sans Regular', 'Lucida Grande', 'Lucida Sans Unicode', Geneva, Verdana, sans-serif;  }  nav{      width: 100%;      height: 100px;      color: black;      display: flex;      justify-content: space-between;      align-items: center;      padding: 28px 53px;  }  .logo{      font-size: 2em;      letter-spacing: 2px;      color: white;  }  .menu a{      text-decoration: none;      color: white;      padding: 10px 20px;      font-size: 20px;      position: relative;  }  .register a{      text-decoration: none;      color: white;      padding: 10px 20px;      font-size: 20px;      background: rgba(60, 60, 255);      border-radius: 8px;  }  .login a{      text-decoration: none;      color: white;      padding: 10px 20px;      font-size: 20px;      background: rgba(60, 60, 255);      border-radius: 8px;  }  .h-text{      max-width: 650px;      position: absolute;      top: 50%;      left: 50%;      transform: translate(-50%, -50%);      text-align: center;      color: white;  }  .h-text span{      letter-spacing: 5px;  }  .h-text h1{      font-size: 3.5em;  }  .h-text a{      text-decoration: none;      background: rgba(60, 60, 255);      color: white;      padding: 10px 20px;      letter-spacing: 5px;      transition: 0.4s;  }  /\* -- Hover Effect -- \*/  .menu a::before{      content: "";      position: absolute;      top: 0;      left: 0;      width: 0%;      height: 100%;      border-bottom: 2px solid indianred;      transition: 0.4s linear;  }  .menu a:hover::before{      width: 90%;  }  .register a:hover{      background: transparent;      border: 1px solid indianred;  }  .h-text a:hover{      background: transparent;      border: 1px solid indianred;  }  **PHP CODE:**  home page  <!DOCTYPE html>  <html lang="en">  <head>    <meta charset="UTF-8">    <meta name="viewport" content="width=device-width, initial-scale=1.0">    <link href='https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css' rel='stylesheet'>    <style>    .myaccount a{      text-decoration: none;      color: white;      padding: 10px 20px;      font-size: 20px;      background: red;      border-radius: 8px;  }  body{      #myVideo {      width: 100vw;      height: 100vh;      object-fit: cover;      position: fixed;      top: 0;      left: 0;      z-index: -1;}  }  /\* -- Header section -- \*/  header{      width: 100%;      height: 100vh;      //background: linear-gradient(rgba(0,0,0,0.8), rgba(0, 0, 0, 0.2)), url("assets/background.jpg");      background-size: cover;      font-family: 'Lucida Sans', 'Lucida Sans Regular', 'Lucida Grande', 'Lucida Sans Unicode', Geneva, Verdana, sans-serif;  }  nav{      width: 100%;      height: 100px;      color: black;      display: flex;      justify-content: space-between;      align-items: center;      padding: 28px 53px;  }  .logo{      font-size: 2em;      letter-spacing: 2px;      color: black;      font-family:cursive;  }  .menu a{      text-decoration: none;      color: red;      padding: 10px 20px;      font-size: 20px;      position: relative;  }  .lbtnr{      text-decoration: none;      color: white;      padding: 10px 20px;      font-size: 20px;      background: rgba(60, 60, 255);      border-radius: 8px;  }  .h-text{      max-width: 650px;      position: absolute;      top: 50%;      left: 50%;      transform: translate(-50%, -50%);      text-align: center;      color: black;  }  .h-text span{      letter-spacing: 3px;  }  .h-text h1{      font-size: 2.5em;  }  .h-text a{      text-decoration: none;      background: red ;      color:black;      padding: 10px 20px;      letter-spacing: 5px;      transition: 0.4s;  }  /\* -- Hover Effect -- \*/  .menu a::before{      content: "";      position: absolute;      top: 0;      left: 0;      width: 0%;      height: 100%;      border-bottom: 2px solid indianred;      transition: 0.4s linear;  }  .menu a:hover::before{      width: 90%;  }  .h-text a:hover{      background: transparent;      border: 1px solid indianred;  }  .lbtn{ text-decoration: none;      color: white;      padding: 10px 20px;      font-size: 20px;      background: rgba(60, 60, 255);      border-radius: 8px;width: auto; padding: 10px 18px; background-color:blue;}  .txt{ text-decoration: none;      color: red;      padding: 10px 20px;      font-size: 20px;      background: rgba(60, 60, 255);      border-radius: 8px;width: auto; padding: 10px 18px; background-color:beige;}  /\* == FOoter == \*/  footer{      background: black;      color: white;      text-align: center;      padding: 20px 0;  }  .social-icons a{      display: inline-block;      width: 35px;      height: 35px;      font-size:1cm;      border-radius: 50%;      margin-right: 22px;      text-align: center;      line-height: 35px;      border: 5px solid white;      outline: 2px solid #7d2ae8;  }  .social-icons a:hover{      transform: translateY(-5px);  }  .social-icons a i:hover{      color: white;  }    </style>    <title>Game Menu</title>  </head>  <body>     <!-- == Header Section == -->   <header>    <nav>        <div class="logo">            GameZone        </div>        <div class="menu">            <a href="redirect.html">Home</a>            <a href="#">About Us</a>            <a href="1.html">Explore/Help</a>        </div>        <div class="register">          <button type="button" class="lbtnr" onclick="window.location.href='login\_ttt.php'" ><span></span>Log In</a>            <button type="button" class="lbtnr" onclick="window.location.href='registration\_ttt.php'"><span></span>Register</a>        </div>    </nav>    <section class="h-text">        <span><h1>Let's the game</h1></span>        <h1>Begin</h1>        <br>        <a href="F:\tic tac toe\Tic\_tac\_toe.exe">play Now</a>     </section>  </header>  <div class="social-icons">      <a href="#"><i class='bx bxl-instagram-alt'></i></a>      <a href="#"><i class='bx bxl-facebook-square' ></i></a>      <a href="#"><i class='bx bxs-message-rounded-dots' ></i></a>  </div>  <!