

# Ping pong game using Web technologies

## Aim:

Using javascript, make a ping pong game, where the player loses the game if he/she misses 3 attempts of hitting the ball.

## Introduction:

Pong is one of the first computer games that ever created, this simple "tennis like" game features two paddles and a ball, the goal is to defeat your opponent by being the first one to gain 10 point, a player gets a point once the opponent misses a ball. The game can be played with two human players, or one player against a computer controlled paddle. Table tennis, a recreational activity and an Olympic sport since 1988, is also known by the term 'ping-pong'.

Grab one of the digital paddles and enjoy an exciting Ping Pong experience. In this simple yet challenging sports game, you need to hit the ball to your opponent's side of the table. First player to reach 3 wins the match! So be quick on your feet and confuse your opponent with tricky shoots. You can even control the speed of your hits by paying attention to the intensity of your flick.

The project shows how the web technologies are helpful in creating amazing games, by combining HTML for structuring your page, CSS for styling, and JAVASCRIPT for making page more dynamic. The paddle movement to hit the ball release user stress. As we know hitting something while anger, always be a chartbuster.

This project plays a role as stress releasing tool or a way to pass time. Whether played on a desktop computer or laptop, the ping pong game promises an immersive experience.

## Technologies used:

- 1. HTML:** HTML stands for **HyperText Markup Language**. It is used to design web pages using the markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages and markup language defines the text document within the tag that define the structure of web pages.
- 2. CSS:** Cascading Style Sheets is used to styles web pages. Cascading Style Sheets are fondly referred to as CSS. The reason for using this is to simplify the process of making web pages presentable. It allows you to apply styles on web pages.
- 3. JavaScript :** JS is the most popular lightweight, interpreted compiled programming language. It can be used for both Client-side as well as Server-side developments. JavaScript also known as a scripting language for web pages.

## DESCRIPTION OF GAME

After open this project we have a screen which give option of selecting number of player as one or two also it provide a screen of rule in which we able to know about detailing in one player choice or two player choice.

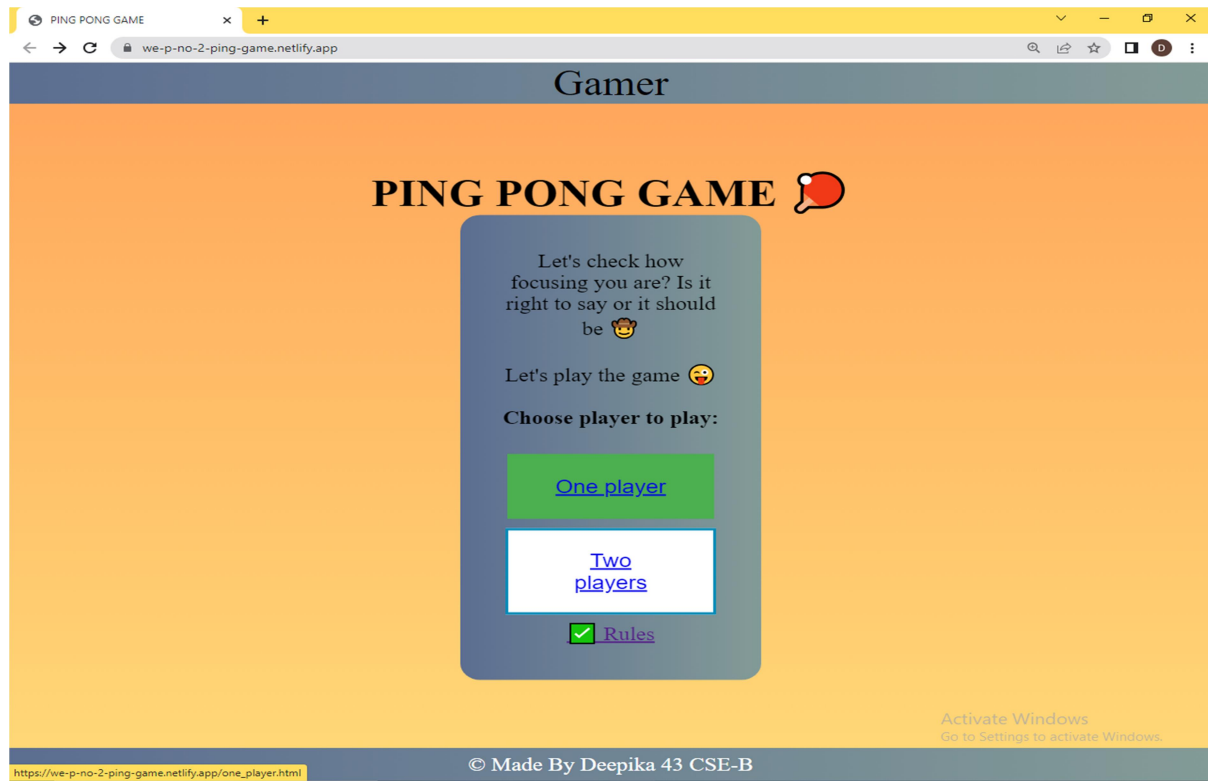


Fig.1.a.

