TE CMPN A **Name : Saravana sundar Nadar** **Roll no. : 70**

PID : 192072 **SEM : V**

INTERNET PROGRAMMING - Experiment 2

ROCK-PAPER-SCISSORS

AIM: Create Rock-Paper-Scissor game using JavaScript (Roll No: **51-71**)

THEORY:

Details about all main **HTML & CSS** tags used in my webpage:

➢ **HEAD**

- **<html> :** The <html> tag represents the root of an HTML document.
  The <html> tag is the container for all other HTML elements.
- **<meta> :** The <meta> tag defines metadata about an HTML document. Metadata is data (information)
  about data. <meta> tags always go inside the <head> element, and are typically used to specify character set, page description, keywords, author of the document, and viewport settings.
- **<link> :** The <link> tag defines the relationship between the current document and an external resource.
  The <link> tag is most often used to link to external style sheets.
  The <link> element is an empty element, it contains attributes only.
- **<title> :** The <title> tag defines the title of the document. The title must be text-only, and it is shown in the browser's title bar or in the page's tab.
- **<head> :** The <head> element is a container for metadata (data about data) and is placed between the <html> tag and the <body> tag.
  The following elements can go inside the <head> element:
  <title> (required in every HTML document) <style> <base> <link> <meta> <script> <noscript>
- **<style> :** Used for adding a internal styling element.

➢ **BODY**

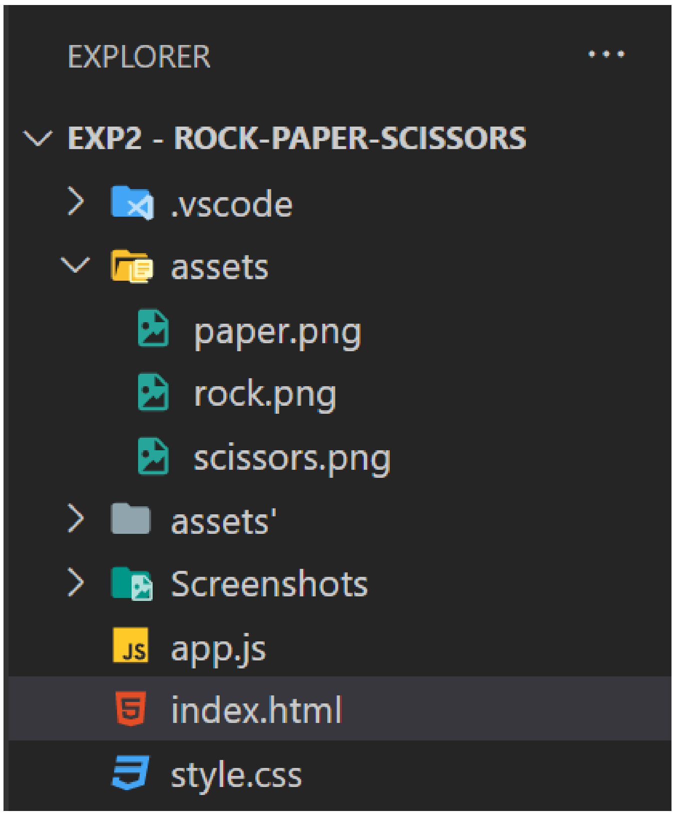
- **<header> :** The <header> element represents a container for introductory content or a set of navigational links.
  A <header> element typically contains:
  - one or more heading elements (<h1> - <h6>)
  - logo or icon
  - authorship information
- **<h1> - <h6> :** The <h1> to <h6> tags are used to define HTML headings.
  <h1> defines the most important heading. <h6> defines the least important heading.

Note: Only use one <h1> per page - this should represent the main heading/subject for the whole page. Also, do not skip heading levels - start with <h1>, then use <h2>, and so on.

- **<div> :** The <div> tag defines a division or a section in an HTML document.
  The <div> tag is used as a container for HTML elements - which is then styled with CSS or manipulated with JavaScript.
  The <div> tag is easily styled by using the class or id attribute.
  Any sort of content can be put inside the <div> tag!
- **<nav> :** The <nav> tag defines a set of navigation links.
  Notice that NOT all links of a document should be inside a <nav> element. The <nav> element is intended only for major block of navigation links.
- **<ul> , <li> :** The <ul> tag defines an unordered (bulleted) list.
  The <li> tag defines a list item.
- **<p> :** The <p> tag defines a paragraph.

