## Front End Engineering-II

Project Report

Semester-IV (Batch-2022)

PING PONG GAME

(Using Bootstrap)

A red and white sign

Description automatically generated with low confidence

**Supervised By: Submitted By:**

Mr. Raveesh Samkaria Yashleen Kaur

2210990981

G-14

**Department of Computer Science and Engineering**

## Chitkara University Institute of Engineering & Technology,

## Chitkara University, Punjab

**ABSTRACT**

This project presents a dynamic implementation of the classic Ping Pong game using HTML, CSS, JavaScript, and Bootstrap, with the utilization of the canvas element for rendering. The game is designed as a two-player experience, with each player controlling a paddle on opposite sides of the screen. Player one controls the left paddle using the W and S keys, while player two controls the right paddle using the up and down arrow keys.

The game features real-time score tracking for both players, displayed prominently above their respective paddles. As players successfully hit the ball past their opponent's paddle, their score increases accordingly. Conversely, when a player misses the ball, their opponent's score increases, creating a competitive atmosphere.

The game continues until one player reaches a predetermined maximum score limit. Upon reaching the maximum score limit, a toast notification is triggered, declaring the winner of the match. This notification provides a clear indication of the game outcome, adding to the overall user experience.

The project leverages Bootstrap for responsive design, ensuring optimal gameplay experiences across various screen sizes and devices. The integration of HTML, CSS, and JavaScript allows for smooth gameplay mechanics, including paddle movement, ball physics, and score tracking.

In conclusion, this Ping Pong game project demonstrates the fusion of classic gameplay with modern web technologies. Through its intuitive controls, dynamic gameplay, and responsive design, it offers players an engaging and entertaining experience reminiscent of the traditional Ping Pong game, while embracing the capabilities of web development for interactive entertainment.

Top of Form

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| S. No. | SECTION | PAGE NO. |
| 1. | Introduction | 4-5 |
| 2. | Problem statement | 6 |
| 3. | Technical details | 6-9 |
| 4. | Proposed Design and Methodology | 9-10 |
| 4. | Schematic Diagram | 11 |
| 5. | File Structure | 12 |
| 6. | Code | 12-18 |
| 7. | Results (Code Snippets) | 19-20 |
| 8. | LIST OF REFERENCES | 20 |

**Introduction**

The Ping Pong game has long been a beloved pastime, known for its simplicity, competitiveness, and endless entertainment. In today's digital age, recreating this classic game in a digital format opens up new avenues for enjoyment and engagement. This project endeavors to bring the timeless appeal of Ping Pong to the realm of web development, utilizing HTML, CSS, JavaScript, and Bootstrap to create an immersive and interactive gaming experience. By leveraging modern web technologies, we aim to capture the essence of the traditional Ping Pong game while offering players a fresh and dynamic approach to gameplay.

* **Background**

The concept of Ping Pong traces back to the early 20th century, evolving from table tennis and gaining popularity as a recreational sport worldwide. With the advent of video games, Ping Pong found its digital counterpart, captivating audiences on various gaming platforms. In recent years, web development frameworks like Bootstrap have revolutionized the creation of web-based games, enabling developers to design responsive and visually appealing experiences that cater to diverse audiences. This project builds upon this rich history, merging the nostalgia of Ping Pong with the innovation of web development to deliver an engaging and accessible gaming experience for players of all ages.

* **Objective**

The primary objective of this project is to develop a fully functional Ping Pong game using HTML, CSS, JavaScript, and Bootstrap. Specifically, the project aims to implement the following key features:

1. Responsive design: Create a visually appealing and user-friendly interface that adapts seamlessly to different screen sizes and devices.
2. Player controls: Enable players to control their paddles using keyboard inputs, providing a familiar and intuitive gameplay experience.
3. Real-time score tracking: Implement a scoring system that updates dynamically as players score points during gameplay.
4. Dynamic gameplay mechanics: Ensure smooth ball movement, realistic physics, and responsive paddle interactions to enhance the overall gameplay experience.
5. Toast notification: Display a toast notification to declare the winner when one player reaches the maximum score limit, providing clear feedback and closure to the game.

* **Significance**

This project holds significance on multiple levels. Firstly, it serves as an educational tool for learning and practicing web development skills, allowing developers to gain hands-on experience with HTML, CSS, JavaScript, and Bootstrap in a practical context. Additionally, the project contributes to the preservation and modernization of classic games, ensuring that timeless favorites like Ping Pong remain accessible and enjoyable in the digital age. Furthermore, the project has the potential to foster social interaction and friendly competition, as players engage in multiplayer matches and share their gaming experiences with others. Overall, the significance of this project lies in its ability to entertain, educate, and inspire, bridging the gap between tradition and innovation in the world of gaming.

**Problem Statement:**

Despite the enduring popularity of the Ping Pong game, there is a lack of accessible and engaging digital versions that capture the essence of the traditional gameplay experience. Existing digital implementations often suffer from limited interactivity, poor user interface design, and a lack of responsiveness across different devices. Furthermore, there is a need for a modern, web-based Ping Pong game that leverages contemporary web development technologies to deliver a seamless and immersive gaming experience.