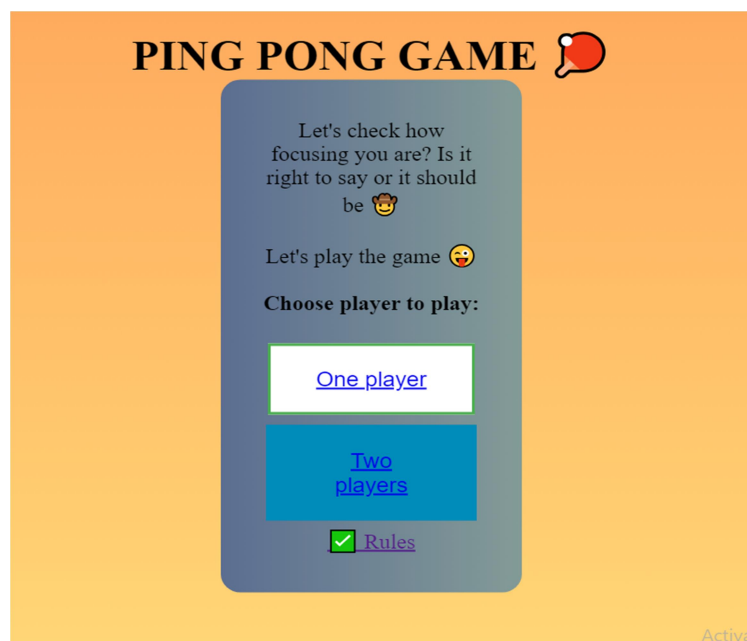


Fig.1.b.

