



# RKDF UNIVERSITY

**Ranchi , Jharkhand**

## **INTERNSHIP REPORT**

Submitted by

**Harsh Vardhan Raj (001CSE22GT029)**

*in partial fulfillment for the award of the degree*

*of*

***Bachelor of Engineering***

*in*

***Computer Science and Engineering***

**RKDF University**

Ranchi , Jharkhand

An Autonomous Institution

MAY 2025

## **BONAFIDE CERTIFICATE**

Certified that this internship report

**“ Rock-Paper-Scissors “**

is the bonafide work of

**Harsh Vardhan Raj ( 001CSE22GT029)**

who carried out the project work under my supervision

SIGNATURE OF HOD

**Shubhangni Dey**

HEAD OF DEPARTMENT

Computer science and Engineering

**RKDF University , Ranchi , Jharkhand**

SIGNATURE OF SUPERVISOR

**Abhishek Kumar Singh**

SUPERVISOR

Computer science and Engineering

**RKDF University , Ranchi , Jharkhand**

Submitted for the Autonomous End Semester Examination Internship Project

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**



## ABSTRACT

The project titled "**Rock Paper Scissors Game Using HTML, CSS, and JavaScript**" is a browser-based implementation of the classic hand game played between two players. The purpose of this project is to demonstrate the application of core web technologies—**HTML for structure, CSS for styling, and JavaScript for interactivity and logic**—in building an engaging and responsive game.

This game allows the user to select one of three options—Rock, Paper, or Scissors—through a simple and intuitive interface. The computer randomly chooses its move, and the result of each round is displayed instantly based on the standard rules: Rock crushes Scissors, Scissors cuts Paper, and Paper covers Rock. The outcome may be a win, loss, or draw.

Through this project, key programming concepts such as **event handling, conditional logic, random number generation, and DOM manipulation** were explored and applied. The game is lightweight, interactive, and responsive, ensuring compatibility across modern web browsers.

This mini project serves as a fundamental exercise in front-end development and lays the groundwork for more advanced interactive applications. It not only strengthens programming logic but also enhances user interface design skills.

## **ACKNOWLEDGEMENT**

I would like to express my sincere gratitude to all those who contributed directly or indirectly to the successful completion of this project titled "**Rock Paper Scissors Game Using HTML, CSS, and JavaScript.**"

First and foremost, I am deeply thankful to **RKDF University** for providing me with the opportunity to undertake this mini internship project, which has greatly enhanced my practical knowledge and understanding of web development technologies.

I extend my heartfelt thanks to my respected mentor and guide, for their continuous support, encouragement, and insightful guidance throughout the course of this project. Their valuable suggestions and timely feedback were instrumental in shaping the direction and outcome of this work.

I also wish to express my gratitude to the **Department of Computer Science and Engineering**, RKDF University, Ranchi for fostering an academic environment that encourages innovation and hands-on learning.

A special thanks to my friends and classmates who offered motivation, constructive feedback, and technical support during the development of the project. Their collaboration helped me overcome several challenges during implementation.

This project has been a significant learning experience and has provided me with a strong foundation in building interactive web applications using core front-end technologies.

Harsh Vardhan Raj

(001CSE22GT029)

B.TECH (CSE)

Semester - VI

# Table of Contents

1. Introduction .....	7
2. Technologies used .....	8
3. System Design .....	9
3.1 System Architecture.....	9
3.2 Components and their Design .....	10
3.3 Flowchart of Execution. ....	10
3.4 Advantages of Design .....	11
3.5 Potential Enhancements .....	11
4. Implementation .....	12
4.1 HTML Implementation.....	12
4.2 CSS Implementation .....	13
4.3 JavaScript Implementation .....	14
4.4 Workflow Summary .....	14
5. Features .....	15
6. Output and Screenshots.....	18
7. Conclusion.....	26
8. References.....	27

# 1. INTRODUCTION

The "Rock Paper Scissors" game is a simple yet engaging digital implementation of the popular hand game commonly played between two players. In this project, the game is developed using core front-end web technologies—**HTML, CSS, and JavaScript**—and allows a single player to compete against the computer.