The challenge lies in developing a Ping Pong game that addresses these shortcomings by utilizing HTML, CSS, JavaScript, and Bootstrap to create a visually appealing, responsive, and intuitive user interface. The game must feature dynamic gameplay mechanics, including realistic ball physics, responsive paddle controls, and real-time score tracking. Additionally, the game should support multiplayer functionality, allowing players to compete against each other in a competitive yet enjoyable manner.

**Technical Details:**

**1. Technologies Used:**

* **HTML (Hypertext Markup Language):** Used for creating the structure and layout of the game interface.
* **CSS (Cascading Style Sheets):** Utilized for styling the game elements, including colors, fonts, and layout design.
* **JavaScript:** Implemented for game logic, including ball movement, paddle controls, collision detection, and scoring.
* **Bootstrap:** Leveraged for responsive design and layout components, ensuring compatibility across various devices and screen sizes.
* **Canvas Element:** Employed for rendering the game graphics and animations, providing a dynamic and interactive visual experience.

**2. Game Mechanics:**

* **Player Controls:** Players control their paddles using keyboard inputs. Player one uses the 'W' and 'S' keys to move the left paddle up and down, respectively, while player two uses the up and down arrow keys to control the right paddle.
* **Ball Movement:** The ball moves dynamically across the game canvas, bouncing off the paddles and walls according to realistic physics.
* **Collision Detection:** Collision detection algorithms are implemented to detect collisions between the ball, paddles, and game boundaries, triggering appropriate reactions such as paddle deflection and scoring.
* **Score Tracking:** Real-time score tracking is implemented for both players, with scores displayed prominently above their respective paddles. Scores are incremented when a player successfully hits the ball past their opponent's paddle.
* **Winner Declaration:** Upon reaching a predetermined maximum score limit, a toast notification is displayed to declare the winner of the match, providing clear feedback and closure to the game.

**3. Responsive Design:**

* The game interface is designed to be responsive, ensuring optimal viewing and interaction experiences across various devices, including desktops, tablets, and smartphones.
* Bootstrap's grid system and responsive utilities are utilized to create a flexible and adaptable layout that adjusts seamlessly to different screen sizes and resolutions.

**4. Multiplayer Functionality:**

* The game supports multiplayer functionality, allowing two players to compete against each other in real-time matches.
* Each player controls their paddle independently, enabling simultaneous gameplay experiences for both players.

**5. Code Structure:**

* The project code is organized into separate HTML, CSS, and JavaScript files for clarity and maintainability.
* JavaScript functions are structured logically, with separate functions handling different aspects of the game logic, such as paddle movement, ball physics, collision detection, and score tracking.

**HTML Structure:**

* The HTML file sets up the basic structure of the webpage, including the canvas element for rendering the game and buttons for controlling the game (start, pause, restart).
* Bootstrap CSS is included for styling, ensuring a visually appealing layout that adapts to different screen sizes.

**JavaScript Logic:**

* JavaScript is responsible for handling game mechanics, including paddle movement, ball physics, collision detection, and scoring.
* Keyboard events are used to control the paddles, with separate functions for handling key press and release events.
* The game state is updated continuously within the game loop function, which orchestrates the update and rendering of game objects.

**Game Mechanics:**

* The game features a dynamic ball that bounces off the paddles and walls, with realistic physics implemented for movement and collisions.
* Paddle movement is controlled by keyboard inputs, allowing players to move their paddles up and down to hit the ball.
* Scores are tracked in real-time for both players, with the game ending and declaring a winner once one player reaches the maximum score limit.

**Responsive Design:**

* Bootstrap's responsive grid system ensures that the game interface adjusts smoothly to different screen sizes and devices, maintaining usability and accessibility.

**Modal for Winner Declaration:**

* A modal (toast notification) is displayed to declare the winner when one player reaches the maximum score limit, providing clear feedback to the players.

**Additional Considerations:**

* The code is well-commented and organized, enhancing readability and maintainability.
* jQuery is used for DOM manipulation, simplifying event handling and interaction with HTML elements.
* Overall, your project demonstrates a solid understanding of HTML, CSS, JavaScript, and Bootstrap, resulting in a functional and engaging Ping Pong game. Well done!

**6. Testing and Optimization:**

* Extensive testing is conducted to ensure the game functions as intended across different browsers and devices.
* Performance optimization techniques are implemented to enhance the game's responsiveness and smoothness, minimizing lag and improving overall user experience.

**7. Deployment:**

* The game is deployed on a web server or hosting platform to make it accessible to players worldwide.
* Deployment considerations include server configuration, domain setup (if applicable), and ensuring compatibility with various web browsers and devices.