## Rule screen

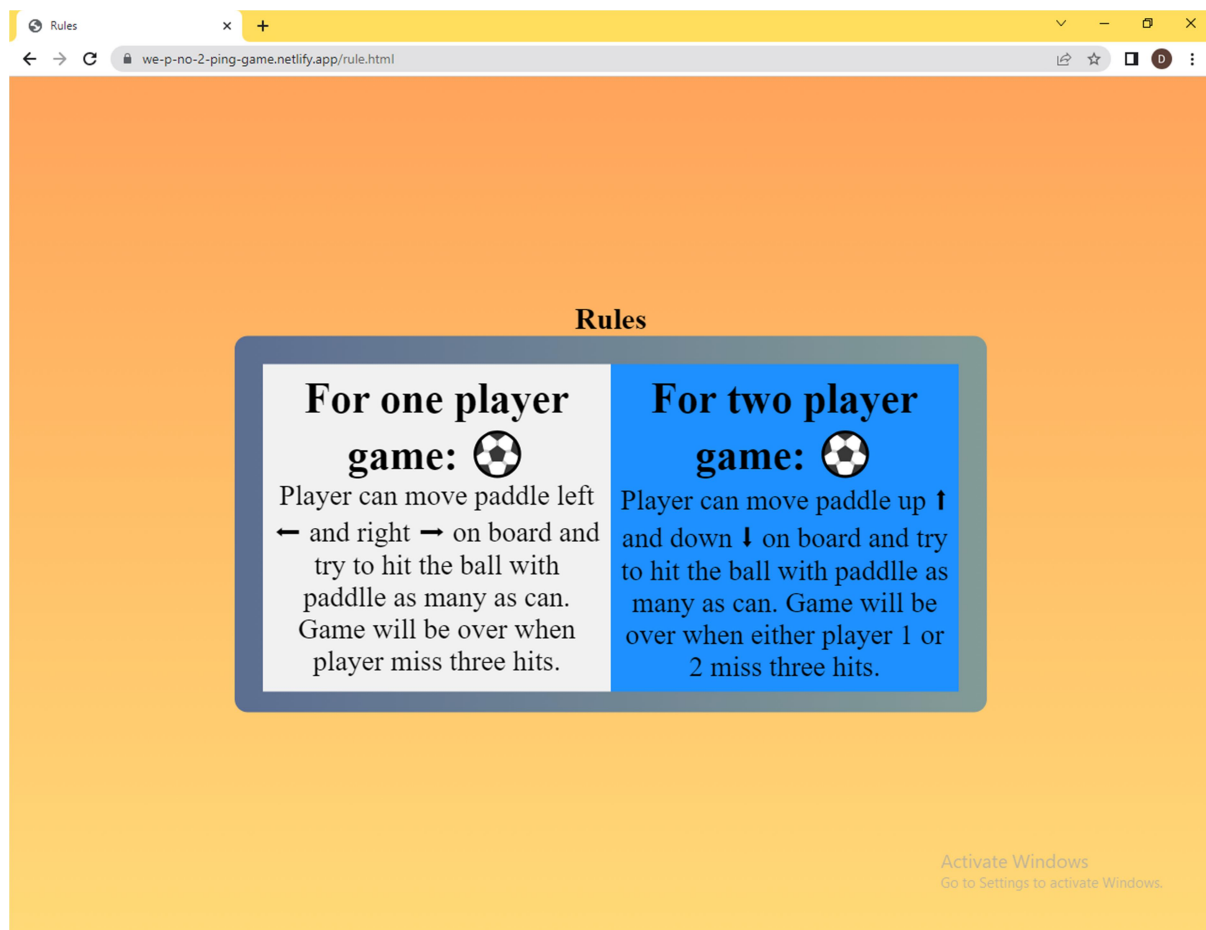


Fig.2.

In one player game, the player can move paddle left and right on board using mouse and score will be counted with missing attempts also and if in case attempts for missing will become three the game will be over and pop up box appear with score.

In two player game, the player on left side is player 1 and player on right side is player 2. Player 1 move the paddle up and down by using 'w' and 's' respectively while player 2 move the paddle up and down using up and down arrow key respectively. On player 1 side the score written are of player 2 missed attempts and vice versa for player 2. After reaching the missed attempts to three the game will be over and pop up appear with score and winning player number.

