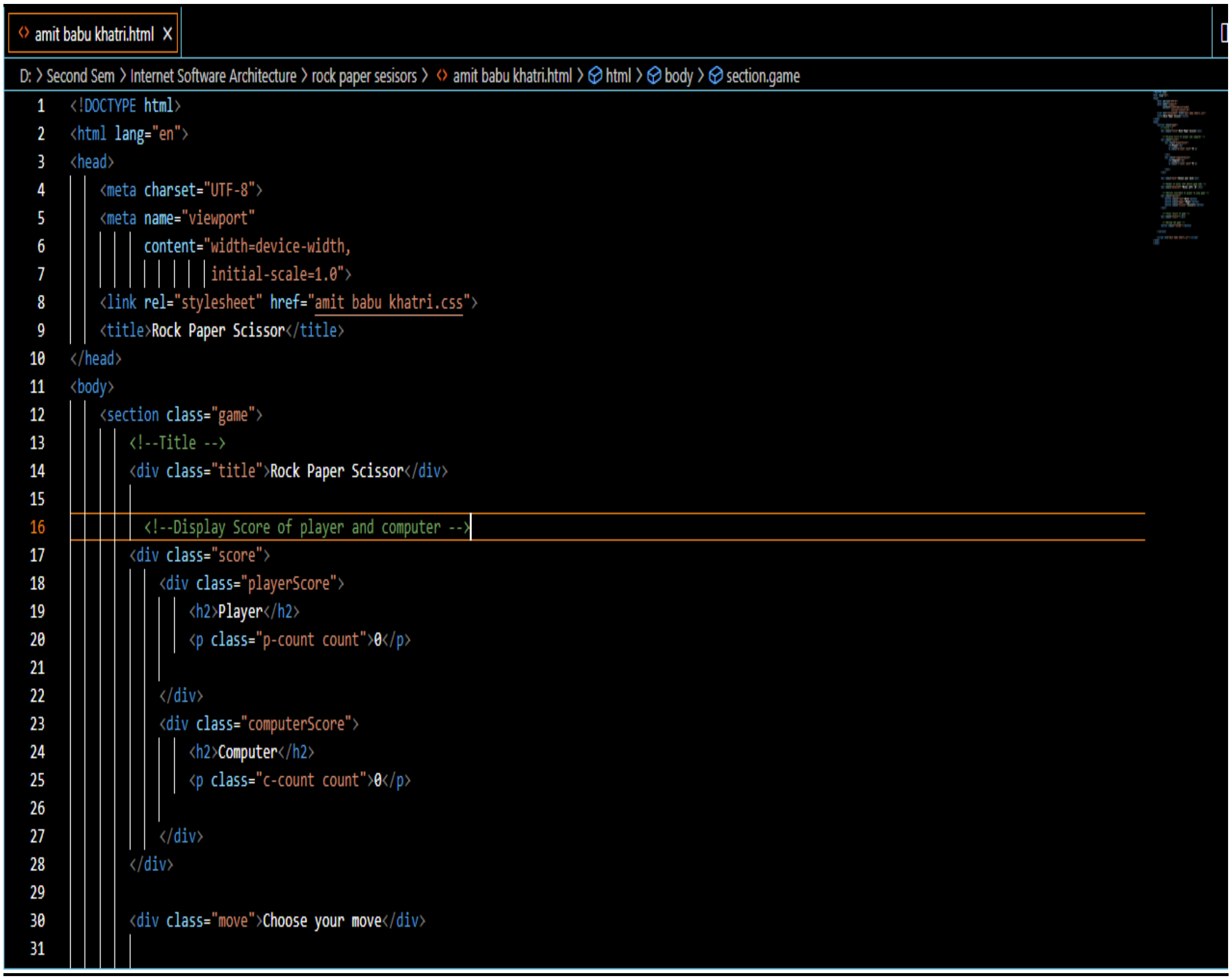
Rock, Paper, Scissors with Javascript

Name= Amit Babu Khatri

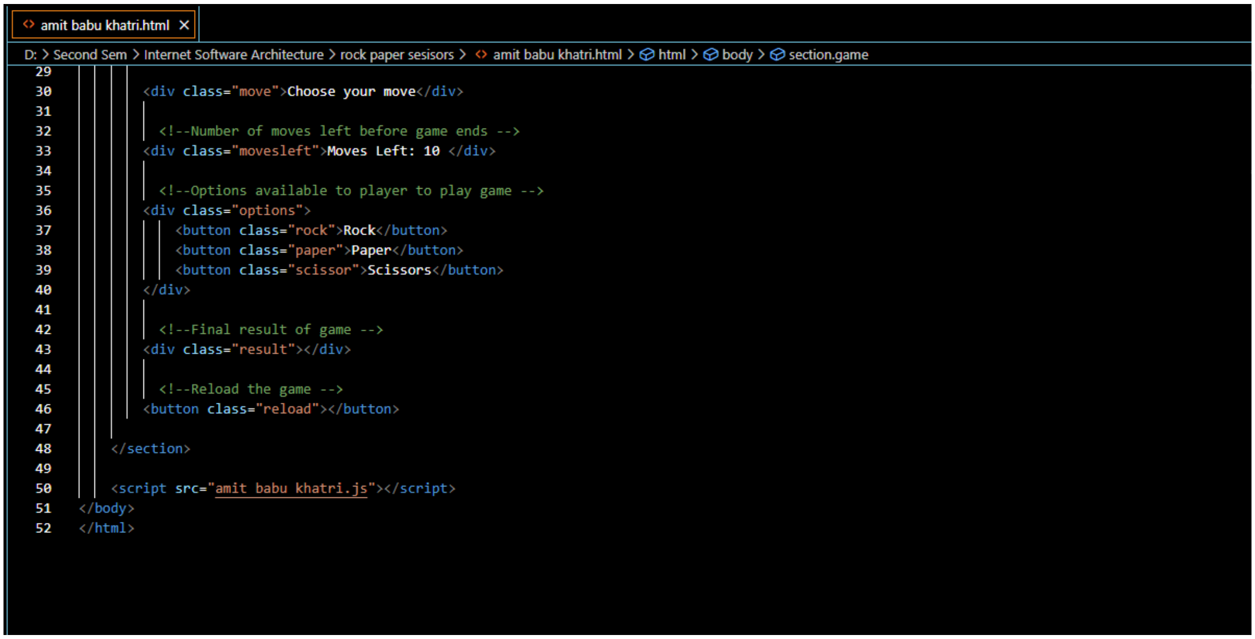
Student id= np03cs4s230291@heraldcollege.edu.np