CODE :

File Structure :



```
EXPLORER
EXP2 - ROCK-PAPER-SCISSORS
  > .vscode
  v assets
      paper.png
      rock.png
      scissors.png
  > assets'
  > Screenshots
    app.js
    index.html
    style.css
```

✓ **Index.html**

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-
scale=1.0" />
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <link rel="stylesheet" href="./style.css" />
    <title>Rock paper and scissors</title>
  </head>
  <body>
    <section class="game">
      <div class="score">
        <div class="player-score">
          <h2>Player</h2>
          <p>0</p>
        </div>
        <div class="computer-score">
          <h2>Computer</h2>
          <p>0</p>
        </div>
      </div>

      <div class="intro">
        <h1>Rock Paper Scissors</h1>
        <button>Let's Play</button>
      </div>

      <div class="match fadeOut">
        <h2 class="winner">Choose your option</h2>
        <div class="hands">
          <img class="player-hand" src="./assets/rock.png" alt="" />
          <img class="computer-
hand" src="./assets/rock.png" alt="" />
        </div>
        <div class="options">
          <button class="rock">rock</button>
          <button class="paper">paper</button>
          <button class="scissors">scissors</button>
        </div>
      </div>
    </section>

    <script src="app.js"></script>
  </body>
</html>
```

- ✓ **style.css**

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

section {
  height: 100vh;
  background-color: rgb(227, 248, 219);
  font-family: sans-serif;
}

.score {
  color: rgb(15, 59, 4);
  height: 20vh;
  display: flex;
```

```css
  justify-content: space-around;
  align-items: center;
}
.score h2 {
  font-size: 30px;
}
.score p {
  text-align: center;
  padding: 10px;
  font-size: 25px;
}

.intro {
  color: rgb(15, 59, 4);
  height: 50vh;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: space-around;
  transition: opacity 0.5s ease;
}

.intro h1 {
  font-size: 50px;
}
.intro button,
.match button {
  width: 150px;
  height: 50px;
  background: none;
  border: none;
  color: rgb(224, 224, 224);
  font-size: 20px;
  background: rgb(45, 117, 96);
  border-radius: 3px;
  cursor: pointer;
}

.intro button:hover {
  background-color: #e0c9a6;
  color: rgb(15, 59, 4);
}

.match {
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  transition: opacity 0.5s ease 0.5s;
}
.winner {
  color: rgb(15, 59, 4);
  text-align: center;
  font-size: 50px;
}

.hands,
.options {
  display: flex;
  justify-content: space-around;
  align-items: center;
}

.rock:hover {
  background-color: #5a4d41;
}

.paper:hover {
  background-color: #e0c9a6;
  color: rgb(15, 59, 4);
}

.scissors:hover {
  background-color: #c0c0c0;
  color: rgb(15, 59, 4);
}

.player-hand {
  transform: rotateY(180deg);
}

div.fadeOut {
  opacity: 0;
  pointer-events: none;
}

div.fadeIn {
  opacity: 1;
  pointer-events: all;
}

@keyframes shakePlayer {
  0% {
    transform: rotateY(180deg) translateY(0px);
  }
  15% {
    transform: rotateY(180deg) translateY(-50px);
  }
  25% {
    transform: rotateY(180deg) translateY(0px);
  }
  35% {
    transform: rotateY(180deg) translateY(-50px);
  }
  50% {
    transform: rotateY(180deg) translateY(0px);
  }
  65% {
    transform: rotateY(180deg) translateY(-50px);
  }
  75% {
    transform: rotateY(180deg) translateY(0px);
  }
  85% {
    transform: rotateY(180deg) translateY(-50px);
  }
  100% {
    transform: rotateY(180deg) translateY(0px);
  }
}

@keyframes shakeComputer {
  0% {
    transform: translateY(0px);
  }
  15% {
    transform: translateY(-50px);
  }
  25% {
    transform: translateY(0px);
  }
  35% {
    transform: translateY(-50px);
  }
  50% {
    transform: translateY(0px);
```

```css
  }
  65% {
    transform: translateY(-50px);
  }
  75% {
    transform: translateY(0px);
  }
  85% {
    transform: translateY(-50px);
  }
  100% {
    transform: translateY(0px);
  }
}
```