**Proposed Design and Methodology:**

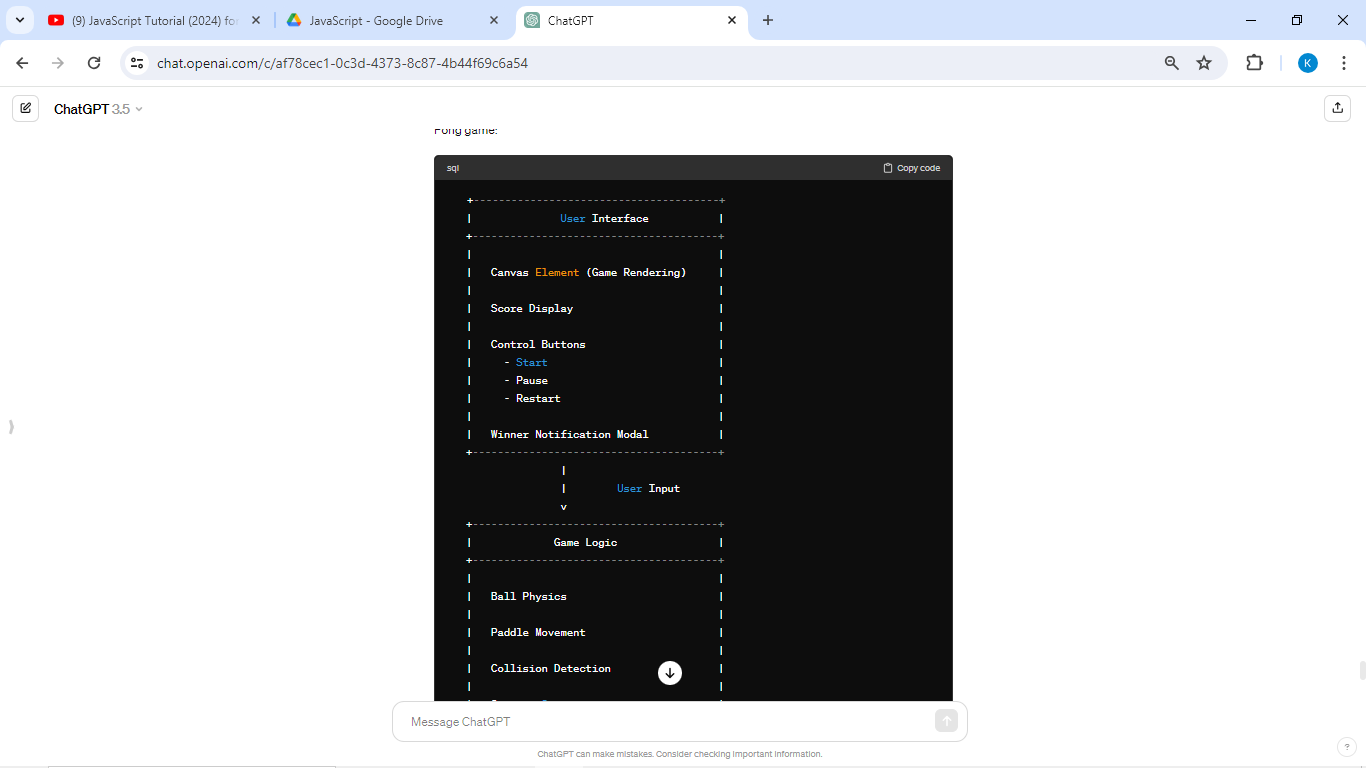
**Design Concept:** The design concept for the Ping Pong game revolves around creating an engaging and intuitive gaming experience that pays homage to the classic arcade-style gameplay while incorporating modern web development techniques. The game aims to strike a balance between nostalgia and innovation, offering players a familiar yet refreshing take on the traditional Ping Pong experience. With an emphasis on responsive design, dynamic gameplay mechanics, and multiplayer functionality, the Ping Pong game seeks to captivate players of all ages and skill levels, providing endless hours of entertainment and enjoyment.