HTML CODE:



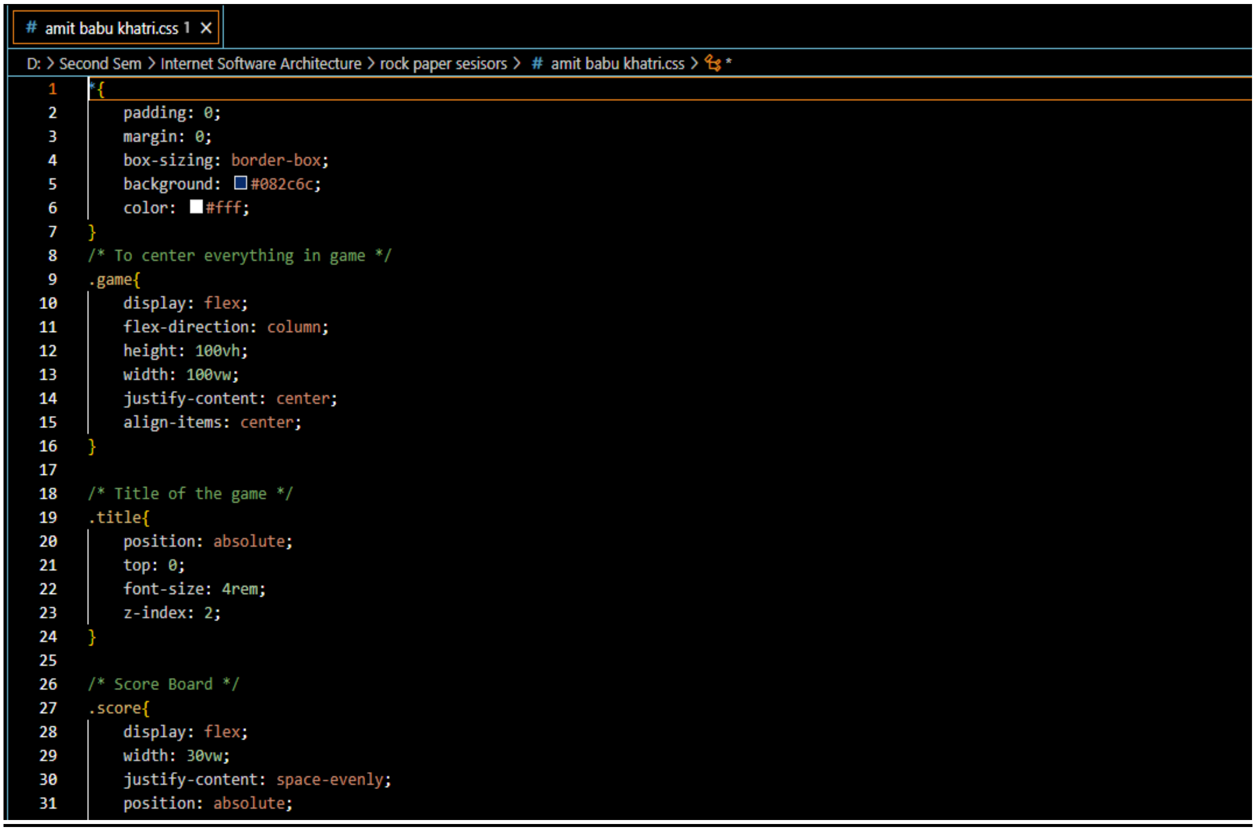
amit babu khatri.html X
D: > Second Sem > Internet Software Architecture > rock paper sesisors > amit babu khatri.html > html > body > section.game

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport"
          content="width=device-width,
                   initial-scale=1.0">
    <link rel="stylesheet" href="amit babu khatri.css">
    <title>Rock Paper Scissor</title>
</head>
<body>
    <section class="game">
        <!--Title -->
        <div class="title">Rock Paper Scissor</div>

          <!--Display Score of player and computer -->
        <div class="score">
            <div class="playerScore">
                <h2>Player</h2>
                <p class="p-count count">0</p>

            </div>
            <div class="computerScore">
                <h2>Computer</h2>
                <p class="c-count count">0</p>

