

A
MAJOR PROJECT REPORT
on
EPIC ROLL SAGA

Submitted in partial fulfillment to the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY
in
INFORMATION TECHNOLOGY

Submitted by

A.RIA	-	22UP1A1201
D.SRAVANTHI	-	22UP1A1223
D.TRISHITHA	-	22UP1A1224
M.CHARITHA	-	22UP1A1236
M.PRASANNA	-	22UP1A1239
T.RADHIKA	-	22UP1A1258

Under the Guidance of
P. Amareshwari
Associate Professor



DEPARTMENT OF INFORMATION TECHNOLOGY
VIGNAN'S INSTITUTE OF MANAGEMENT AND TECHNOLOGY
FOR WOMEN

Accredited to NAAC + & NBA (CSE & ECE)

(Affiliated to Jawaharlal Nehru Technological University (Hyderabad) Kondapur (Village),
Ghatkesar (Mandal), Medchal (Dist.), Telangana. Pincode-501301

2020-2024



VIGNAN'S INSTITUTE OF MANAGEMENT AND TECHNOLOGY FOR WOMEN

Sponsored by Lavu Educational Society
Affiliated to JNTU, Hyderabad & Approved by AICTE, New Delhi
Kondapur (V), Ghatkesar (M), Medchal-Malkajgiri (Dist.)-501301, Ph: 9652910002/3



DEPARTMENT OF INFORMATION

CERTIFICATE

This is to certify that project work entitled "**EPIC ROLL SAGA**" submitted by **A.Ria** (22UP1A1201), **D.Sravanthi** (22UP1A1223), **D.Trishitha** (22UP1A1224), **M.Charitha**(22UP1A1236), **M.Prasanna** (22UP1A1239), **T.Radhika** (22UP1A1258) in the partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Information Technology** **VIGNAN'S INSTITUTE OF MANAGEMENT AND TECHNOLOGY FOR WOMEN** is a record of bonafide work carried by them under my guidance and supervision. The results embodied in this project report have not been submitted to any other University or institute for the award of any degree.

PROJECT GUIDE

P.Amareshwari

(Associate Professor)

HEAD OF DEPARTMENT

Dr. B. PHIJIK

Department of IT

(External Examiner)



VIGNAN'S INSTITUTE OF MANAGEMENT AND TECHNOLOGY FOR WOMEN

Sponsored by Lavu Educational Society

Affiliated to JNTU, Hyderabad & Approved by AICTE, New Delhi

Kondapur (V), Ghatkesar (M), Medchal-Malkajgiri (Dist.)-501301, Ph: 9652910002/3



DEPARTMENT OF INFORMATION

DECLARATION

We hereby declare that the Project entitled "**“EPIC ROLL SAGA”**" is a bonafide work duly completed by us. It does not contain any part of the project or that is submitted by any other candidate to this or another Institute of the University. All such materials that have been obtained from other sources have been duly acknowledged.

A.RIA	-	22UP1A1201
D.SRAVANTHI	-	22UP1A1223
D.TRISHITHA	-	22UP1A1224
M.CHARITHA	-	22UP1A1236
M.PRASANNA	-	22UP1A1239
T.RADHIKA	-	22UP1A1258



VIGNAN'S INSTITUTE OF MANAGEMENT AND TECHNOLOGY FOR WOMEN

Sponsored by Lavu Educational Society

Affiliated to JNTU, Hyderabad & Approved by AICTE, New Delhi
Kondapur (V), Ghatkesar (M), Medchal-Malkajgiri (Dist.)-501301, Ph: 9652910002/3



ACKNOWLEDGEMENT

We would like to express sincere gratitude to **Dr. G. APPARAO NAIDU**, Principal, Vignan's Institute of Management and Technology for Women for his timely suggestions which helped us to complete the project in time.

We would also like to thank our sir **Dr. B. PHILJIK, Head of the Department, Information Technology**, for providing us with consultant encouragement and resources which helped us to complete the project in time.

We would also like to thank our madam **P.Amareshwari, Associate Professor, Internal Guide, Department of IT**, for providing us with constant encouragement and resources which helped us to complete the project in time with her valuable suggestions throughout the project. We are indebted to her for the opportunity given to work under her guidance. Our sincere thanks to all the teaching and non-teaching staff of the Department of Information Technology for their support throughout our project work.