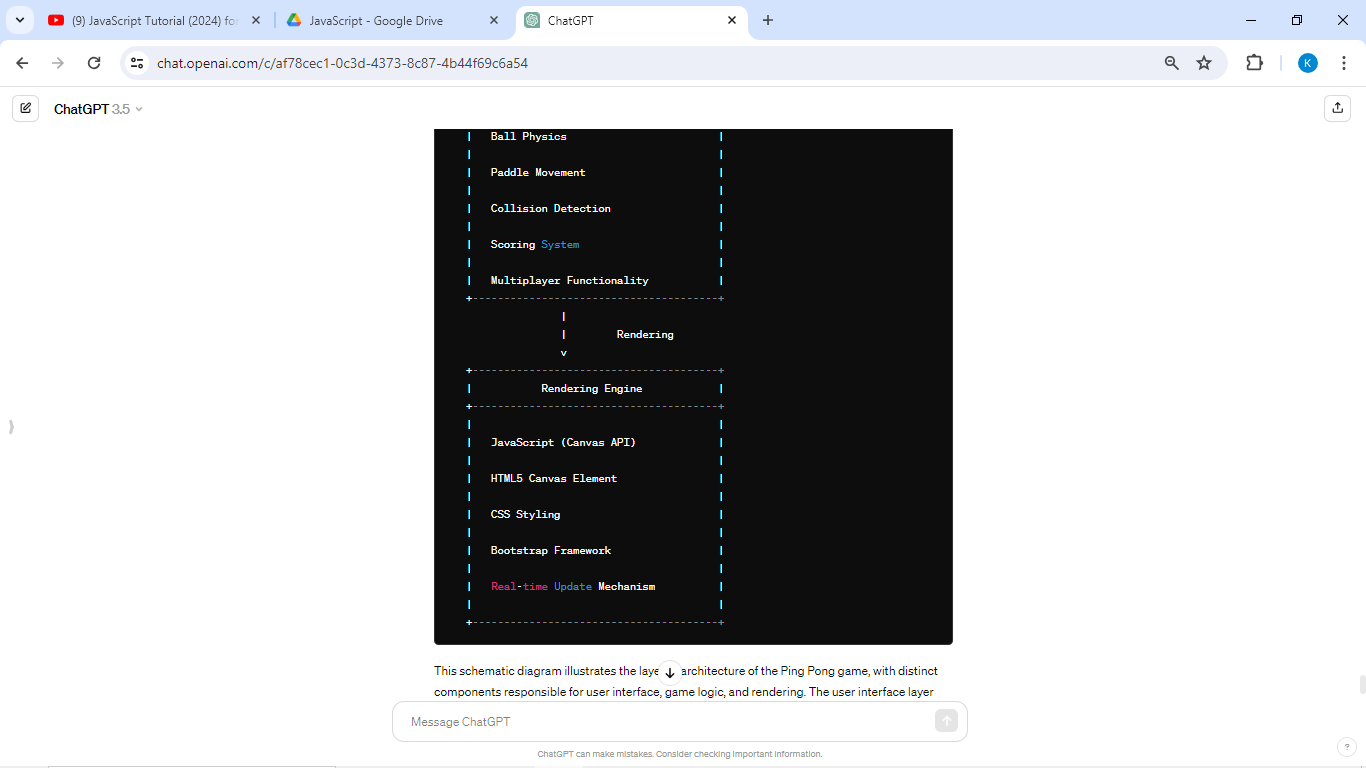
**User-Interface Design:**

1. **Game Layout:**
   * **Canvas Element:** The game's primary interface will consist of an HTML canvas element, strategically positioned within the webpage to showcase the game graphics and animations.
   * **Scoring Display:** Real-time scoring for both players will be prominently displayed above their respective paddles, ensuring players can track their progress and stay engaged throughout the game.
2. **Control Buttons:**
   * **Start Button:** A Start button will initiate the game and begin gameplay, allowing players to start a new match and enjoy the Ping Pong experience.
   * **Pause Button:** A Pause button will enable players to temporarily halt the game, providing a brief intermission without ending the current match.
   * **Restart Button:** A Restart button will reset the game state and allow players to start a new match from scratch, providing a convenient way to replay the game after completion.
3. **Keyboard Controls:**
   * **Player Left Controls (W and S):** Player one will control the left paddle using the W and S keys, enabling vertical movement up and down to intercept the ball.
   * **Player Right Controls (Arrow Keys):** Player two will control the right paddle using the up and down arrow keys, mirroring the functionality of the left paddle controls for intuitive gameplay.
4. **Modal Notification:**
   * **Winner Notification:** A modal notification will be displayed when one player reaches the maximum score limit, declaring the winner of the match and providing closure to the game session.
   * **Close Button:** The modal notification will include a Close button to allow players to dismiss the notification and return to the main game interface seamlessly.

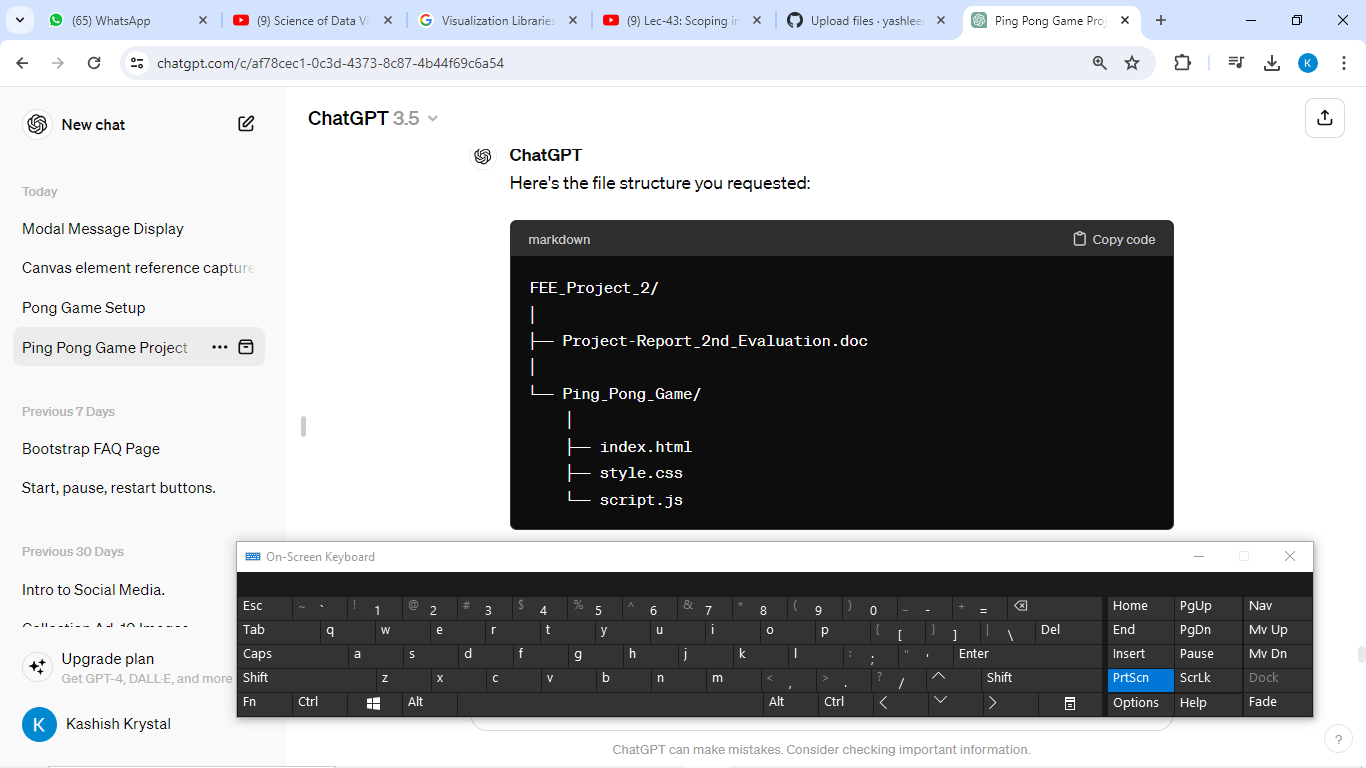
By implementing this user-friendly interface design and adhering to the proposed design concept, the Ping Pong game aims to deliver a visually appealing, responsive, and immersive gaming experience that resonates with players of all backgrounds and preferences. With intuitive controls, dynamic gameplay mechanics, and multiplayer functionality, the game promises to entertain and delight players, bringing the excitement of Ping Pong to the digital realm with style and sophistication.