            </div>
        </div>

        <div class="move">Choose your move</div>

```

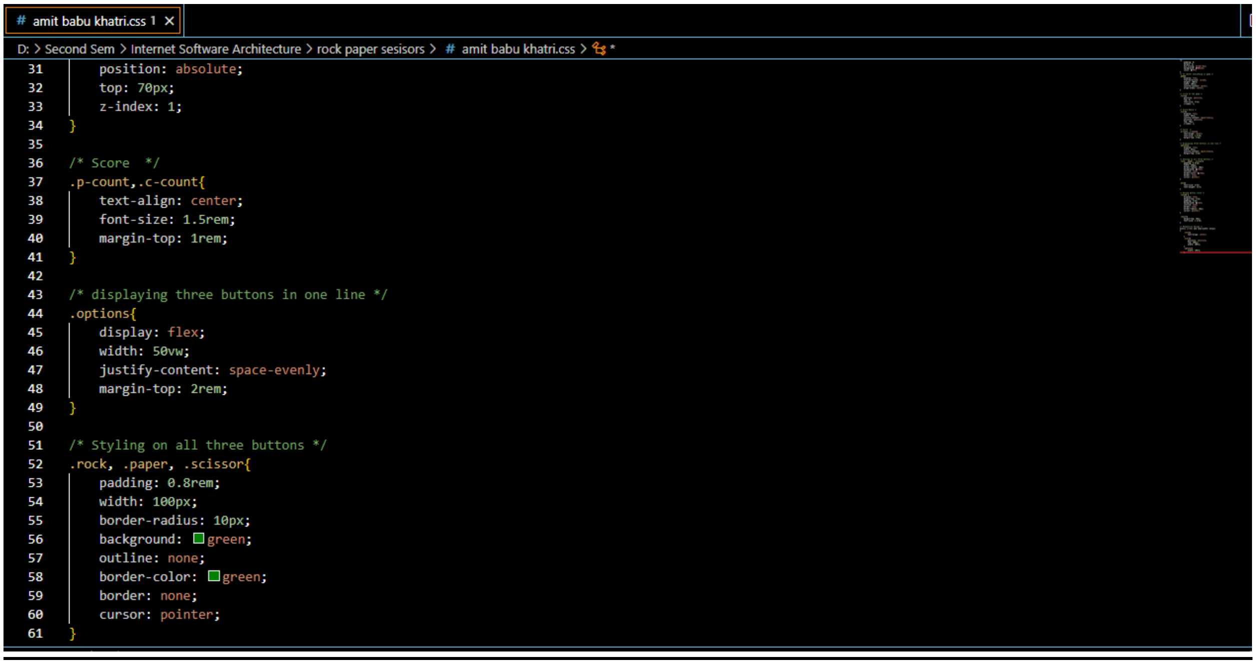

amit babu khatri.html
D: > Second Sem > Internet Software Architecture > rock paper sesisors > amit babu khatri.html > html > body > section.game

```html

        <div class="move">Choose your move</div>

          <!--Number of moves left before game ends -->
        <div class="movesleft">Moves Left: 10 </div>

          <!--Options available to player to play game -->
        <div class="options">
            <button class="rock">Rock</button>
            <button class="paper">Paper</button>
            <button class="scissor">Scissors</button>
        </div>

          <!--Final result of game -->
        <div class="result"></div>

          <!--Reload the game -->
        <button class="reload"></button>

    </section>

    <script src="amit babu khatri.js"></script>
</body>
</html>
```

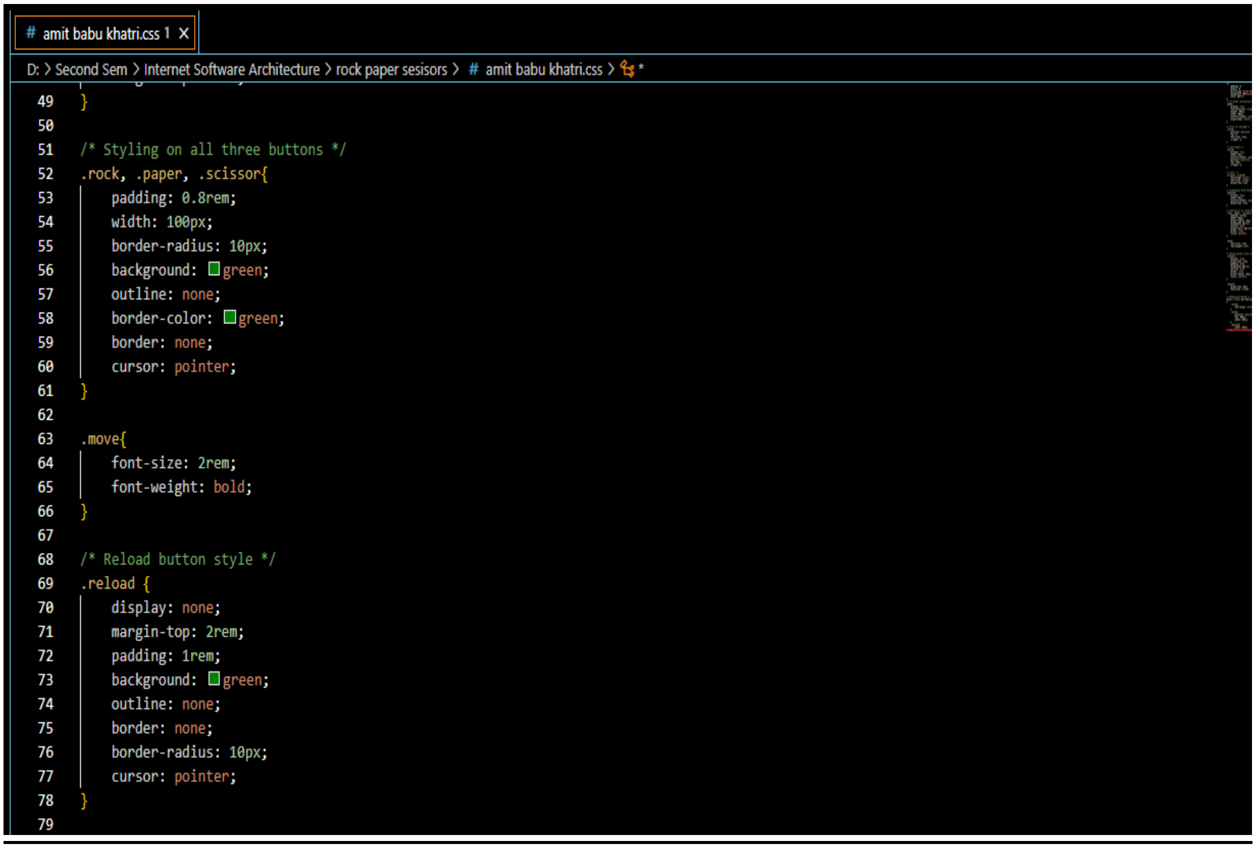
CSS CODE:


amit babu khatri.css 1
D: > Second Sem > Internet Software Architecture > rock paper sesisors > # amit babu khatri.css > *