A.RIA	-	22UP1A1201
D.SRAVANTHI	-	22UP1A1223
D.TRISHITHA	-	22UP1A1224
M.CHARITHA	-	22UP1A1236
M.PRASANNA	-	22UP1A1239
T.RADHIKA	-	22UP1A1258

EPIC ROLL SAGA: A WEB-BASED ENTERTAINMENT EXPERIENCE

ABSTRACT

The Dice Roll Game is a dynamic and interactive web application developed using HTML, JavaScript, and CSS. The game offers users an engaging experience where they can simulate rolling a pair of dice and compete for high scores.

Implemented with HTML for the structure, JavaScript for the functionality, and CSS for styling, the game provides a seamless user interface with visually appealing design elements. Through JavaScript, the game logic manages the dice rolling process, calculates scores based on the outcome, and updates the user interface accordingly.

• REQUIREMENTS:

- HTML
- CSS
- JAVASCRIPT
- DICE (PICTURES)

In conclusion, the Dice Roll Game offers users an entertaining and interactive experience through a combination of HTML, JavaScript, and CSS technologies, providing hours of fun for players of all ages.

• PROJECT BY:

- ❖ A.RIA-(22UP1A1201)
- ❖ D.SRAVANTHI-(22UP1A1223)
- ❖ D.TRISHITHA-(22UP1A1224)
- ❖ M.CHARITHA-(22UP1A1236)
- ❖ M.PRASANNA-(22UP1A1239)
- ❖ T.RADHIKA-(22UP1A1258)

INDEX

S.No	TITLE	Pg.No
1	Introduction	7
2	System analysis	9
3	Hardware & Software Requirements	14
4	System Design (Architecture)	15
5	Code	20
6	Testing	25
7	Process of execution	27
8	Conclusion	30

Introduction

The Dice Game is a classic game of chance involving the rolling of two dice. The primary objective of this project is to create a web-based version of the Dice Game, leveraging the capabilities of HTML, CSS, JavaScript, and Java. The game aims to provide a visually appealing and interactive experience to users. Players can roll the dice by clicking a button, and the result of the dice roll is displayed on the screen. The project encompasses both front-end and back-end development, ensuring a comprehensive understanding of web application development.

The front-end component will be developed using HTML for the structure, CSS for styling, and JavaScript for dynamic interactions. The back-end will be implemented using Java, which will handle the game logic and potentially store player scores and game histories if connected to a database. This combination of technologies provides a robust and scalable framework for the Dice Game.

The significance of this project lies in its ability to demonstrate the integration of different web technologies to build a functional and interactive web application. It also serves as an educational tool, highlighting the importance of each technology in web development and how they work together to create a seamless user experience.

➤ Scope

The scope of the Dice Game project encompasses the development of an interactive and user-friendly web-based game using HTML, CSS, JavaScript, and Java. This project aims to deliver a complete gaming experience by integrating both front-end and back-end technologies. The front-end will be responsible for presenting the game interface and handling user interactions, while the back-end will manage the game logic and potential data storage. This project will cover:

- Designing a responsive and visually appealing user interface.
- Implementing game mechanics using JavaScript for real-time interactivity.
- Developing a Java-based backend to handle game logic and data management.

- Ensuring cross-platform compatibility and accessibility across different devices and browsers.
- Integrating a database to store player scores and game history.

➤ Purpose

The primary purpose of the Dice Game project is to provide an engaging and interactive gaming experience that can be accessed via a web browser. The project serves as a practical application of web development skills, combining front-end and back-end technologies to create a cohesive and functional web application. The specific objectives include:

- **Educational Value:** Demonstrating the integration of HTML, CSS, JavaScript, and Java in a real-world project, providing a learning platform for developers.
- **Entertainment:** Creating a fun and interactive game that users can enjoy, enhancing their online experience.
- **Skill Development:** Enhancing the developer's skills in web development, user interface design, and server-side programming.
- **Feature Expansion:** Offering additional features such as score tracking and game history to enrich the gaming experience and provide a sense of progression.

➤ Need for the System

There is a need for this system for several reasons:

- **Engaging User Experience :** Many existing dice games lack engaging and interactive interfaces. This project aims to fill that gap by providing a visually appealing and responsive game that users can enjoy on any device.
- **Learning Resource:** For developers and students, this project serves as a comprehensive example of how to build a full-stack web application. It demonstrates the use of multiple technologies and how they can work together to create a functional product.
- **Accessibility :** Ensuring that the game is accessible across different devices and browsers meets the growing demand for cross-platform compatibility

in web applications. This makes the game available to a wider audience, increasing its usability and reach.