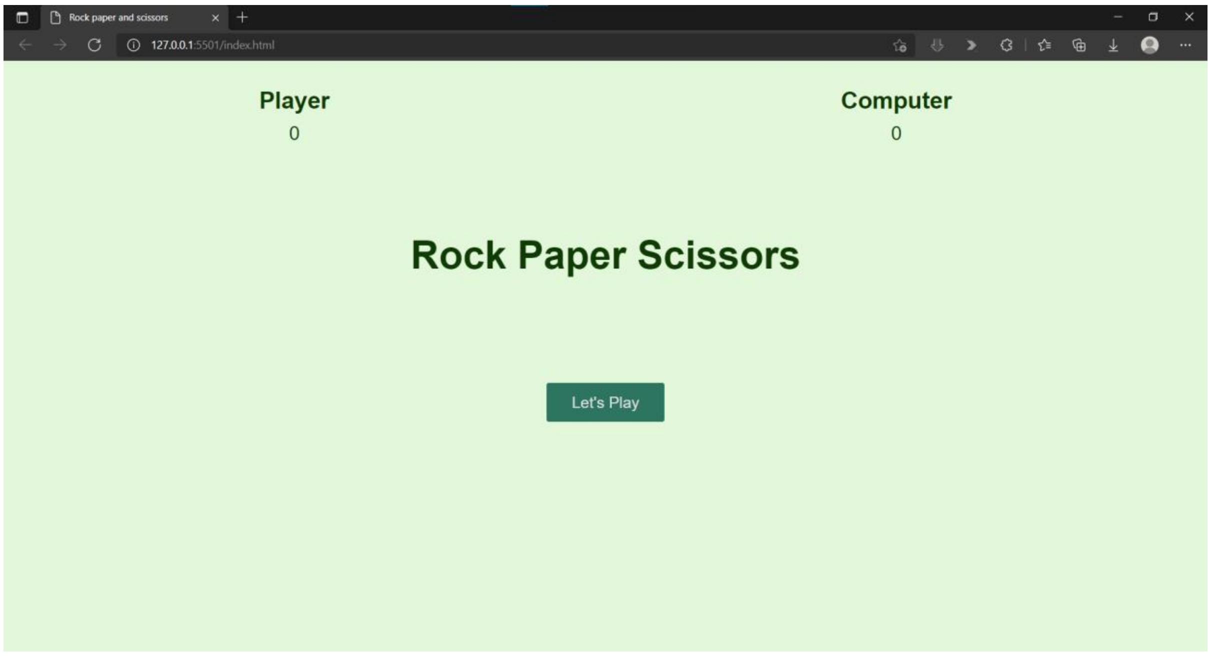
- ✓ **app.js**

```javascript
const game = () => {
  let pScore = 0;
  let cScore = 0;

  //Start the Game
  const startGame = () => {
    const playBtn = document.querySelector(".intro button");
    const introScreen = document.querySelector(".intro");
    const match = document.querySelector(".match");

    playBtn.addEventListener("click", () => {
      introScreen.classList.add("fadeOut");
      match.classList.add("fadeIn");
    });
  };
  //Play Match
  const playMatch = () => {
```