```css
*{
    padding: 0;
    margin: 0;
    box-sizing: border-box;
    background: #082c6c;
    color: #fff;
}
/* To center everything in game */
.game{
    display: flex;
    flex-direction: column;
    height: 100vh;
    width: 100vw;
    justify-content: center;
    align-items: center;
}

/* Title of the game */
.title{
    position: absolute;
    top: 0;
    font-size: 4rem;
    z-index: 2;
}

/* Score Board */
.score{
    display: flex;
    width: 30vw;
    justify-content: space-evenly;
    position: absolute;
```

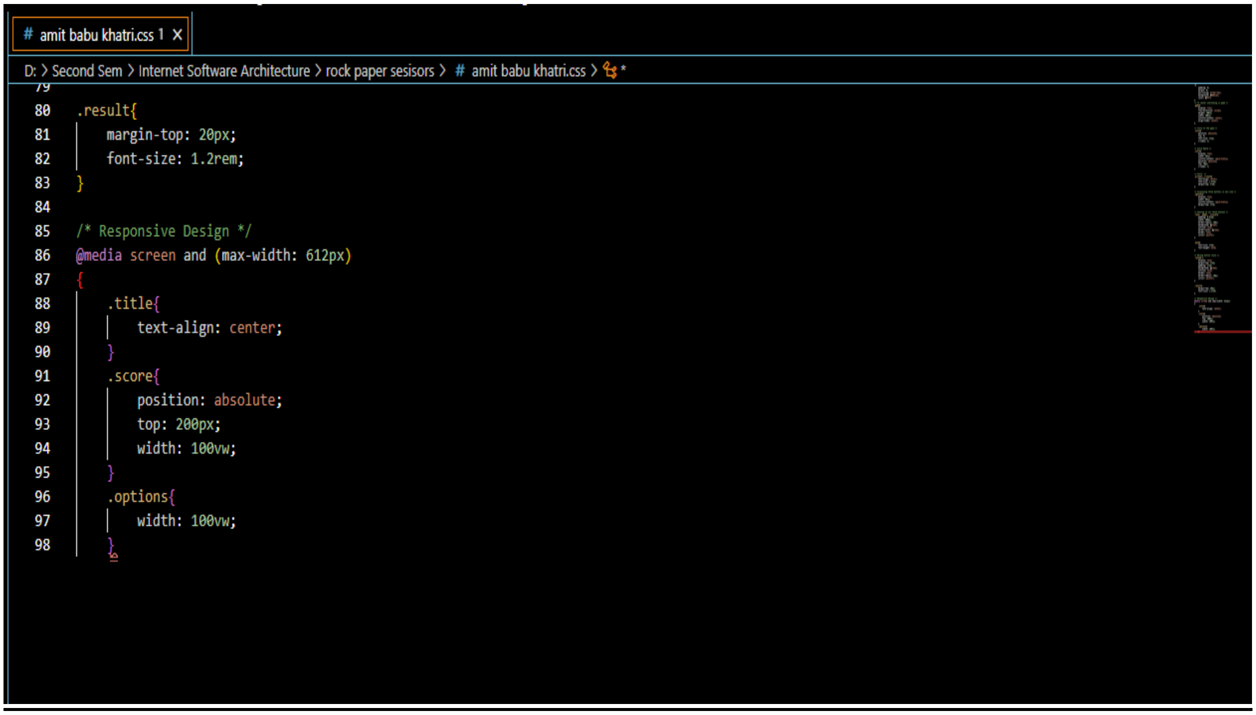

# amit babu khatri.css 1 ×
D: > Second Sem > Internet Software Architecture > rock paper sesisors > # amit babu khatri.css > *

```
    position: absolute;
    top: 70px;
    z-index: 1;
}

/* Score  */
.p-count,.c-count{
    text-align: center;
    font-size: 1.5rem;
    margin-top: 1rem;
}

/* displaying three buttons in one line */
.options{
    display: flex;
    width: 50vw;
    justify-content: space-evenly;
    margin-top: 2rem;
}

/* Styling on all three buttons */
.rock, .paper, .scissor{
    padding: 0.8rem;
    width: 100px;
    border-radius: 10px;
    background: 🟩green;
    outline: none;
    border-color: 🟩green;
    border: none;
    cursor: pointer;
}
```


# amit babu khatri.css 1 ×
D: > Second Sem > Internet Software Architecture > rock paper sesisors > # amit babu khatri.css > *