## ONE PLAYER GAME

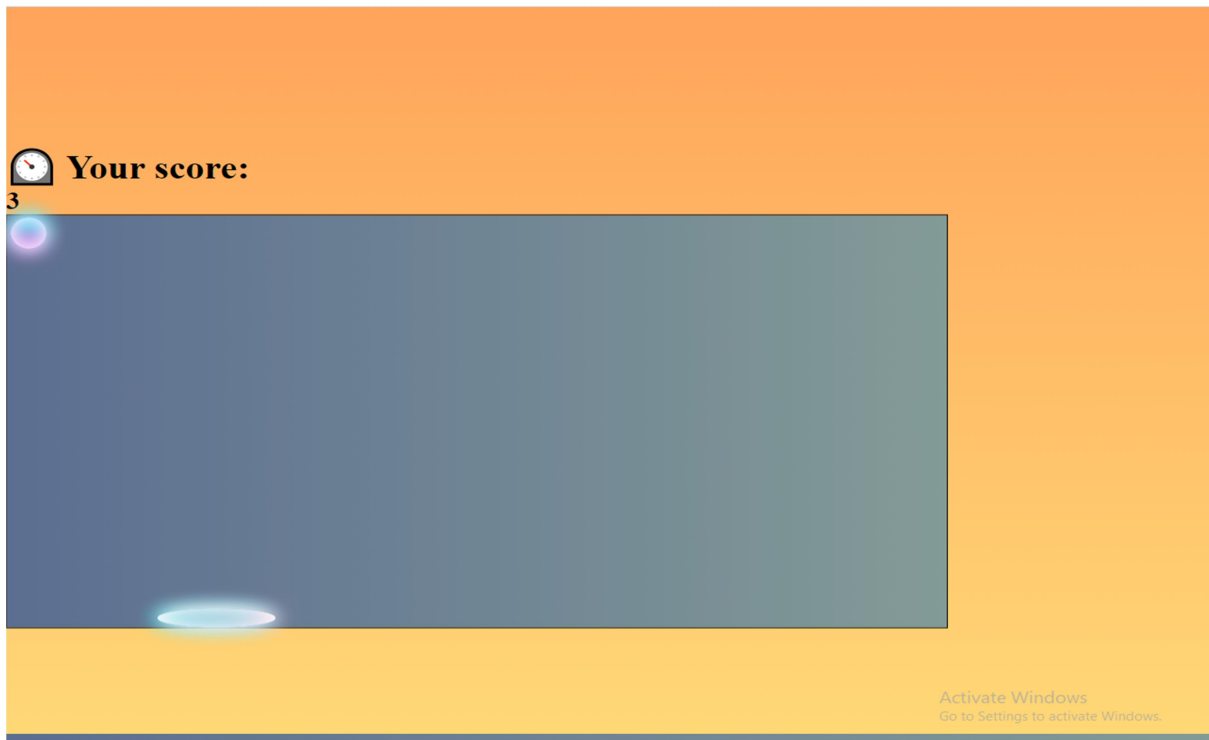


FIG.3.a

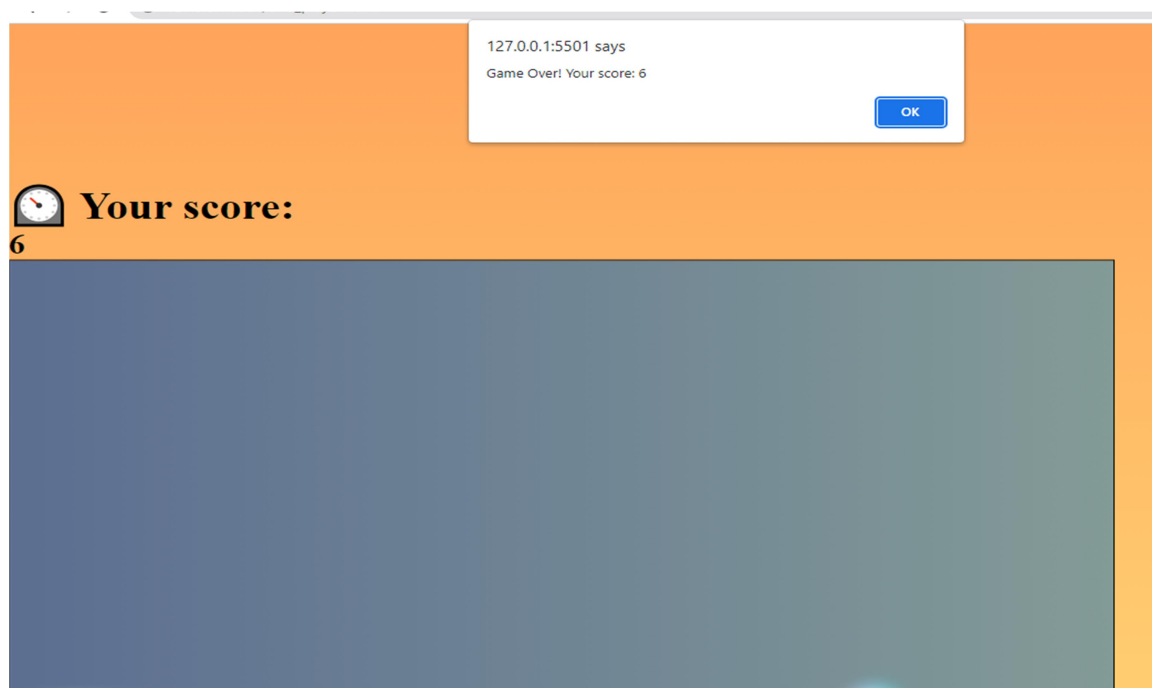


Fig.3.b

## TWO PLAYER GAME



Fig.4.a

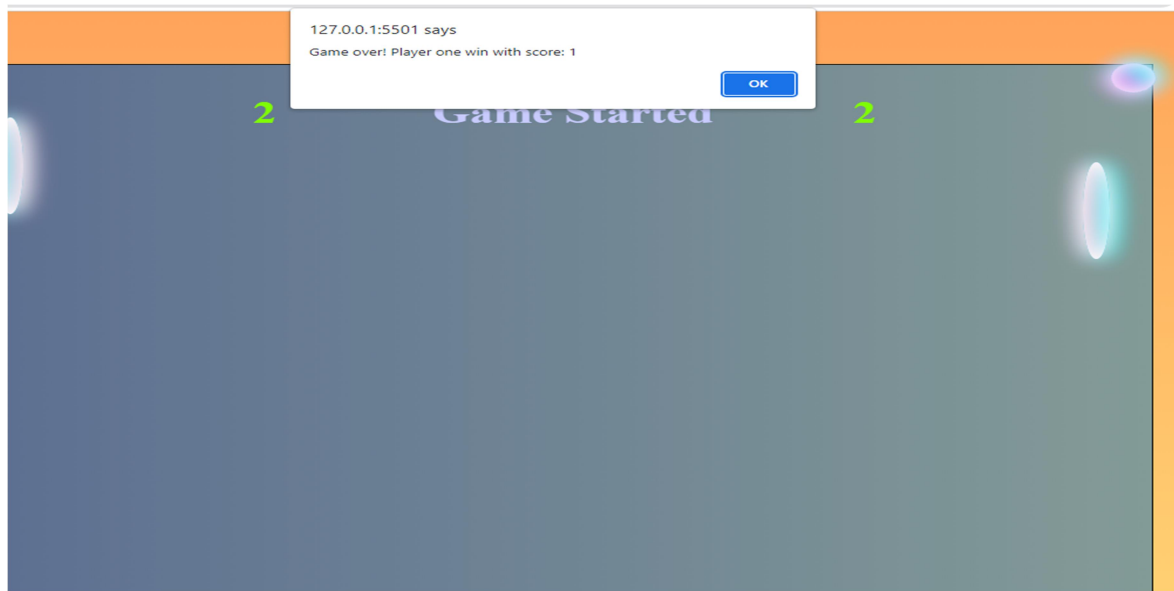


Fig.4.b

**IMPLEMENTATION****A.MAIN FILE OF HTML**

//index.html

```

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title> PING PONG GAME</title>
  <link rel="stylesheet" href="main.css">
</head>

<body>
  <div class="header">Gamer</div>
  <main>
    <h1>PING PONG GAME &#127955;</h1>
    <div class="content">
      <p>Let's check how focusing you are? Is it right to say or it should be
&#129312;</p><br>
      <p>Let's play the game &#128540;</p><br>
      <h4>Choose player to play:</h4><br>
      <button class="button button1"><a href=one_player.html>One player</a></button><br>
      <button class="button button2"><a href=two_player.html>Two players</button><br>
      <a href="rule.html">&#9989; Rules </a>
    </div>
  </main>
  <div class = "footer">
    <p>© Made By Deepika 43 CSE-B</p>
  </div>
</body>
</html>

```

```
//main.css
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

body {
  height: 100vh;
  width: 100vw;
  background-image: linear-gradient(to top, #ffda77, #ffa45b);
  display: block;
  text-align: center;
  justify-content: center;
  align-items: center;
}

.header {
  text-align: center;
  font-size: 30px;
  background-image: linear-gradient(to right, #5c6e91, #839b97);
}

main {
  position: fixed;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
}

.content {
  margin: auto;
  padding: 30px;
  width: 50%;
  background-image: linear-gradient(to right, #5c6e91, #839b97);
  border-radius: 14px;
  text-align: center;
}

.button {
  border: none;
  color: white;
  padding: 16px 32px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 16px;
  margin: 4px 2px;
  transition-duration: 0.4s;
  cursor: pointer;
}
```

```
.button1 {  
  background-color: white;  
  color: black;  
  border: 2px solid #4CAF50;  
}  
  
.button1:hover {  
  background-color: #4CAF50;  
  color: white;  
}  
  
.button2 {  
  background-color: white;  
  color: black;  
  border: 2px solid #008CBA;  
}  
  
.button2:hover {  
  background-color: #008CBA;  
  color: white;  
}  
  
.footer {  
  position: absolute;  
  bottom: 0;  
  width: 100%;  
  display: flex;  
  justify-content: center;  
  padding: 5px;  
  background-image: linear-gradient(to right, #5c6e91, #839b97);  