--<section class="info">    <div class="feature-card">    <i class='bx bxl-instagram-alt'></i>    <i class='bx bxl-facebook-square' ></i>    <i class='bx bxs-message-rounded-dots' ></i>      </div>  </section>-->    <!-- -- Footer -- -->    <footer>      &copy; 2023 TIC\_TAC\_TOE. All rights reserved.  </footer>    <!--<div class="button-container">      <form>      <button type="submit"class="submit1" >New Game</button>      <button type="submit"class="submit2">Settings</button>      <button type="submit"class="submit3"formaction="1.html">Quit</button>    </div>  </form>-->    <video autoplay muted loop id="myVideo">        <source /src="4K\_15.mp4"/ src="assets/4K\_15.mp4" type="video/mp4">    </video>  </body>  </html>  Registration Page  <?php  require\_once('connection\_pg.php');//php predefind keyword which is used to embed PHP code from another file.  if (isset($\_POST['submit\_btn'])) {      $user\_name = isset($\_POST["user\_name"]) ? $\_POST["user\_name"] : '';      $email = isset($\_POST["email"]) ? $\_POST["email"] : '';      $cont\_no = isset($\_POST["cont\_no"]) ? $\_POST["cont\_no"] : '';      $password = isset($\_POST["password"]) ? $\_POST["password"] : '';          // Check if the email or contact number already exist      $emailCheck = "SELECT email FROM regi WHERE email = '$email'";      $contactCheck = "SELECT cont\_no FROM regi WHERE cont\_no = '$cont\_no'";        $emailResult = mysqli\_query($conn, $emailCheck);      $contactResult = mysqli\_query($conn, $contactCheck);      if (mysqli\_num\_rows($emailResult) > 0) {          echo "Email is already in use. Please choose a different email address.";      } elseif (mysqli\_num\_rows($contactResult) > 0) {          echo "Contact number is already in use. Please choose a different contact number.";      } else {          // Insert data into the database          $sql = "INSERT INTO regi (user\_name, email,  cont\_no ,password)              VALUES ('$user\_name', '$email',  '$cont\_no', '$password')";            if (mysqli\_query($conn, $sql)) {              // Redirect to home.php after successful data insertion              header("Location: login\_ttt.php");              exit;          } else {              echo "Error";          }      }  }  ?>  <html>      <head>          <style>             .bg-image {                          background-image: url("assets/back.jpg");                          filter: blur(8px);                          -webkit-filter: blur(8px);                          height: 100%;                          background-position: center;                          background-repeat: no-repeat;                           background-size: cover;                      }              .bg-text {                          background-color: rgb(0,0,0); /\* Fallback color \*/                           background-color: rgba(0,0,0, 0.4); /\* Black w/opacity/see-through \*/                          color: white;                          font-weight: bold;                           border: 3px solid #f1f1f1;                           position: absolute;                           top: 50%;                           left: 50%;                          transform: translate(-50%, -50%);                           z-index: 2;                          width: 80%;                          padding: 20px;                          text-align: center;                      }              .lbtn{width: auto; padding: 10px 18px; background-color:blue;}              .cancelbtn{width: auto; padding: 10px 18px; background-color:red;}          </style>      </head>      <body>          <form>              <div class="bg-image">                </div>              <div class="bg-text">             <h1><b>REGISTRATION FORM</b><br><br><br></h1>              <div class="container">                  <label for="textbox" id="name" >USERNAME</label><br>                      <input type="textbox" id="name" placeholder="Enter Username" class="uname"><br><br>                  <label for="text" id="email">EMAIL</label><br>                      <input type="text" id="email" placeholder="Email" class="email"><br><br>                  <label for="textbox" id="mobile">MOBILE NUMBER</label><br>                      <input type="textbox" id="mobile" placeholder="Enter Mobile Number" class="no"><br><br>                    <label for="password" id="password">PASSWORD</label><br>                      <input type="password" id="password" placeholder="Password" class="pass"><br><br>                  <label ><br>                      <input type="checkbox" checked="checked" name="AGREE"> <b>I accept all the terms and condition</b>                  </label><br><br><br>              <button type="button" class="lbtn" onclick="window.location.href='login\_ttt.php'">SUBMIT</button>              <button type="button" class="cancelbtn" onclick="window.location.href='redirect.php'">CANCEL</button><br>              </div>          </form>      </div>      </body>  </html>  Login Page  <?php  require\_once('connection\_pg.php');//php predefind keyword which is used to embed PHP code from another file.  if (isset($\_POST['submit\_btn'])) {        $email = isset($\_POST["email"]) ? $\_POST["email"] : '';      $password = isset($\_POST["password"]) ? $\_POST["password"] : '';          // Check if the email or contact number already exist      $emailCheck = "SELECT email FROM regi WHERE email = '$email'";        $emailResult = mysqli\_query($conn, $emailCheck);        if (mysqli\_num\_rows($emailResult) > 0) {          echo "Email is already in use. Please choose a different email address.";      }      else {          // Insert data into the database          $sql = "INSERT INTO regi (user\_name, email,  cont\_no ,password)              VALUES ('$user\_name', '$email',  '$cont\_no', '$password')";            if (mysqli\_query($conn, $sql)) {              // Redirect to home.php after successful data insertion              header("Location: redirect.php");              exit;          }          else           {              echo "Error";          }      }  }  ?>  <html>      <head>          <style>              h2{color: aqua;}              .container {padding: 16px;}              .cancelbtn {width: auto; padding: 10px 18px; background-color: #f44336;}              .lbtn{width: auto; padding: 10px 18px; background-color:blue;}              .email{width: auto; padding: 10px 18px; background-color:whitesmoke;}              .pass{width: auto; padding: 10px 18px; background-color:whitesmoke;}              .number{width: auto; padding: 10px 18px; background-color:whitesmoke;}              .bg-image {                          background-image: url("assets/empty.jpg");                          filter: blur(8px);                          -webkit-filter: blur(8px);                          height: 100%;                          background-position: center;                          background-repeat: no-repeat;                           background-size: cover;                      }              .bg-text {                          background-color: rgb(0,0,0); /\* Fallback color \*/                           background-color: rgba(0,0,0, 0.4); /\* Black w/opacity/see-through \*/                          color: white;                          font-weight: bold;                           border: 3px solid #f1f1f1;                           position: absolute;                           top: 50%;                           left: 50%;                          transform: translate(-50%, -50%);                           z-index: 2;                          width: 80%;                          padding: 20px;                          text-align: center;                      }          </style>      </head>      <body>            <form>              <div class="bg-image">                </div>              <div class="bg-text">              <h2><b>WELCOME</b><br></h2>              <div class="container">              <input type="email" id="email"  placeholder="Email" class="email"><br><br>              <input type="password" id="password" placeholder="Password" class="pass"><br><br>              <button type="button" class="lbtn" onclick="window.location.href='redirect.php'">LOGIN</button>              <button type="button" class="cancelbtn" onclick="window.location.href='redirect.php'">CANCEL</button><br>              <label>                  <input type="checkbox" checked="checked" name="remember"><b> Remember me</b>              </label><br><br>              <label>                  <button type="button" name="create" onclick="window.location.href='registration\_ttt.php'" > Create account              </label>              </div>          </form>      </div>      </body>  </html>  **Now comes the connection part**  **Connection page;**  <?php  $username = "root";  $password= "";  $server = 'localhost';  $db = 'tictac';  $port= 3306;  $conn = mysqli\_connect($server, $username, $password, $db, $port);  if($conn){      //echo "Connection Successfull";  }  else{      echo  "not connected";  }  ?>  **CHAPTER 5 TESTING**  **5.1 Unit Testing:**  Now we are going to test our code. So required data of Test cases(input Excepected,Output ,Actual output)  Now we are going to show our interface of the project before testing it the Actual output of the project as shown in the screen shot:  TIC-TAC-TOE:  Html: this is my Excepted output  <link rel="stylesheet" href="style.css">  </head>  <body>  <div class="wrapper">  <div class="container">  <button class="button-option"></button>  <button class="button-option"></button>  <button class="button-option"></button>  <button class="button-option"></button>  <button class="button-option"></button>  <button class="button-option"></button>  <button class="button-option"></button>  <button class="button-option"></button>  <button class="button-option"></button>  </div>  <button id="restart">Restart</button>  </div><div class="popup hide">  <p id="message">Sample Message</p>  <button id="new-game">New Game</button>  </div>  </body>  </html>  But actual output came when there is something changes made by us on button id and the of the game and some content the output of the code came is that  Actual output when tested:  <button class="button-option"></button>  <button class="button-option"></button>  <button class="button-option"></button>  </div><button id="restart"></button>  </div><div class="popup hide">  <p