```
}

/* Styling on all three buttons */
.rock, .paper, .scissor{
    padding: 0.8rem;
    width: 100px;
    border-radius: 10px;
    background: 🟩green;
    outline: none;
    border-color: 🟩green;
    border: none;
    cursor: pointer;
}
```

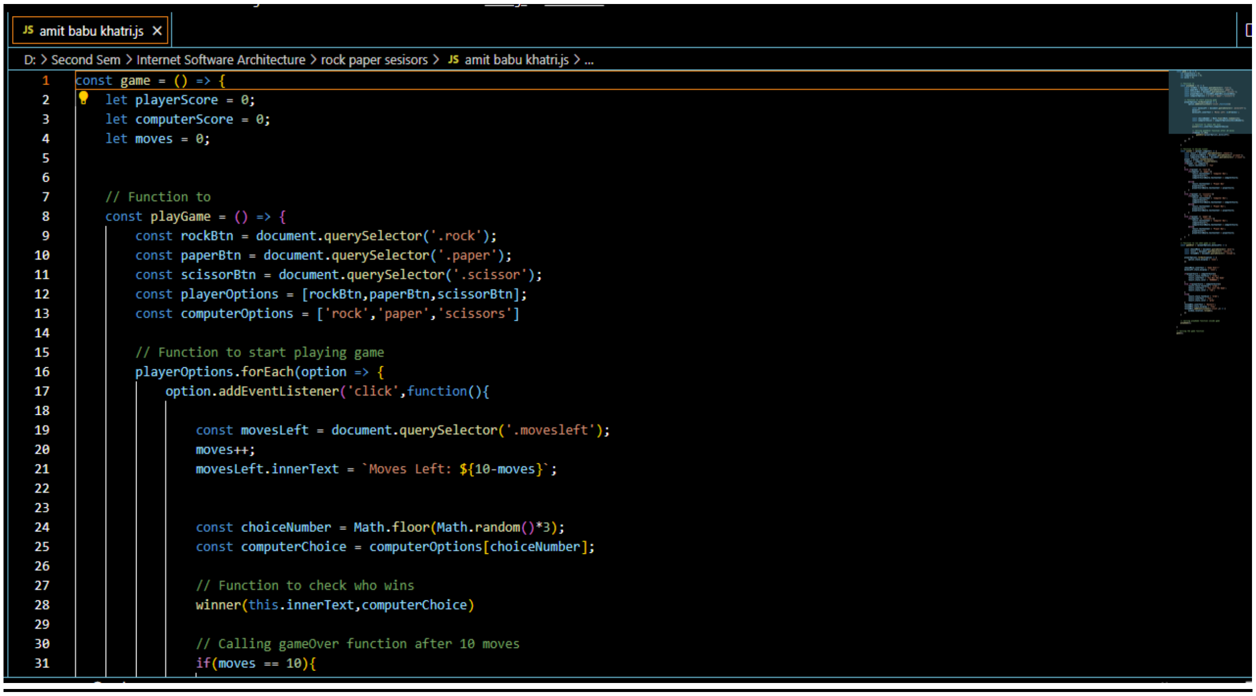
```

.move{
    font-size: 2rem;
    font-weight: bold;
}

/* Reload button style */
.reload {
    display: none;
    margin-top: 2rem;
    padding: 1rem;
    background: 🟩green;
    outline: none;
    border: none;
    border-radius: 10px;
    cursor: pointer;
}

```



# amit babu khatri.css 1 X
D: > Second Sem > Internet Software Architecture > rock paper sesisors > # amit babu khatri.css > *

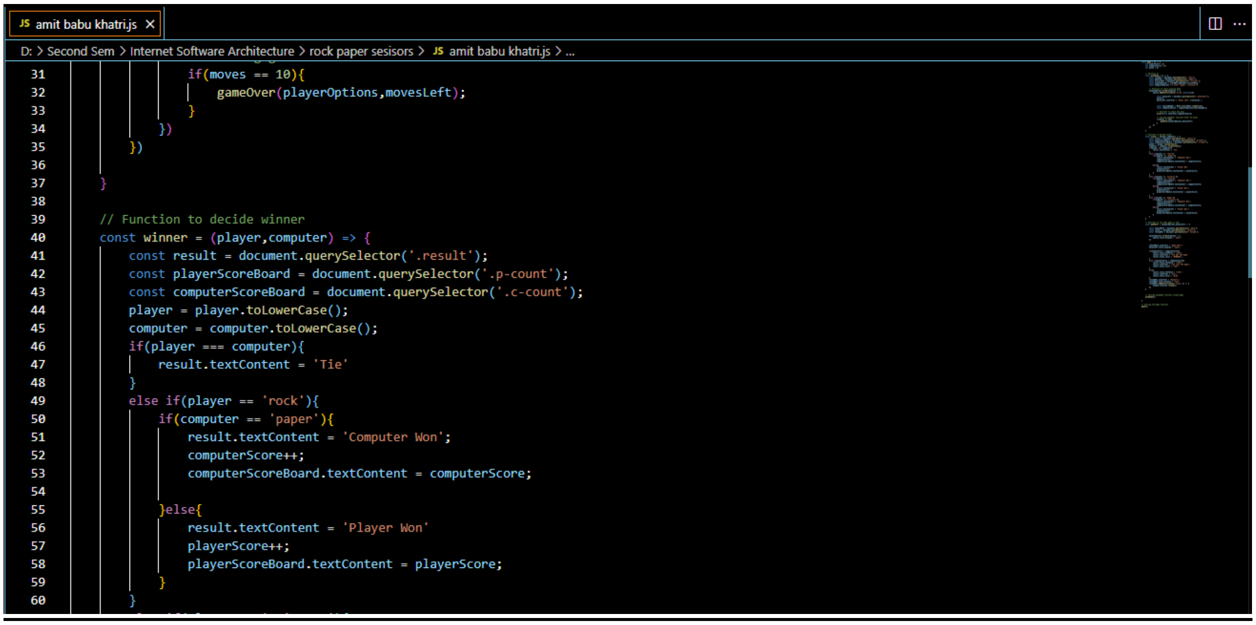
```css
.result{
    margin-top: 20px;
    font-size: 1.2rem;
}