The game operates on basic rules where:

- **Rock crushes Scissors**
- **Scissors cut Paper**
- **Paper covers Rock**

The goal is to make a choice and see if it beats the randomly selected choice of the computer.

This project serves as an introductory-level exercise in **web development**, offering hands-on experience with essential concepts such as:

- Structuring content using **HTML**
- Styling elements for a better user interface using **CSS**
- Implementing interactive behavior and logic using **JavaScript**

The idea behind this project is not just to create a functioning game, but also to gain practical understanding of event handling, conditional statements, random number generation, and Document Object Model (DOM) manipulation—all of which are crucial skills for a front-end developer.

Moreover, this project demonstrates how basic programming skills can be used to create interactive and entertaining web applications. It also encourages creativity and experimentation, paving the way for building more advanced games and applications in the future.

## 2. Tools and Technologies Used

The development of the "**Rock Paper Scissors**" game involved the use of basic yet powerful front-end technologies and tools. Below is a detailed list of the tools and technologies utilized in building the project:

### 1. *HTML (HyperText Markup Language)*

- Used to create the **structure** of the web page.
- Defined buttons for user input and sections to display results.
- Provided the semantic layout for the game interface.

### 2. *CSS (Cascading Style Sheets)*

- Used to **style** the game elements including buttons, fonts, background, and layout.
- Enhanced the visual appeal and user experience.
- Ensured responsiveness and consistent design across different devices.

### 3. *JavaScript*

- Served as the **logic engine** of the game.
- Handled user interactions via event listeners.
- Implemented game logic to determine the winner based on user and computer choices.
- Used random number generation to simulate the computer's moves.

### 4. *Visual Studio Code*

- Used as the **primary code editor** for writing and testing the code.
- Provided useful extensions like Live Server for real-time preview and syntax highlighting.

### 5. *Web Browser (Google Chrome / Mozilla Firefox)*

- Used to **run and test** the game.
- Ensured that the game was responsive and compatible across modern browsers.

## 3. System Design

The **system design** of the "Rock Paper Scissors" game outlines how different components work together to create an interactive and fully functional browser-based game. It includes both **logical design** (how the system behaves) and **physical design** (how the interface looks and functions). This project adopts a **client-side architecture**, meaning everything runs within the web browser without the need for a server or database.

### 3.1. System Architecture

The system consists of **three main layers**:

*a) Presentation Layer (HTML & CSS)*

- Responsible for the layout and appearance of the game.
- Displays all visible elements such as buttons, headings, and the results.
- Uses **HTML** for structure and **CSS** for styling.

*b) Application Logic Layer (JavaScript)*

- Handles all decision-making and game logic.
- Processes user input, generates the computer's move, compares both, and determines the result.
- Updates the interface dynamically using the DOM.

*c) Interaction Layer (DOM Manipulation)*

- JavaScript uses the **Document Object Model (DOM)** to access and manipulate elements on the page.
- Displays outcomes like "You Win", "You Lose", or "It's a Draw" in real time.

### 3.2. Components and Their Design

#### a) User Interface

- **Buttons** for player input: Rock, Paper, Scissors.
- **Text display area** for showing:
  - User's choice
  - Computer's choice
  - Final outcome
- **Simple layout** using flexbox or centered content for responsiveness.

#### b) JavaScript Game Engine

- **Input Function:** Captures the player's selection when a button is clicked.
- **Randomizer Function:** Uses `Math.random()` to generate a random move for the computer.
- **Comparison Logic:**
  - Rock beats Scissors
  - Paper beats Rock
  - Scissors beats Paper
- **Output Function:** Updates the result section of the page with relevant messages.

### 3.3. Flow of Execution

Here's the step-by-step **flow of control**:

1. **Start:** Game loads in the browser.
2. **User Clicks Button:** Rock, Paper, or Scissors.
3. **JavaScript Triggers:**
  - Captures user's selection.
  - Randomly selects a move for the computer.
4. **Comparison Logic:**
  - Compares user and computer choices.
  - Determines win, lose, or draw.
5. **Display Result:**
  - Updates the result area with text: choices + outcome.
6. **End:** Waits for the next user input to repeat the cycle.