id="message">Sample Message</p>  <button id="new-game">Old Game</button>  </div>  CSS:  padding: 0;  margin: 0;  box-sizing: border-box;  font-family: "Raleway", sans-serif;  }body {  height: 100vh;  background: linear-gradient(135deg, #8052ec, #d161ff);  }  html {  font-size: 16px;  }  .wrapper {  position: absolute;  transform: translate(-50%, -50%);  top: 50%;  left: 50%;  }  **CHAPTER-6 CONCLUSION & LIMITATION**  **6.1 LIMITATION**  Although we have put our best efforts to make the website flexible, easy to operate but limitations cannot be ruled out even by us. it was not possible to make the software full proof and dynamic. Lack of time also compelled us to ignore some part. List of limitations which are present in Our tic tac toe website Project:  Privacy Concerns: Taking personal information for registration raise privacy concerns, especially in environments where there is sensitive or personal information. Balancing the need for security with user privacy is an ongoing challenge.  Interaction impact: Users looking for a competitive or cooperative gaming experience with friends or others online could offset the lack of multiplayer functionality.  Future Improvements: Introducing multiplayer features could make the website more appealing and provide a more dynamic and social gaming environment. Limited set options:  Description: The function provides users with limited options for customization. Players cannot change game settings such as board size, difficulty levels, or visual themes.  Impact: The lack of customization can create a less personalized experience for users who want to customize the game environment future.  Improvements: Implement customizable settings to meet the needs of a wider audience, allowing users to personalize the gaming experience  **6.2 CONCLUSION**  Tic tac toe website is a comprehensive gaming platform that provides a range of features for users or players. Tic tac toe website is a excellent example of how modern web technologies can be used to build robust and user-friendly gaming platforms. The Tic tac toe website is a comprehensive and innovative solution that addresses the need for accessible, affordable, and effective education. Tic tac toe website is a comprehensive gaming platform that provides a range of features for users or players.  Here are some features our website offers;  **User-Friendly Interface:** The website prioritizes a user-friendly interface, ensuring a seamless and enjoyable experience for players. The design is clean, intuitive, and responsive across various devices, promoting accessibility.  **Feature-rich Gameplay:** The Tic Tac Toe game boasts a range of features, including single-player and two-player modes, customizable game settings, and an interactive scoreboard. These elements enhance the overall gaming experience and cater to diverse user preferences.  **Responsive Design:** Recognizing the importance of accessibility, the website is designed to be responsive, adapting to different screen sizes and devices. This ensures that users can enjoy the game regardless of the platform they are using.  **Learning Opportunities:** Throughout the development process, I encountered challenges that provided valuable learning opportunities. Overcoming these hurdles enhanced my problem-solving skills and deepened my understanding of web development concepts.  **Community Engagement:** Incorporating user feedback and engaging with the online community played a crucial role in refining the website. Continuous communication with users helped identify areas for improvement and implement updates to enhance the overall experience.  **Future improvements:**  While the current iteration of the website provides a solid gaming experience, there is always room for improvement and expansion. Future improvements may include:  1. Multiplayer Function:  Add a multiplayer function that allows users to play against each other, locally or online, and increases the social competitive aspect of the game.  2. Options:  Give users the ability to customize game settings such as board size, difficulty levels, or themes, to create a personalized gaming experience  3.Leadership and Control:  Use point systems, leaderboards, and achievement badges to add competitive performance and encourage players to strive for continuous improvement.  4. Access Facilities:  Enhance accessibility by adding features such as voice commands, keyboard navigation and widely varying options to ensure inclusiveness for users with different needs.  Lessons to be Learned:  Creating this gaming website has been an invaluable learning experience. The main takeaways are:  1. Teamwork:  Collaboration is key to the success of the project. Effective communication and division of labor among team members contribute to ease and efficiency.  2. Problem Solving:  Practicing challenges, whether for game logic or debugging, has enhanced problem solving skills. The iterative process of testing, identifying problems and implementing solutions has been critical to the success of the project.  3.User considerations:  What matters most to the user experience is what matters most. Design options and functionality have been implemented with the end user in mind, ensuring an enjoyable and convenient gaming experience for a variety of audiences  **CHAPTER 7 Reference/Bibliography**   * Google (https://www.google.com/ ) * Youtube (<https://www.youtube.com/> )   Bottom of Form |