**1.3 Schemantic Diagram:**





**File Structure:**

****

Top of Form

**HTML Code :**

<!DOCTYPE html>

<html>

<head>

  <title>Ping Pong Game</title>

  <!-- Add the Bootstrap CSS file -->

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">

 <!-- Linked the css file -->

<link rel="stylesheet" href="style.css">

</head>

<body>

  <div class="container"><br>

    <h1>Ping Pong Game </h1>

    <canvas id="canvas" class="bg-dark mt-4 custom-border" width="600" height="400"></canvas>

    <div class="row justify-content-center mt-4">

      <button id="start-btn" class="btn btn-primary mx-2">Start</button>

      <button id="pause-btn" class="btn btn-primary mx-2">Pause</button>

      <button id="restart-btn" class="btn btn-primary mx-2">Restart</button>

  </div>

</div>

 <p class="control">Control: Player Left(W and S) | Player Right(↑ and ↓)</p>

  <!-- Toast notification for Winner-->

<div class="modal" tabindex="-1" role="dialog" id="message-modal">

  <div class="modal-dialog" role="document">

    <div class="modal-content">

      <div class="modal-body">

        <h5 id="message"></h5>

      </div>

      <div class="modal-footer">

        <button type="button" id="message-modal-close" class="btn btn-secondary" data-dismiss="modal">Close</button>

      </div>

    </div>

  </div>

</div>

  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>

  <script src="script.js"></script>

</body>

</html>

**CSSCode :**

.container {

    display: flex;

    flex-direction: column;

    align-items: center;

    justify-content: center;

    background-color: #b0b3b0;

}

body{

  background-color: #b0b3b0;

}

  .button{

    display: flex;

    justify-content: center;

    margin-top: 15px;

  }

  .btn {

    background-color: #02b608;

    border: none;

    color: white;

    padding: 10px 20px;

    text-align: center;

    text-decoration: none;

    display: inline-block;

    font-size: 16px;

    margin: 10px;

    cursor: pointer;

    border-radius: 5px;

  }

  .btn:hover {

    background-color: #da1d0f;

  }

  .control {

      text-align: center;

  }

  .custom-border {

    border: 2 px solid red; /\* Change the border width and color as needed \*/

}

**1.5JavaScriptCode :**    description:

// canvas

var canvas = document.getElementById("canvas");

var ctx = canvas.getContext("2d");

var startBtn = document.getElementById("start-btn");

var pauseBtn = document.getElementById("pause-btn");

var restartBtn = document.getElementById("restart-btn");

var animationId;

var gameRunning = false;

startBtn.addEventListener("click", function() {

  if (!gameRunning) { // only start the game if gameRunning is false

    gameRunning = true; // set gameRunning to true when the game starts

    loop();

  }

});

pauseBtn.addEventListener("click", function() {

  gameRunning = false;

  cancelAnimationFrame(animationId);

});

restartBtn.addEventListener("click", function() {

  document.location.reload();

});

addEventListener("load", (event) => {

  draw();

});

// Define ball properties

var ballRadius = 10;

var ballX = canvas.width / 2;

var ballY = canvas.height / 2;

var ballSpeedX = 5;

var ballSpeedY = 5;

// Define paddle properties

var paddleHeight = 80;

var paddleWidth = 10;

var leftPaddleY = canvas.height / 2 - paddleHeight / 2;

var rightPaddleY = canvas.height / 2 - paddleHeight / 2;

var paddleSpeed = 10;

// Define score properties

var leftPlayerScore = 0;

var rightPlayerScore = 0;

var maxScore = 3;

// Listen for keyboard events

document.addEventListener("keydown", keyDownHandler);

document.addEventListener("keyup", keyUpHandler);