### 3.4 Advantages of the Design

- **Lightweight and Fast** – No server or database needed.
- **Highly Interactive** – Real-time response using JavaScript.
- **Modular Code** – Easy to modify or expand in the future.
- **Cross-browser Compatible** – Works in all modern web browsers.
- **Responsive Design** – Looks good on desktops, tablets, and phones.

### 3.5. Potential Enhancements

- Add **scoreboard tracking** across rounds.
- Include **animations or sound effects** for better user experience.
- Add **timer or round-based gameplay**.
- Implement **multiplayer mode** or store high scores using local storage.

## 4. Implementation

The Rock-paper-Scissors game is implemented using **HTML**, **CSS**, and **JavaScript**, each playing a critical role in developing the interactive web application.

### 4.1. HTML (*Structure of the Game*)

HTML was used to define the layout and user interface elements of the game. The page includes:

- A **title** or heading
- **Three buttons** for the player to choose Rock, Paper, or Scissors
- A **result display area** to show the choices and the outcome of each round

```
html
CopyEdit
<!DOCTYPE html>
<html>
<head>
    <title>Rock Paper Scissors</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <h1>Rock Paper Scissors</h1>
    <div class="buttons">
        <button onclick="play('rock')">Rock</button>
        <button onclick="play('paper')">Paper</button>
        <button
onclick="play('scissors')">Scissors</button>
    </div>
    <div id="result"></div>
    <script src="script.js"></script>
</body>
</html>
```

## *4.2. CSS (Styling and Layout)*

CSS was used to style the interface, making it visually appealing and user-friendly. The layout is kept simple and centered, with styled buttons and text.

```
css
CopyEdit
body {
    text-align: center;
    font-family: Arial, sans-serif;
    background-color: #f0f0f0;
}

h1 {
    margin-top: 30px;
    color: #333;
}

.buttons button {
    margin: 10px;
    padding: 15px 25px;
    font-size: 18px;
    cursor: pointer;
    border: none;
    background-color: #4CAF50;
    color: white;
    border-radius: 5px;
    transition: 0.3s;
}

.buttons button:hover {
    background-color: #45a049;
}

#result {
    margin-top: 20px;
    font-size: 22px;
    color: #333;
}
```

#### *4.3. JavaScript (Game Logic and Interactivity)*

JavaScript controls the entire logic of the game. It performs the following tasks:

- Detects which button the user clicked
- Randomly selects a move for the computer
- Compares both choices to determine the winner
- Displays the outcome dynamically

#### *4.4. Workflow Summary*

- The game starts when the user clicks any of the three buttons.
- JavaScript captures the user's input and randomly generates the computer's choice.
- The outcome is calculated based on comparison logic.
- The result is shown dynamically without reloading the page.

## 5. Features

The "**Rock Paper Scissors**" game project includes several well-implemented features that together provide a smooth, interactive, and user-friendly experience. Below is a detailed explanation of each major feature:

### *(i). Intuitive and Interactive User Interface*

- The game presents a clean, minimal, and user-friendly interface.
- Players can choose between **Rock**, **Paper**, or **Scissors** by clicking clearly labeled buttons.
- The UI responds instantly to clicks, enhancing engagement.
- The design is intuitive even for first-time users, requiring no instructions.

### *(ii). Real-Time Gameplay and Output*

- Once a player makes a selection, the game:
  - Randomly generates the computer's move.
  - Compares both choices.
  - Displays the result (Win, Lose, or Draw) immediately.
- The real-time nature of the result feedback makes the game dynamic and engaging.

### *(iii). Randomized Computer Move*

- The computer's move is generated using JavaScript's `Math.random()` function, ensuring fairness.
- Every round is independent and unpredictable, simulating a real opponent's choice.

#### *(iv). Dynamic Result Display*

- The outcome of each round is displayed clearly on the screen.
- It includes:
  - What the player chose
  - What the computer chose
  - The final result (e.g., "You Win!", "It's a Draw!")
- Results are dynamically inserted into the DOM without refreshing the page.

#### *(v). Simple Yet Effective Game Logic*

- The game follows the classic rules:
  - Rock beats Scissors
  - Scissors beats Paper
  - Paper beats Rock
- All scenarios are handled using conditional statements in JavaScript.
- The code is easy to understand and can be enhanced further.