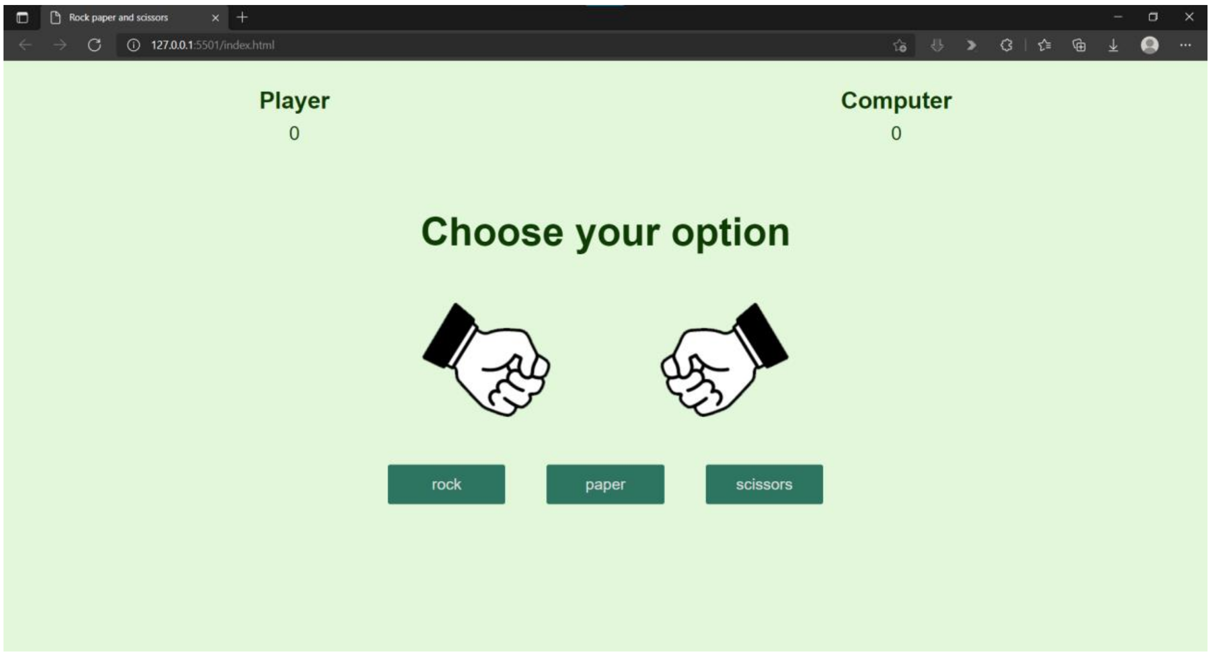
```
  const options = document.querySelectorAll(".options button");
  const playerHand = document.querySelector(".player-hand");
  const computerHand = document.querySelector(".computer-hand");
  const hands = document.querySelectorAll(".hands img");

  hands.forEach(hand => {
    hand.addEventListener("animationend", function() {
      this.style.animation = "";
    });
  });
  //Computer Options
  const computerOptions = ["rock", "paper", "scissors"];

  options.forEach(option => {
    option.addEventListener("click", function() {
      //Computer Choice
      const computerNumber = Math.floor(Math.random() * 3);
      const computerChoice = computerOptions[computerNumber];

      setTimeout(() => {
        //Here is where we call compare hands
        compareHands(this.textContent, computerChoice);
        //Update Images
        playerHand.src = `./assets/${this.textContent}.png`;
        computerHand.src = `./assets/${computerChoice}.png`;
      }, 2000);
      //Animation
      playerHand.style.animation = "shakePlayer 2s ease";