  color: #fff;  
}
```



**B. IF USER SELECT ONE PLAYER OPTION**

```
//one_player.html
<!-- one player html -->
<!DOCTYPE html>
<html>

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>One player</title>
  <link rel="stylesheet" href="one_player.css">
</head>

<body>
  <div class="board">
    <h1>&#x23F2; Your score:</h1>
    <h2 class="player_score">0</h2>
    <div id="gameArea">
      <div id="ball">
        <div id="ball_effect"></div>
      </div>
      <div id="paddle"></div>
    </div>
  </div>
  <div class="footer">
    <p>© Made By Deepika 43 CSE-B</p>
  </div>
//SCRIPT FILE
<script>

  // Game variables
  var gameArea = document.getElementById("gameArea");
  var ball = document.getElementById("ball");
  var paddle = document.getElementById("paddle");
  var score = document.querySelector('.player_score');
  var attempts = 0;

  // Set initial position and speed of the ball
  var ballX = 0;
  var ballY = 0;
  var ballSpeedX = 3;
  var ballSpeedY = 3;

  // Move the paddle with the mouse
  gameArea.addEventListener("mousemove", function (event) {
    var mouseX = event.clientX - gameArea.offsetLeft;
    if (mouseX >= 0 && mouseX <= gameArea.offsetWidth - paddle.offsetWidth) {
      paddle.style.left = mouseX + "px";
    }
  })
}
```

```

});
// Game loop
function gameLoop() {
  // Update ball position
  ballX += ballSpeedX;
  ballY += ballSpeedY;

  // Reverse ball direction if it hits the walls
  if (ballX <= 0 || ballX >= gameArea.offsetWidth - ball.offsetWidth) {
    ballSpeedX = -ballSpeedX;
  }
  if (ballY <= 0) {
    ballSpeedY = -ballSpeedY;
  }

  // Check if the ball hits the paddle
  if (
    ballY + ball.offsetHeight >= paddle.offsetTop &&
    ballX + ball.offsetWidth >= paddle.offsetLeft &&
    ballX <= paddle.offsetLeft + paddle.offsetWidth
  ) {
    ballSpeedY = -ballSpeedY;
    score.innerHTML = +score.innerHTML + 1;
  }

  // Check if the ball misses the paddle
  if (ballY >= gameArea.offsetHeight - ball.offsetHeight) {
    attempts++;
    if (attempts === 3) {
      // Game over
      alert("Game Over! Your score: " + score.innerHTML);
      resetGame();
    } else {
      resetBall();
    }
  }
}

// Move the ball
ball.style.left = ballX + "px";
ball.style.top = ballY + "px";

// Run the game loop recursively
requestAnimationFrame(gameLoop);
}

// Reset ball position and speed
function resetBall() {
  ballX = 0;
  ballY = 0;
  ballSpeedX = 3;