#### *(vi). Responsive and Cross-Browser Compatible*

- The layout is designed using flexible CSS, making it compatible with various screen sizes (mobile, tablet, desktop).
- The game functions properly on all major web browsers like Chrome, Firefox, Edge, and Safari.

#### *(vii). No Page Reloads or External Dependencies*

- The game operates entirely on the client-side.
- JavaScript handles logic and updates the UI without any need for page reloads.
- No external libraries or back-end servers are required.

*(viii). Educational Value*

- The project is ideal for beginners learning front-end development.
- Demonstrates:
  - DOM manipulation
  - Event handling
  - Conditional logic
  - Basic UI design

*(ix). Lightweight and Fast*

- The code is optimized for performance.
- Very quick load time due to the minimal use of resources.
- Works well even on low-end devices or slow internet connections.

## 6. Outputs and Screenshots

### Code:

#### Index.html

```
<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Rock Paper Scissors</title>

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

<script src="script.js"></script>

</head>

<body>

<h1 id="opponent-score">0</h1>

<img id="opponent-choice">

<br>

<img id="your-choice">

<div id="choices">

</div>

<h1 id="your-score">0</h1>

</body>

</html>
```

## Style.css

```
body {  
    font-family: Arial, Helvetica, sans-serif;  
    text-align: center;  
}  
  
#opponent-choice {  
    width: 240px;  
    height: 240px;  
    /* background-color: cyan; */  
    margin-top: 10px;  
}  
  
#your-choice {  
    width: 240px;  
    height: 240px;  
    /* background-color: yellow; */  
    margin-top: 10px;  
}  
  
#choices {  
    width: 240px;  
    height: 80px;  
    /* background-color: green; */
```

```
margin: 0 auto;  
margin-top: 10px;  
}  
  
#choices img {  
width: 80px;  
height: 80px;  
}
```