/* Responsive Design */
@media screen and (max-width: 612px)
{
    .title{
        text-align: center;
    }
    .score{
        position: absolute;
        top: 200px;
        width: 100vw;
    }
    .options{
        width: 100vw;
    }
```

JAVA SCRIPT CODE:

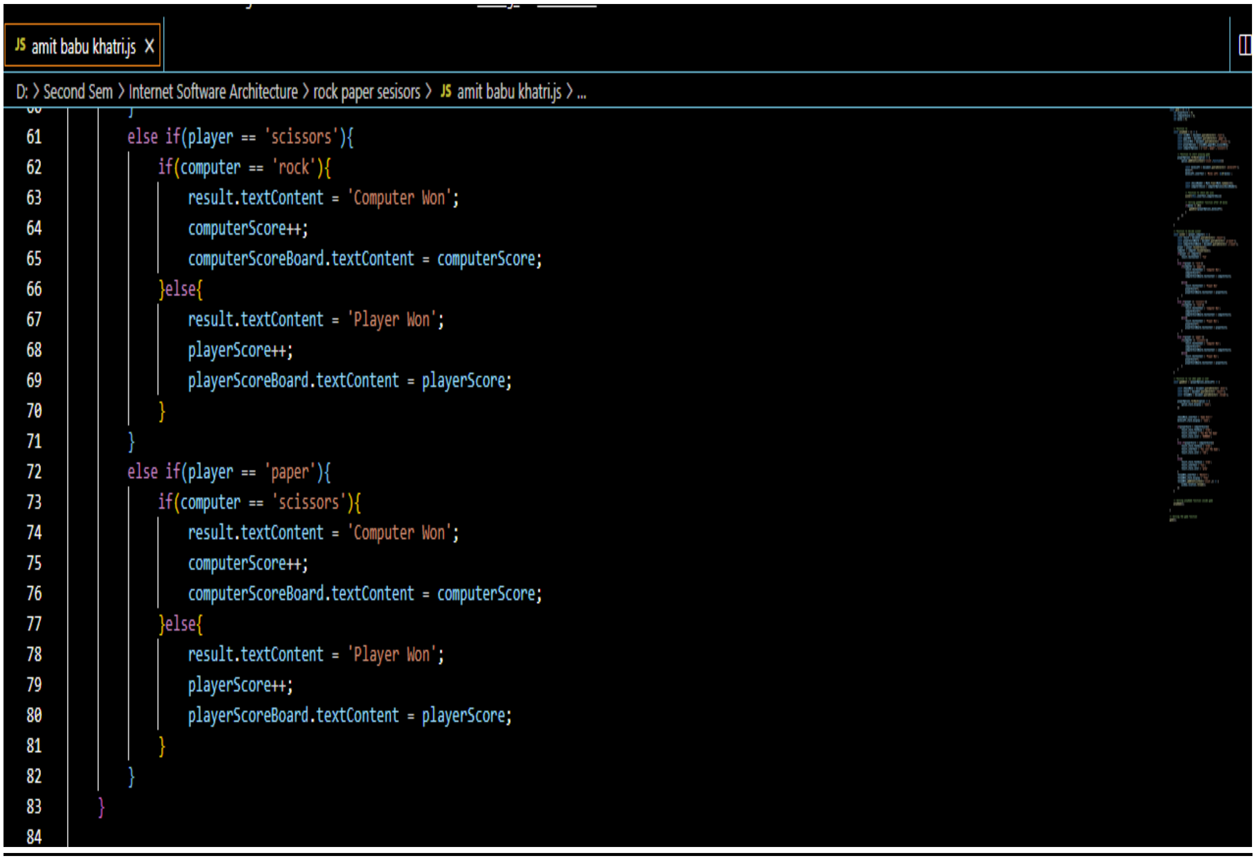


JS amit babu khatri.js X
D: > Second Sem > Internet Software Architecture > rock paper sesisors > JS amit babu khatri.js > ...

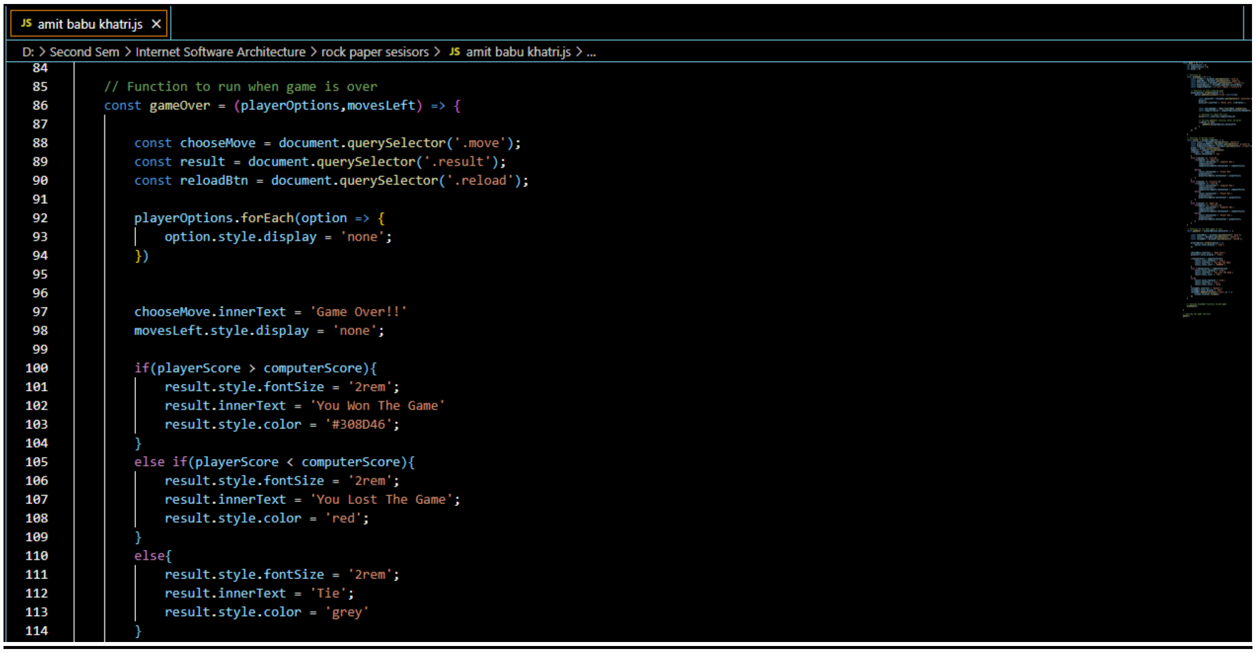
```javascript
const game = () => {
    let playerScore = 0;
    let computerScore = 0;
    let moves = 0;