```

```

    ballSpeedY = 3;
  }
  // Reset game
  function resetGame() {
    score = 0;
    attempts = 0;
    resetBall();
  }

  // Start the game loop
  gameLoop();
</script>
</body>

</html>

```

### //one\_player.css

```

* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

body {
  height: 100vh;
  width: 100vw;
  background-image: linear-gradient(to top, #ffda77, #ffa45b);
  display: block;
  justify-content: center;
  align-items: center;
}

.board {
  position: relative;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
}

#gameArea {
  width: 800px;
  height: 400px;
  position: relative;
  border: 1px solid #000;
  background-image: linear-gradient(to right, #5c6e91, #839b97);
}

#ball {
  width: 30px;

```

```

    height: 30px;
    position: absolute;
    background-color: #000;
    border-radius: 50%;
}

#ball_effect {
    height: 100%;
    width: 100%;
    border-radius: 100px;
    animation: spinBall 0.1s linear infinite;
    box-shadow: inset 0 0 18px #fff,
        inset 6px 0 18px violet,
        inset -6px 0 18px #0ff,
        inset 6px 0 30px violet,
        inset -6px 0 30px #0ff,
        0 0 18px #fff, -4px 0 18px violet, 4px 0 18px #0ff;
}

@keyframes spinBall {
    100% {
        -webkit-transform: rotate(360deg);
        transform: rotate(360deg);
    }
}

#paddle {
    width: 100px;
    height: 18px;
    position: absolute;
    bottom: 0;
    border-radius: 50%;
    box-shadow: inset 0 0 18px #fff,
        inset -6px 0 18px #f3bad6,
        inset 6px 0 18px #0ff,
        inset -6px 0 30px #f3bad6,
        inset 6px 0 30px #0ff,
        0 0 18px #fff, 4px 0 18px #f3bad6, -4px 0 18px #0ff;
}

.footer {
    position: absolute;
    bottom: 0;
    width: 100%;
    display: flex;
    justify-content: center;
    padding: 5px;
    background-image: linear-gradient(to right, #5c6e91, #839b97);
    color: #fff;
}

```

### C. IF USER WANT TO CHECK RULES FOR GAME

//rule.html

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Rules</title>
  <link rel="stylesheet" href="rule.css">
</head>

<body>
  <div class="board">
    <h1>Rules</h1>
    <div class="content">
      <div class="flex-container">
        <div class="flex-item-left">
          <h2>For one player game: &#9917;</h2>
          <p>
            Player can move paddle left &#11013; and right &#10145; on board and try to hit the
            ball with paddlle as
            many as
            can.
            Game will be over when player miss three hits.
          </p>
        </div>
        <div class="flex-item-right">
          <h2>For two player game: &#9917;</h2>
          <p>
            Player can move paddle up &#11014; and down &#11015; on board and try to hit the
            ball with paddlle as many
            as can.
            Game will be over when either player 1 or 2 miss three hits.
          </p>
        </div>
      </div>
    </div>
  </div>
</body>

</html>
```

**//rule.css**

```
/* rule.css */
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

body {
  height: 100vh;
  width: 100vw;
  background-image: linear-gradient(to top, #ffda77, #ffa45b);
  display: block;
  text-align: center;
  justify-content: center;
  align-items: center;
}

.board {
  position: fixed;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
}

.content {
  margin: auto;
  padding: 30px;
  width: 800px;
  height: 400px;
  background-image: linear-gradient(to right, #5c6e91, #839b97);
  border-radius: 14px;
  text-align: center;
}

.flex-container {
  display: flex;
  flex-wrap: wrap;
  font-size: 30px;
  text-align: center;
}

.flex-item-left {
  background-color: #f1f1f1;
  padding: 10px;
  flex: 50%;
}
```

```
.flex-item-right {
  background-color: dodgerblue;
  padding: 10px;
  flex: 50%;
}

/* Responsive layout - makes a one column-layout instead of a two-column layout */
@media (max-width: 800px) {

  .flex-item-right,
  .flex-item-left {
    flex: 100%;
  }
}
```

#### **D. IF USER SELECT TWO PLAYER**

```
//two players simultaneously play html
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Two players</title>
  <link rel="stylesheet" href="two_player.css">
</head>