// Handle key press

var upPressed = false;

var downPressed = false;

let wPressed = false;

let sPressed = false;

function keyDownHandler(e) {

  if (e.key === "ArrowUp") {

    upPressed = true;

  } else if (e.key === "ArrowDown") {

    downPressed = true;

  } else if (e.key === "w") {

    wPressed = true;

  } else if (e.key === "s") {

    sPressed = true;

  }

}

// Handle key release

function keyUpHandler(e) {

  if (e.key === "ArrowUp") {

    upPressed = false;

  } else if (e.key === "ArrowDown") {

    downPressed = false;

  } else if (e.key === "w") {

    wPressed = false;

  } else if (e.key === "s") {

    sPressed = false;

  }

}

// Update game state

function update() {

  // Move paddles

  if (upPressed && rightPaddleY > 0) {

    rightPaddleY -= paddleSpeed;

  } else if (downPressed && rightPaddleY + paddleHeight < canvas.height) {

    rightPaddleY += paddleSpeed;

  }

  // Move right paddle based on "w" and "s" keys

  if (wPressed && leftPaddleY > 0) {

    leftPaddleY -= paddleSpeed;

  } else if (sPressed && leftPaddleY + paddleHeight < canvas.height) {

    leftPaddleY += paddleSpeed;

  }

  // Move ball

  ballX += ballSpeedX;

  ballY += ballSpeedY;

  // Check if ball collides with top or bottom of canvas

  if (ballY - ballRadius < 0 || ballY + ballRadius > canvas.height) {

    ballSpeedY = -ballSpeedY;

  }

  // Check if ball collides with left paddle

  if (

    ballX - ballRadius < paddleWidth &&

    ballY > leftPaddleY &&

    ballY < leftPaddleY + paddleHeight

  ) {

    ballSpeedX = -ballSpeedX;

  }

  // Check if ball collides with right paddle

  if (

    ballX + ballRadius > canvas.width - paddleWidth &&

    ballY > rightPaddleY &&

    ballY < rightPaddleY + paddleHeight

  ) {

    ballSpeedX = -ballSpeedX;

  }

  // Check if ball goes out of bounds on left or right side of canvas

  if (ballX < 0) {

    rightPlayerScore++;

    reset();

  } else if (ballX > canvas.width) {

    leftPlayerScore++;

    reset();

  }

  // Check if a player has won

  if (leftPlayerScore === maxScore) {

    playerWin("Left player");

  } else if (rightPlayerScore === maxScore) {

    playerWin("Right player");

  }

}

function playerWin(player) {

  var message = "Congratulations! " + player + " win!";

  $('#message').text(message); // Set the message text

  $('#message-modal').modal('show'); // Display the message modal

  reset();

}

// Reset ball to center of screen

function reset() {

  ballX = canvas.width / 2;

  ballY = canvas.height / 2;

  ballSpeedX = -ballSpeedX;

  ballSpeedY = Math.random() \* 10 - 5;

}

// Draw objects on canvas

function draw() {

  // Clear canvas

  ctx.clearRect(0, 0, canvas.width, canvas.height);

  ctx.fillStyle = "#FFF";

  ctx.font = "15px Arial";

  ctx.beginPath();

  ctx.moveTo(canvas.width / 2, 0);

  ctx.lineTo(canvas.width / 2, canvas.height);

  ctx.strokeStyle = "#FFF"; // Set line color to white

  ctx.stroke();

  ctx.closePath();

  // Draw ball

  ctx.beginPath();

  ctx.arc(ballX, ballY, ballRadius, 0, Math.PI \* 2);

  ctx.fill();

  ctx.closePath();

  // Draw left paddle

  ctx.fillRect(0, leftPaddleY, paddleWidth, paddleHeight);

  // Draw right paddle

  ctx.fillRect(canvas.width - paddleWidth, rightPaddleY, paddleWidth, paddleHeight);

  // Draw scores

  ctx.fillText("Left Player Score: " + leftPlayerScore, 15, 20);

  ctx.fillText("Right Player Score: " + rightPlayerScore, canvas.width - 170, 20);

}

// Game loop

function loop() {

  update();

  draw();

  animationId = requestAnimationFrame(loop);