    // Function to
    const playGame = () => {
        const rockBtn = document.querySelector('.rock');
        const paperBtn = document.querySelector('.paper');
        const scissorBtn = document.querySelector('.scissor');
        const playerOptions = [rockBtn,paperBtn,scissorBtn];
        const computerOptions = ['rock','paper','scissors']

        // Function to start playing game
        playerOptions.forEach(option => {
            option.addEventListener('click',function(){

                const movesLeft = document.querySelector('.movesleft');
                moves++;
                movesLeft.innerText = `Moves Left: ${10-moves}`;


                const choiceNumber = Math.floor(Math.random()*3);
                const computerChoice = computerOptions[choiceNumber];

                // Function to check who wins
                winner(this.innerText,computerChoice)

                // Calling gameOver function after 10 moves
                if(moves == 10){
```


JS amit babu khatri.js X
D: > Second Sem > Internet Software Architecture > rock paper sesisors > JS amit babu khatri.js > ...

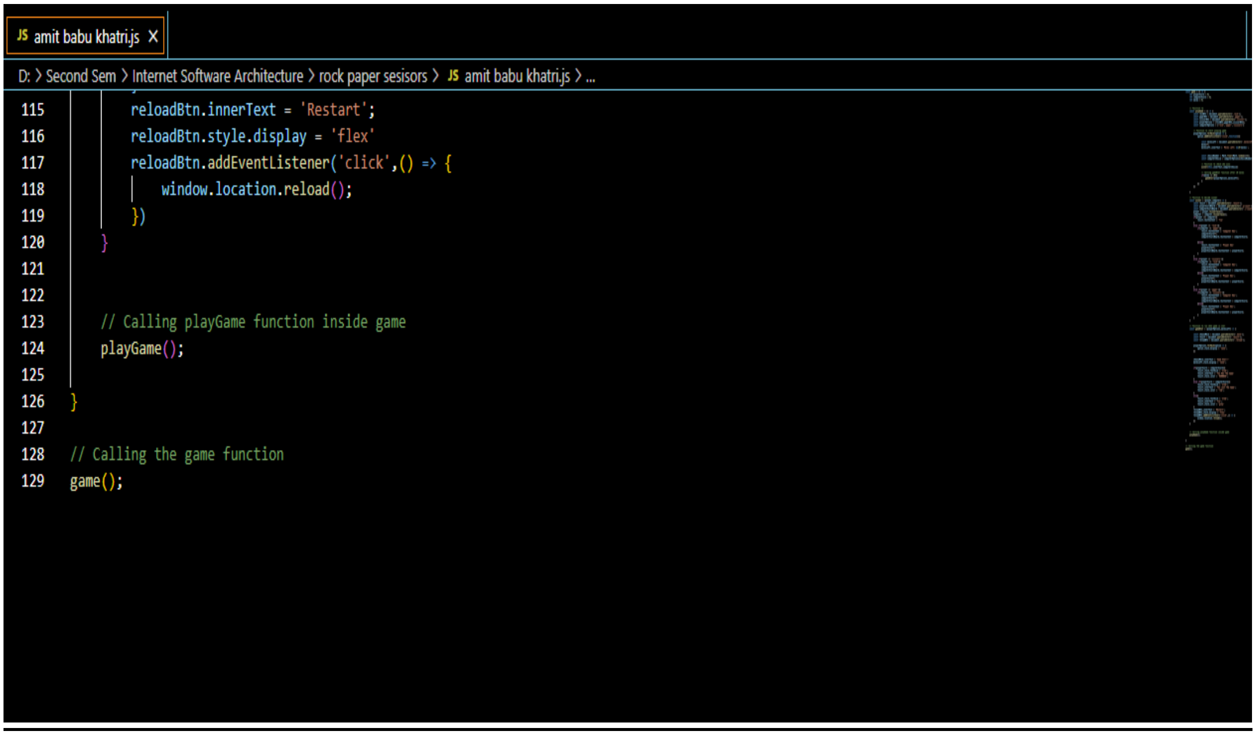
```
                if(moves == 10){
                    gameOver(playerOptions,movesLeft);
                }
            })
        })

    }

    // Function to decide winner
    const winner = (player,computer) => {
        const result = document.querySelector('.result');
        const playerScoreBoard = document.querySelector('.p-count');
        const computerScoreBoard = document.querySelector('.c-count');
        player = player.toLowerCase();
        computer = computer.toLowerCase();
        if(player === computer){
            result.textContent = 'Tie'
        }
        else if(player == 'rock'){
            if(computer == 'paper'){
                result.textContent = 'Computer Won';
                computerScore++;
                computerScoreBoard.textContent = computerScore;

            }else{
                result.textContent = 'Player Won'
                playerScore++;
                playerScoreBoard.textContent = playerScore;
            }
        }
```


JS amit babu khatri.js X
D: > Second Sem > Internet Software Architecture > rock paper sesisors > JS amit babu khatri.js > ...