<body>
  <div class="board">
    <div class="ball">
      <div class="ball_effect"></div>
    </div>
    <div class="paddle_1 paddle"></div>
    <div class="paddle_2 paddle"></div>
    <h1 class="player_1_score">0</h1>
    <h1 class="player_2_score">0</h1>
    <h1 class="message">
      Press Enter to Play Pong
    </h1>
  </div>
  <div class="footer">
    <p>© Made By Deepika 43 CSE-B</p>
  </div>
  <script src="two_player.js"></script>
</body>
</html>
```

**//two\_player.css**

/\* two players simultaneously play css \*/

```

* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

body {
  height: 100vh;
  width: 100vw;
  background-image: linear-gradient(to top, #ffda77, #ffa45b);
  display: flex;
  justify-content: center;
  align-items: center;
}

.board {
  height: 85vh;
  width: 80vw;
  background-image: linear-gradient(to right, #5c6e91, #839b97);
  border: 1px solid #000;
}

.footer {
  position: absolute;
  bottom: 0;
  width: 100%;
  display: flex;
  justify-content: center;
  padding: 5px;
  background-image: linear-gradient(to right, #5c6e91, #839b97);
  color: #fff;
}

.ball {
  height: 30px;
  width: 30px;
  border-radius: 50%;
  position: fixed;
  top: calc(50% - 15px);
  left: calc(50% - 15px);
}

.ball_effect {
  height: 100%;
  width: 100%;
  border-radius: 100px;
  animation: spinBall 0.1s linear infinite;
  box-shadow: inset 0 0 18px #fff,
    inset 6px 0 18px violet,
    inset -6px 0 18px #0ff,

```



```

    inset 6px 0 30px violet,
    inset -6px 0 30px #0ff,
    0 0 18px #fff, -4px 0 18px violet, 4px 0 18px #0ff;
}

@keyframes spinBall {
  100% {
    -webkit-transform: rotate(360deg);
    transform: rotate(360deg);
  }
}

.paddle {
  height: 100px;
  width: 18px;
  border-radius: 50%;
  position: fixed;
}

.paddle_1 {
  top: calc(7.5vh + 55px);
  left: calc(10vw + 30px);
  box-shadow: inset 0 0 18px #fff,
    inset -6px 0 18px #f3bad6,
    inset 6px 0 18px #0ff,
    inset -6px 0 30px #f3bad6,
    inset 6px 0 30px #0ff,
    0 0 18px #fff, 4px 0 18px #f3bad6, -4px 0 18px #0ff;
}

.paddle_2 {
  top: calc(85vh + 7.5vh - 100px - 55px);
  right: calc(10vw + 30px);
  box-shadow: inset 0 0 18px #fff,
    inset 6px 0 18px #f3bad6,
    inset -6px 0 18px #0ff,
    inset 6px 0 30px #f3bad6,
    inset -6px 0 30px #0ff,
    0 0 18px #fff, -4px 0 18px #f3bad6, 4px 0 18px #0ff;
}

.player_1_score {
  height: 50px;
  width: 50px;
  color: chartreuse;
  position: fixed;
  left: 30vw;
  margin-top: 30px;
}

```

```
.player_2_score {
  height: 50px;
  width: 50px;
  color: chartreuse;
  position: fixed;
  left: 70vw;
  margin-top: 30px;
}

.message {
  position: fixed;
  /* color: #48426d; */
  height: 10vh;
  width: 30vw;
  color: #c9cbff;
  left: 38vw;
  margin: 30px auto auto auto;
}

#result {
  height: 200px;
  width: 400px;
  background-color: rgb(56, 56, 56);
  margin: 1rem auto;
  border-radius: 20px;
  font-size: 30px;
  text-align: center;
  color: whitesmoke;
  font-family: 'Courier New', Courier, monospace;
  display: none;
  margin-top: 250px;
}

#btn {
  padding: 0.5rem 1rem;
  border-radius: 20px;
  border: none;
  color: rgb(0, 0, 0);
  background-color: rgb(195, 195, 195);
  font-size: 20px;
  margin-top: 10px;
  cursor: pointer;
  text-transform: uppercase;
  font-family: 'Courier New', Courier, monospace;
}
```

```

//two_player.js
// two players simultaneously play js
// Game variables
let gameState = 'start';
let paddle_1 = document.querySelector('.paddle_1');
let paddle_2 = document.querySelector('.paddle_2');
let board = document.querySelector('.board');
let initial_ball = document.querySelector('.ball');
let ball = document.querySelector('.ball');
const WINNING_SCORE = 3;
var player1 = 0;
var player2 = 0;
// Set initial position and speed of the ball
let score_1 = document.querySelector('.player_1_score');
let score_2 = document.querySelector('.player_2_score');
let message = document.querySelector('.message');
let paddle_1_coord = paddle_1.getBoundingClientRect();
let paddle_2_coord = paddle_2.getBoundingClientRect();
let initial_ball_coord = ball.getBoundingClientRect();
let ball_coord = initial_ball_coord;
let board_coord = board.getBoundingClientRect();
let paddle_common =
  document.querySelector('.paddle').getBoundingClientRect();
var show = false;
let dx = Math.floor(Math.random() * 4) + 3;
let dy = Math.floor(Math.random() * 4) + 3;
let dxd = Math.floor(Math.random() * 2);
let dyd = Math.floor(Math.random() * 2);