}

$('#message-modal-close').on('click', function() {

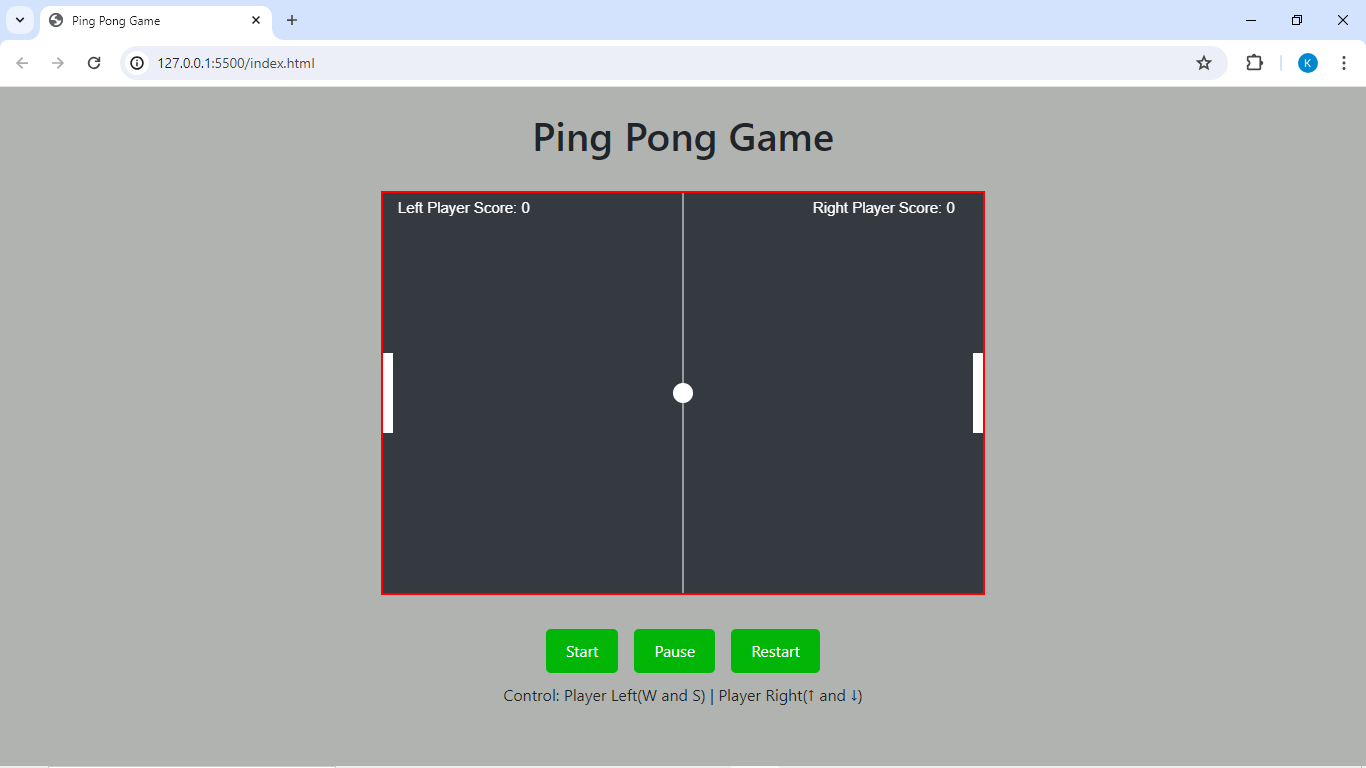
  document.location.reload();

});

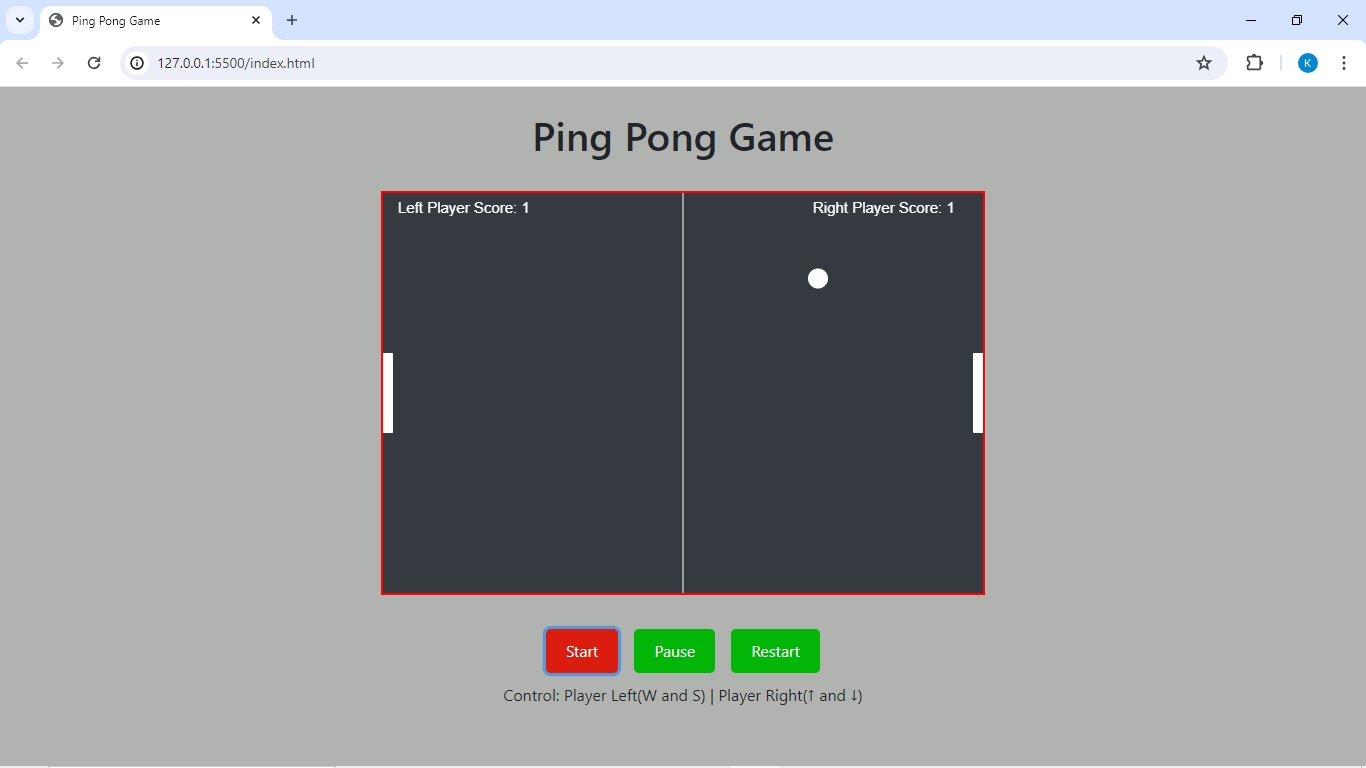
**Results:**

Snippets of the project attached below:

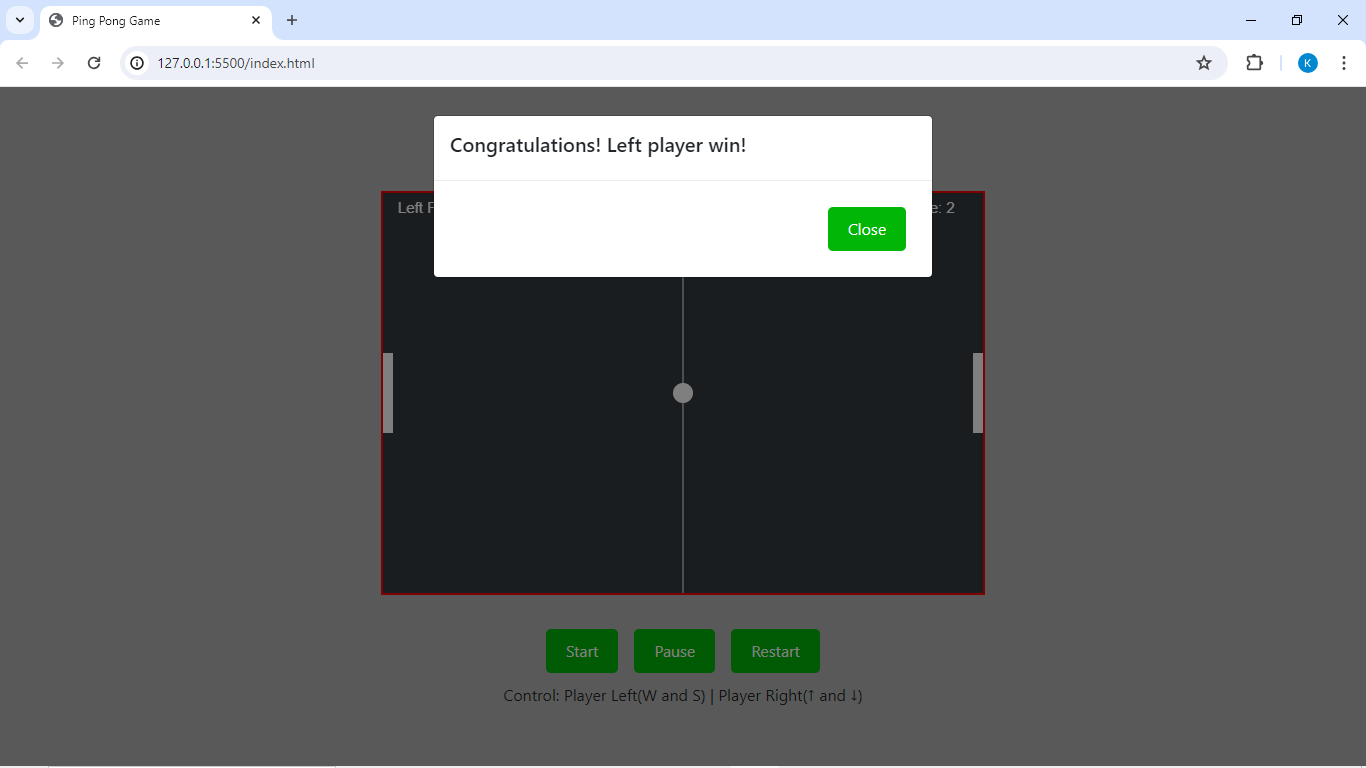
Inicial Page:



Click on button start:



Winner Declared when maxscore reaches:



**LIST OF REFERENCES:**

1. MDN Web Docs - Canvas API: <https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API>
2. Bootstrap Documentation: https://getbootstrap.com/docs/4.0/getting-started/introduction/
3. W3Schools - HTML Canvas Tutorial: https://www.w3schools.com/graphics/canvas\_intro.asp
4. CSS-Tricks - A Complete Guide to Grid: https://css-tricks.com/snippets/css/complete-guide-grid/
5. JavaScript.info - JavaScript Basics: <https://javascript.info/>
6. jQuery Documentation: https://api.jquery.com/
7. Stack Overflow: <https://stackoverflow.com/>
8. FreeCodeCamp - Introduction to the DOM: https://www.freecodecamp.org/news/javascript-dom-manipulation-for-beginners-acec0c2fddbd/
9. CodePen - Explore and Fork Code Snippets: <https://codepen.io/>
10. GitHub - Version Control and Collaboration Platform: <https://github.com/>