## Script.js

```
var you;  
  
var yourScore = 0;  
  
var opponent;  
  
var opponentScore = 0;  
  
var choices = ["rock", "paper", "scissors"];  
  
window.onload = function() {  
  
    for (let i = 0; i < 3; i++) {  
  
        //   
  
        let choice = document.createElement("img");  
  
        choice.id = choices[i];  
  
        choice.src = choices[i] + ".png";  
  
        choice.addEventListener("click", selectChoice);  
  
        document.getElementById("choices").append(choice);  
  
    }  
  
}  
  
function selectChoice() {  
  
    you = this.id;  
  
    document.getElementById("your-choice").src = you + ".png";  
  
    //random for opponent  
  
    opponent = choices[Math.floor(Math.random() * 3)]; //0- .999999 * 3 = 0-2.99999
```

```
document.getElementById("opponent-choice").src = opponent + ".png";  
  
//check for winner  
  
if (you == opponent) {  
  
    yourScore += 1;  
  
    opponentScore += 1;  
  
}  
  
else {  
  
    if (you == "rock") {  
  
        if (opponent == "scissors") {  
  
            yourScore += 1;  
  
        }  
  
        else if (opponent == "paper") {  
  
            opponentScore += 1;  
  
        }  
  
    }  
  
    else if (you == "scissors") {  
  
        if (opponent == "paper") {  
  
            yourScore += 1;  
  
        }  
  
        else if (opponent == "rock") {  
  
            opponentScore += 1;  
  
        }  
  
    }  
  
}
```

```
}

}

else if (you == "paper") {

if (opponent == "rock") {

yourScore += 1;

}

else if (opponent == "scissors") {

opponentScore += 1;

}

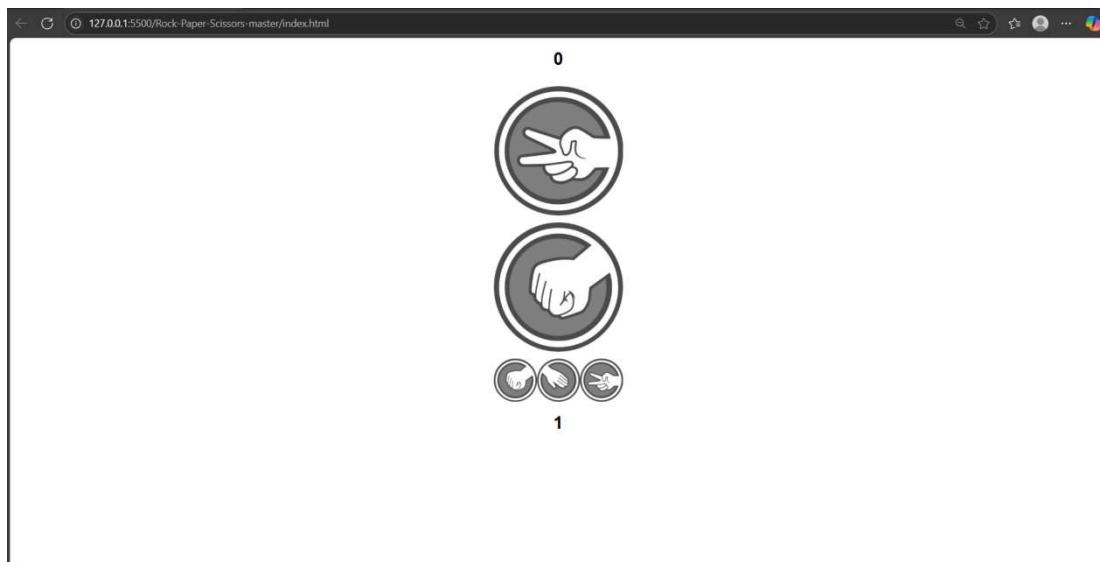
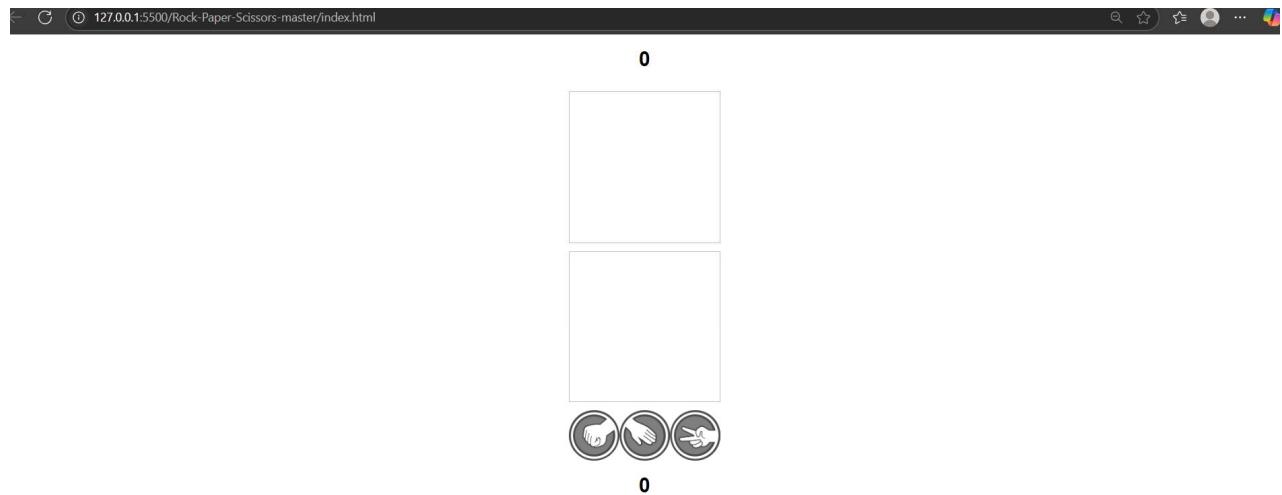
}

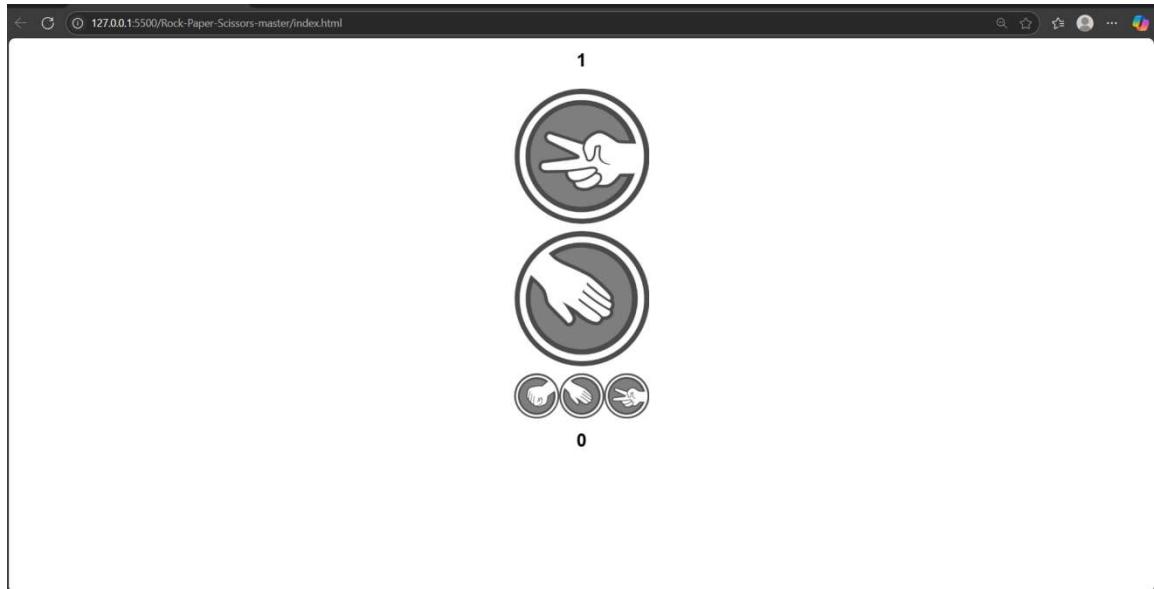
document.getElementById("your-score").innerText = yourScore;

document.getElementById("opponent-score").innerText = opponentScore;

}
```