      computerHand.style.animation = "shakeComputer 2s ease";
    });
  });
};

const updateScore = () => {
  const playerScore = document.querySelector(".player-score p");
  const computerScore = document.querySelector(".computer-
score p");
  playerScore.textContent = pScore;
  computerScore.textContent = cScore;
};

const compareHands = (playerChoice, computerChoice) => {
  //Update Text
  const winner = document.querySelector(".winner");
  //Checking for a tie
  if (playerChoice === computerChoice) {
    winner.textContent = "It is a tie";
    return;
  }
  //Check for Rock
  if (playerChoice === "rock") {
    if (computerChoice === "scissors") {
      winner.textContent = "Player Wins";
      pScore++;
      updateScore();
      return;
    } else {
      winner.textContent = "Computer Wins";
      cScore++;
      updateScore();
      return;
    }
  }
  //Check for Paper
  if (playerChoice === "paper") {
    if (computerChoice === "scissors") {
      winner.textContent = "Computer Wins";
      cScore++;
      updateScore();
      return;
    } else {
      winner.textContent = "Player Wins";
      pScore++;
      updateScore();
      return;
    }
  }
  //Check for Scissors
  if (playerChoice === "scissors") {
    if (computerChoice === "rock") {
      winner.textContent = "Computer Wins";
      cScore++;
      updateScore();
      return;
    } else {
      winner.textContent = "Player Wins";
      pScore++;
      updateScore();
      return;
    }
  }
};

//Is call all the inner function
startGame();
playMatch();
};

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

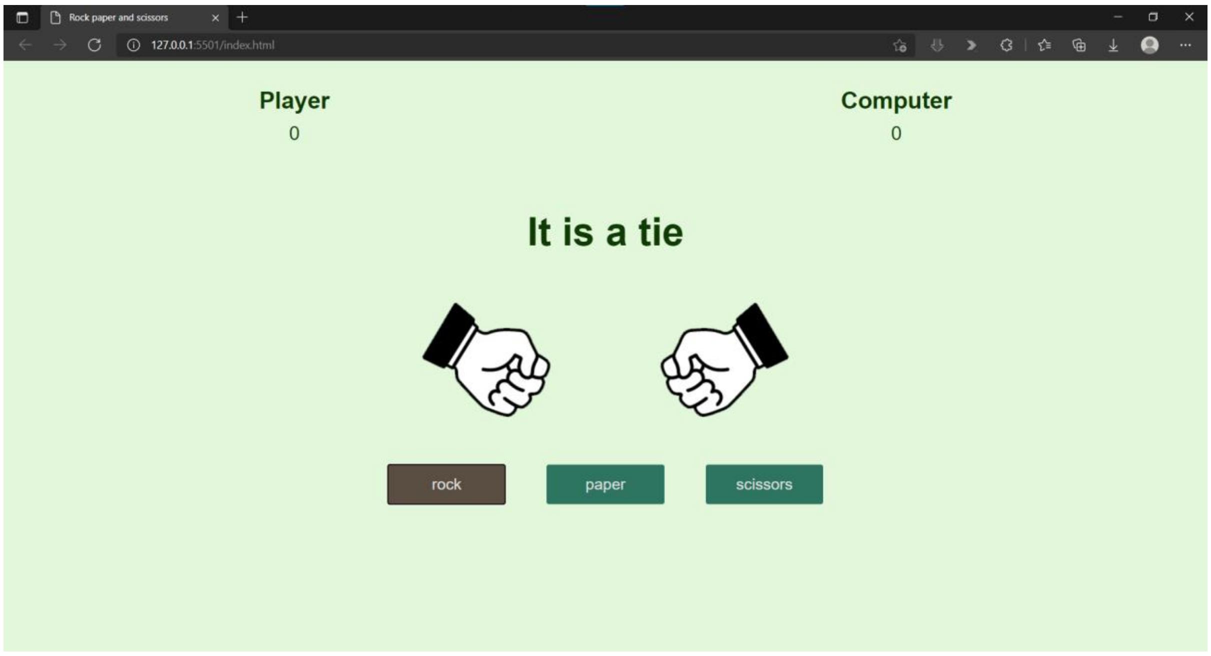
OUTPUT :

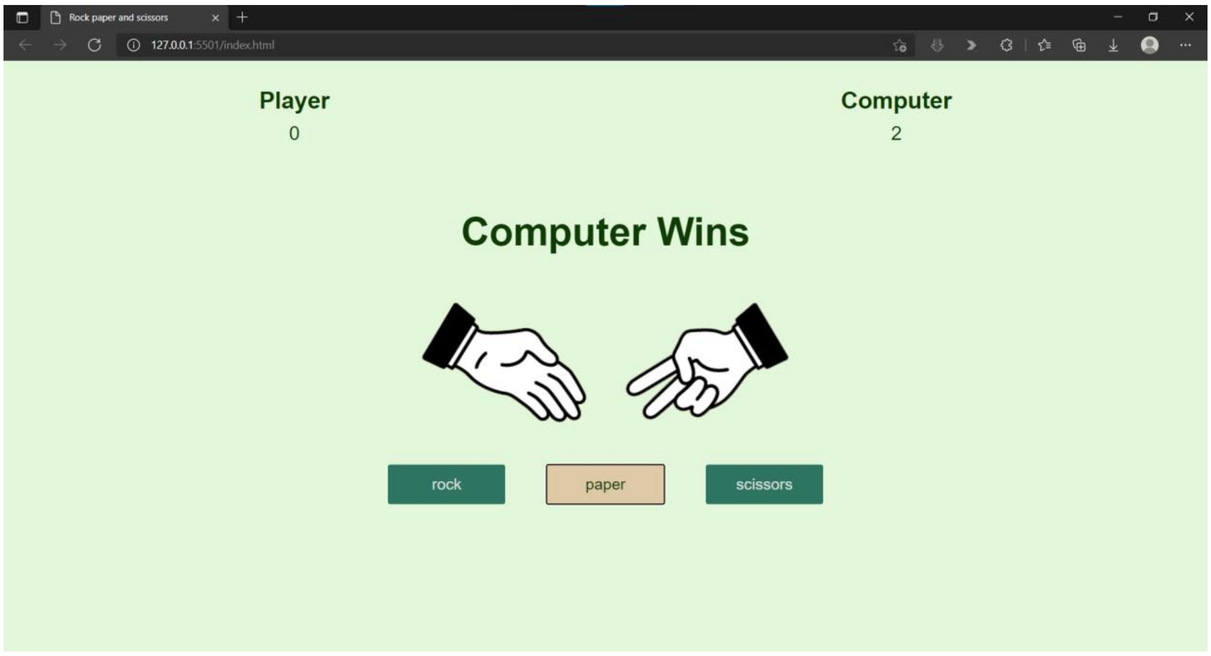
- **Home**

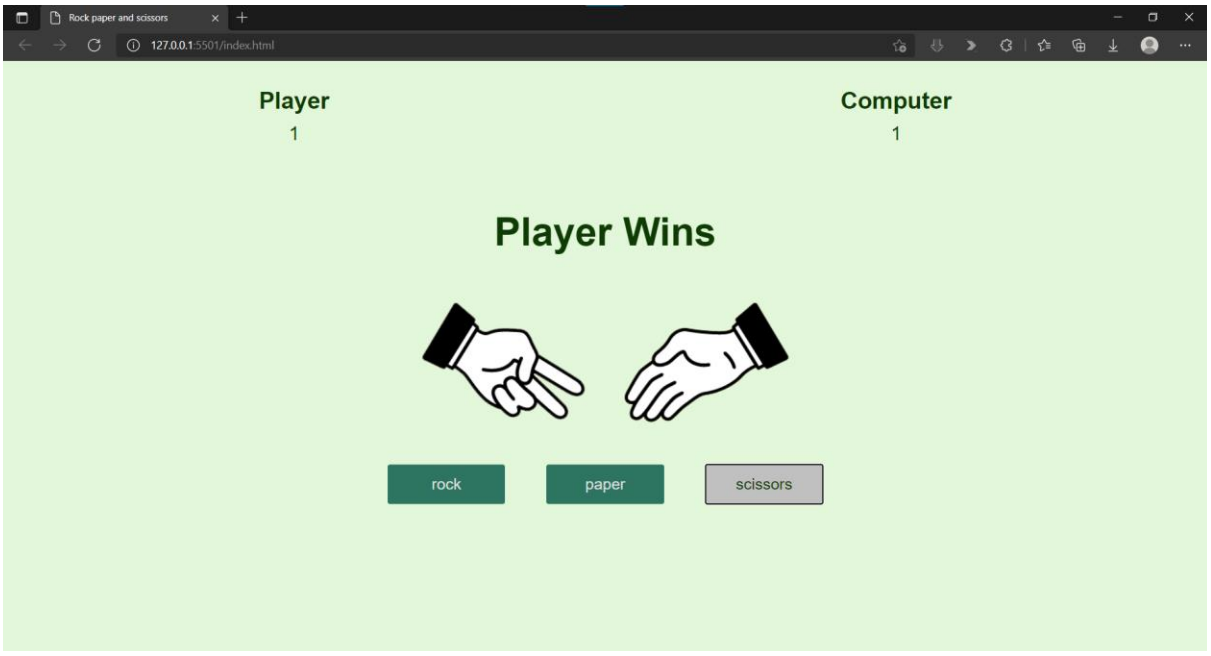


- **Game**




Rock paper and scissors
127.0.0.1:5501/index.html
Player
0
Computer
0
It is a tie
rock
paper
scissors


Rock paper and scissors
127.0.0.1:5501/index.html
Player
0
Computer
2
Computer Wins
rock
paper
scissors



CONCLUSION : Hence , by this experiment we have implemented the basic HTML and CSS by creating this Webpage using various HTML & CSS tags. Also we have implemented the game logic of rock-paper-scissors in JavaScript which has helped in understanding some JavaScript concepts.