document.addEventListener('keydown', (e) => {
  if (e.key === 'Enter') {
    gameState = gameState === 'start' ? 'play' : 'start';
    if (gameState === 'play') {
      message.innerHTML = 'Game Started';
      message.style.left = 42 + 'vw';
      requestAnimationFrame(() => {
        dx = Math.floor(Math.random() * 4) + 3;
        dy = Math.floor(Math.random() * 4) + 3;
        dxd = Math.floor(Math.random() * 2);
        dyd = Math.floor(Math.random() * 2);
        moveBall(dx, dy, dxd, dyd);
      });
    }
  }
  if (gameState === 'play') {
    if (e.key === 'w') {
      paddle_1.style.top =
        Math.max(
          board_coord.top,
          paddle_1_coord.top - window.innerHeight * 0.06
        )
    }
  }

```

```

        ) + 'px';
        paddle_1_coord = paddle_1.getBoundingClientRect();
    }
    if (e.key === 's') {
        paddle_1.style.top =
            Math.min(
                board_coord.bottom - paddle_common.height,
                paddle_1_coord.top + window.innerHeight * 0.06
            ) + 'px';
        paddle_1_coord = paddle_1.getBoundingClientRect();
    }

    if (e.key === 'ArrowUp') {
        paddle_2.style.top =
            Math.max(
                board_coord.top,
                paddle_2_coord.top - window.innerHeight * 0.1
            ) + 'px';
        paddle_2_coord = paddle_2.getBoundingClientRect();
    }
    if (e.key === 'ArrowDown') {
        paddle_2.style.top =
            Math.min(
                board_coord.bottom - paddle_common.height,
                paddle_2_coord.top + window.innerHeight * 0.1
            ) + 'px';
        paddle_2_coord = paddle_2.getBoundingClientRect();
    }
    }
});

// Update ball position
function moveBall(dx, dy, dxd, dyd) {
    if (ball_coord.top <= board_coord.top) {
        dyd = 1;
    }
    if (ball_coord.bottom >= board_coord.bottom) {
        dyd = 0;
    }
    // Check if the ball hits the paddle
    if (
        ball_coord.left <= paddle_1_coord.right &&
        ball_coord.top >= paddle_1_coord.top &&
        ball_coord.bottom <= paddle_1_coord.bottom
    ) {
        player1 = +player1 + 1;
        dxd = 1;
        dx = Math.floor(Math.random() * 4) + 3;
        dy = Math.floor(Math.random() * 4) + 3;
    }
}

```

```

if (
  ball_coord.right >= paddle_2_coord.left &&
  ball_coord.top >= paddle_2_coord.top &&
  ball_coord.bottom <= paddle_2_coord.bottom
) {
  player2 = +player2 + 1;
  dx = 0;
  dx = Math.floor(Math.random() * 4) + 3;
  dy = Math.floor(Math.random() * 4) + 3;
}
// Check if the ball misses the paddle
if (
  ball_coord.left <= board_coord.left ||
  ball_coord.right >= board_coord.right
) {
  if (ball_coord.left <= board_coord.left) {
    score_2.innerHTML = +score_2.innerHTML + 1;

    resetBall();
  } else {
    score_1.innerHTML = +score_1.innerHTML + 1;

    resetBall();
  }

  if (score_1.innerHTML >= WINNING_SCORE) {
    alert("Game over! Player one win with score: " + player1)
    resetGame();
  }
  else if (score_2.innerHTML >= WINNING_SCORE) {
    alert("Game over! Player two winwin with score: " + player2)
    resetGame();
  }
  return;
}
ball.style.top = ball_coord.top + dy * (dyd == 0 ? -1 : 1) + 'px';
ball.style.left = ball_coord.left + dx * (dxd == 0 ? -1 : 1) + 'px';
ball_coord = ball.getBoundingClientRect();
requestAnimationFrame(() => {
  moveBall(dx, dy, dxd, dyd);
});
}

// Reset ball position and speed
function resetBall() {
  gameState = 'start';
  ball_coord = initial_ball_coord;
  ball.style = initial_ball.style;
  message.innerHTML = 'Press Enter to Play Pong';
  message.style.left = 38 + 'vw';

```

```
}  
  
// Reset game  
function resetGame() {  
    score_2.innerHTML = 0;  
    score_1.innerHTML = 0;  
    player1 = 0;  
    player2 = 0;  
    resetBall();  
}  
moveBall(dx, dy, dxd, dyd);
```