## Outputs :





## 7. Conclusion

The development of the "**Rock Paper Scissors**" game using HTML, CSS, and JavaScript has been a highly educational and rewarding experience. This project not only helped in understanding the fundamental principles of front-end web development but also provided practical exposure to building an interactive and engaging browser-based game from scratch.

Through the implementation of this project, various essential concepts such as **structuring a web page with HTML**, **styling with CSS**, and **adding logic and interactivity using JavaScript** were explored in depth. The project required a good understanding of **event listeners**, **random number generation**, **conditional statements**, and **DOM manipulation** to process user input and dynamically update the results on the screen.

One of the most valuable outcomes of this project was learning how to combine different technologies to work together seamlessly. HTML provided the skeleton of the game interface, CSS enhanced its look and feel, and JavaScript brought the game to life through real-time interactivity. This full integration showcases the power of front-end development in creating interactive web applications.

Furthermore, the project emphasizes **clean and maintainable code** by following structured logic and user-friendly design principles. The simplicity of the game logic made it easier to implement and understand, yet it also presented opportunities for future improvements, such as adding score tracking, animations, sound effects, or even multiplayer capabilities.

This project has also strengthened debugging and problem-solving skills, which are critical in real-world software development. Working through challenges during development helped reinforce logical thinking and attention to detail.

In conclusion, the "Rock Paper Scissors" game is more than just a fun mini project; it is a foundational step toward mastering web development. It provides a practical understanding of how user interfaces and game logic work together, and lays the groundwork for building more advanced and dynamic web-based applications in the future.

## 8. References

- **MDN Web Docs (Mozilla Developer Network)**

<https://developer.mozilla.org/>

- Used for understanding HTML elements, CSS properties, and JavaScript functions.

- **W3Schools**

<https://www.w3schools.com/>

- Referred for syntax clarification, examples, and tutorials on front-end web development.

- **Stack Overflow**

<https://stackoverflow.com/>

- Helped in resolving coding issues and understanding best practices in JavaScript.

- **GeeksforGeeks – JavaScript Tutorials**

<https://www.geeksforgeeks.org/javascript/>

- Referred for logic building, random number generation, and event handling.

- **CSS-Tricks**

<https://css-tricks.com/>

- Used for styling tips and layout ideas.

- **YouTube Tutorials**

- Various tutorial videos were referred to for visual guidance on building simple JavaScript games.

- **Github Link :** <https://github.com/Harshvardhan56/Rock-paper-scissors.git>