- **Feature-Rich Experience:** By including features like score tracking and game history, the project offers more than just a basic game. These additional features enhance the overall user experience and provide users with reasons to return and play again.
- **Modern Design:** The project will utilize modern web design principles, making the game not only functional but also visually appealing. This is important in today's web landscape where user expectations for design and usability are high.

By addressing these needs, the Dice Game project aims to deliver a robust, enjoyable, and educational product that benefits both users and developers.

System Analysis

➤ EXISTING SYSTEM:

- The existing dice game system is a digital implementation of a classic dice game, designed primarily for casual gaming and entertainment purposes.
- It allows two players to participate in rounds of rolling dice to achieve various objectives and accumulate points.
- The system is currently accessible through a web browser, offering a straightforward interface for users to interact with.

➤ Features:

Dice Rolling Mechanism:

- Players can roll virtual dice with a click of a button.
- The system generates random outcomes based on dice mechanics (e.g., 6-sided dice).

Gameplay Flow:

- Players simultaneously roll dice and perform actions.
- Rules and objectives specific to the game are implemented (e.g., achieving higher numbers on the dice).

Player Interaction:

- Support for 2 players.
- Interaction facilitated through a centralized game board or interface.

Scoring and Progress Tracking:

- Automated winning message based on predefined game rules.

User Interface:

- Simple and intuitive design with basic controls for dice rolling and game navigation.
- Minimalist approach to maintain focus on gameplay experience.

Technology Stack:

- **Backend:** Developed using Node.js for server-side logic.
- **Frontend:** HTML, CSS, and JavaScript for client-side rendering and interaction.
- **Data Storage:** MongoDB used for persistent data storage (game sessions, player scores).

Security Measures:

- Basic security measures implemented (e.g., data encryption, user authentication for accessing game sessions).

➤ **Limitations:**

User Experience:

- Limited visual and interactive elements in the user interface.
- Potential for monotony in extended gameplay sessions due to simplistic design.

Feature Set:

- Lack of advanced gameplay features that could enhance engagement (e.g., Creating a score table to track players progress).

Maintenance and Support:

- Regular updates for bug fixes and minor improvements.
- Minimal ongoing support required due to stable performance and low complexity of the system.

➤ **Conclusion:**

The existing dice game system serves its purpose as a basic, functional platform for playing the game online. While it meets fundamental requirements for gameplay and user interaction, there are opportunities to enhance both the technical capabilities and user experience to cater to a broader audience and improve engagement levels.

➤ **PROPOSED SYSTEM:**

➤ **Goals and Objectives:**

The proposed system aims to enhance the existing dice game by introducing new features, improving user experience, and implementing a more robust

architecture to support scalability and future expansion. Key objectives include:

- **Enhanced Gameplay:** Introduce new game modes, interactive elements, and challenges to keep players engaged.
- **Improved User Interface:** Revamp the UI to be more visually appealing, intuitive, and responsive across different devices.
- **Scalability:** Design a scalable architecture that can handle increased traffic .
- **Advanced Features:** Implement features such as real-time social integrations, and customizable game settings by adding a score table.
- **Security:** Strengthen data security measures to protect user information and game integrity.

➤ System Requirements:

- **Functional Requirements:**

- Real-time dice rolling mechanics with animations.
 - Enhanced scoring algorithms to support game rules.

- **Non-functional Requirements:**

- **Performance:** Response times optimized for smooth gameplay experience.
 - **Reliability:** Minimal downtime and robust error handling.
 - **Compatibility:** Support for various web browsers.
 - **Data Integrity:** Backup and recovery mechanisms to ensure data safety.
 - **Accessibility:** Compliance with accessibility standards for inclusive user experience.

➤ User Interface Design:

- **Visual Design:** Adopt a modern and visually appealing design language with engaging animations and intuitive controls.
- **Responsive Layout:** Ensure compatibility across desktop and tablet for seamless gameplay.
- **Accessibility:** Implement features like high contrast mode and screen reader compatibility for accessibility compliance.

➤ Data Management:

- **Security:** Implement encryption for sensitive data transmission and storage, secure authentication mechanisms, and regular security audits.

➤ Implementation Plan:

- **Phase 1 (Development):** Build core game mechanics, UI/UX design, and basic multiplayer functionality.
- **Phase 2 (Testing):** Conduct rigorous testing and optimization for performance and security.
- **Phase 3 (Deployment):** Deploy the system in a staged manner, starting with a limited release and scaling up based on feedback and performance metrics.