```
        else if(player == 'scissors'){
            if(computer == 'rock'){
                result.textContent = 'Computer Won';
                computerScore++;
                computerScoreBoard.textContent = computerScore;
            }else{
                result.textContent = 'Player Won';
                playerScore++;
                playerScoreBoard.textContent = playerScore;
            }
        }
        else if(player == 'paper'){
            if(computer == 'scissors'){
                result.textContent = 'Computer Won';
                computerScore++;
                computerScoreBoard.textContent = computerScore;
            }else{
                result.textContent = 'Player Won';
                playerScore++;
                playerScoreBoard.textContent = playerScore;
            }
        }
    }

```

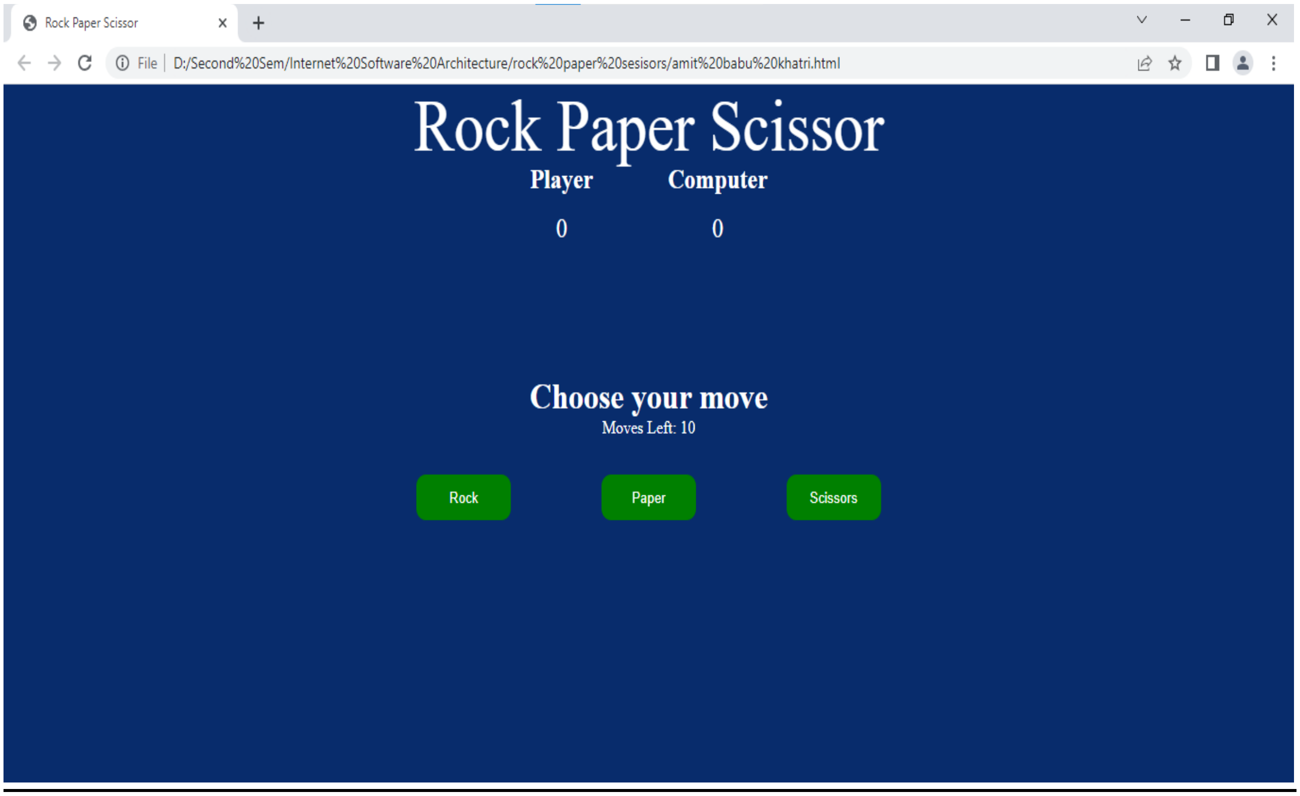

JS amit babu khatri.js ×
D: > Second Sem > Internet Software Architecture > rock paper sesisors > JS amit babu khatri.js > ...

```

    // Function to run when game is over
    const gameOver = (playerOptions,movesLeft) => {

        const chooseMove = document.querySelector('.move');
        const result = document.querySelector('.result');
        const reloadBtn = document.querySelector('.reload');

        playerOptions.forEach(option => {
            option.style.display = 'none';
        })


        chooseMove.innerText = 'Game Over!!'
        movesLeft.style.display = 'none';

        if(playerScore > computerScore){
            result.style.fontSize = '2rem';
            result.innerText = 'You Won The Game'
            result.style.color = '#308D46';
        }
        else if(playerScore < computerScore){
            result.style.fontSize = '2rem';
            result.innerText = 'You Lost The Game';
            result.style.color = 'red';
        }
        else{
            result.style.fontSize = '2rem';
            result.innerText = 'Tie';
            result.style.color = 'grey'
        }
```


JS amit babu khatri.js ×
D: > Second Sem > Internet Software Architecture > rock paper sesisors > JS amit babu khatri.js > ...

```
        reloadBtn.innerText = 'Restart';
        reloadBtn.style.display = 'flex'
        reloadBtn.addEventListener('click',() => {
            window.location.reload();
        })
    }


    // Calling playGame function inside game
    playGame();

}

// Calling the game function
game();
```

RUNNING AND DEBUGGING OF CODE:


Rock Paper Scissor
File | D:/Second%20Sem/Internet%20Software%20Architecture/rock%20paper%20sesisors/amit%20babu%20khatri.html
Rock Paper Scissor
Player
Computer
0
0
Choose your move
Moves Left: 10
Rock
Paper
Scissors

Conclusion:

Overall, this code sets up a basic implementation of the Rock Paper Scissors game using HTML, CSS, and JavaScript. Players can make their move by clicking the corresponding button, and the computer makes a random choice. The game keeps track of scores, displays the result, and allows players to restart after the game is over.

From this code, we can learn about the following concepts:

1.HTML structure and elements.

2.CSS styling and responsive design.

3.JavaScript event handling and DOM manipulation.

4.Implementing game logic using JavaScript functions.

5.Using variables to store and update game data.

6.Reloading the page using JavaScript.