➤ Maintenance and Support Strategy:

- **Continuous Updates:** Release regular updates for bug fixes, feature enhancements, and security patches.
- **Monitoring and Analytics:** Implement monitoring tools to track performance metrics, user engagement, and gameplay analytics.

➤ Conclusion:

- The proposed system for the dice game aims to elevate the gaming experience through advanced features, improved usability, and a scalable architecture.
- By addressing the limitations of the existing system and leveraging modern technologies, the new system is poised to attract a larger user base and enhance overall user satisfaction.

Hardware & Software Requirements

HARDWARE

Processor (CPU):

- Minimum: Dual-core processor, such as Intel Core i3 or AMD equivalent
- Recommended: Quad-core processor, such as Intel Core i5 or AMD Ryzen 5

■ **Memory (RAM):**

- Minimum: 4 GB RAM
- Recommended: 8 GB RAM

■ **Graphics Card (GPU):**

- Minimum: Integrated graphics (e.g., Intel HD Graphics 4000 or equivalent)
- Recommended: Dedicated graphics card with at least 2 GB VRAM

■ **Storage:**

- Minimum: 2 GB available space
- Recommended: 5 GB available space

■ **Operating System:**

- Windows 7/8/10 (64-bit)
- Linux distributions with recent kernel and OpenGL support

■ **Input Devices:**

- Keyboard and mouse

■ **Internet Connection:**

- Required for online multiplayer modes or updates

SOFTWARE

Operating system : Windows 11

Coding Language : JavaScript

Designing : Html, css, javascript

SYSTEM ARCHITECTURE

Creating a system architecture for a dice game involves designing the structure of the software components and their interactions. Here's a breakdown of the system architecture:

System Architecture Overview

➤ Frontend:

- HTML: Structure of the web page.
- CSS: Styling for visual presentation.
- JavaScript: Game logic and user interaction.

➤ Testing Framework (Optional):

- Tools like ‘Jest’ for unit testing JavaScript functions.
- ‘Selenium’ or ‘Puppeteer’ for UI testing.

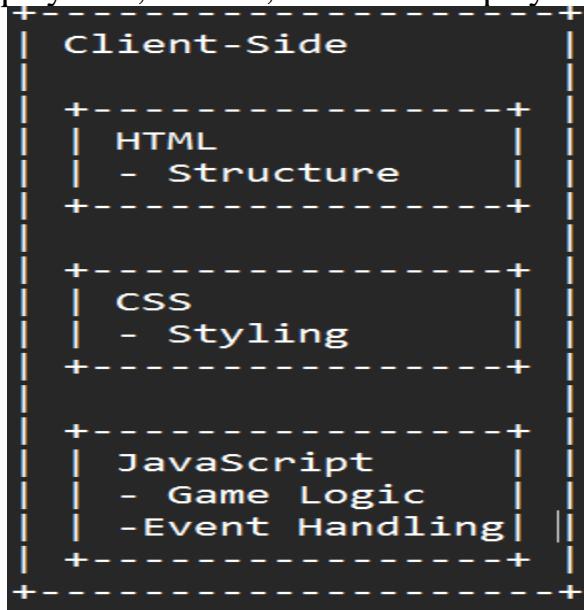
Development and Deployment Tools:

- Web Server: For hosting the game locally or on a web server.

Detailed Component Breakdown

HTML (Structure)

The HTML file sets up the basic structure and elements of the dice game. This includes the dice display area, buttons, and result display.



Key Elements:

- dice-container: A container for dice.
- roll-button: A button to trigger the dice roll.
- result: Displays the result.

CSS (Styling)

- The CSS file styles the HTML elements to enhance the visual presentation.

Key Elements:

- dice: Styles for dice display.
- roll-button: Styles for the roll button.
- result: Styles for result display.

JavaScript (Game Logic and Event Handling)

- The JavaScript file handles the game logic, including rolling the dice and updating the UI.

Key Elements:

- Function rollDie: Generates a random number between 1 and 6.
- Function rollDice: Rolls two dice, updates the UI with the dice values and their sum.
- Event Listener: Handles the button click to roll dice.

Development and Deployment

- Code Editor: Use a code editor like VS Code to develop the game.
- Web Server: For local testing, open the HTML file directly in a browser or use a local server (e.g., http-server or a simple Python server).

Deploying on a Web Server:

- Upload the HTML, CSS, and JavaScript files to your web server.
- Ensure the files are accessible and linked correctly.

Summary

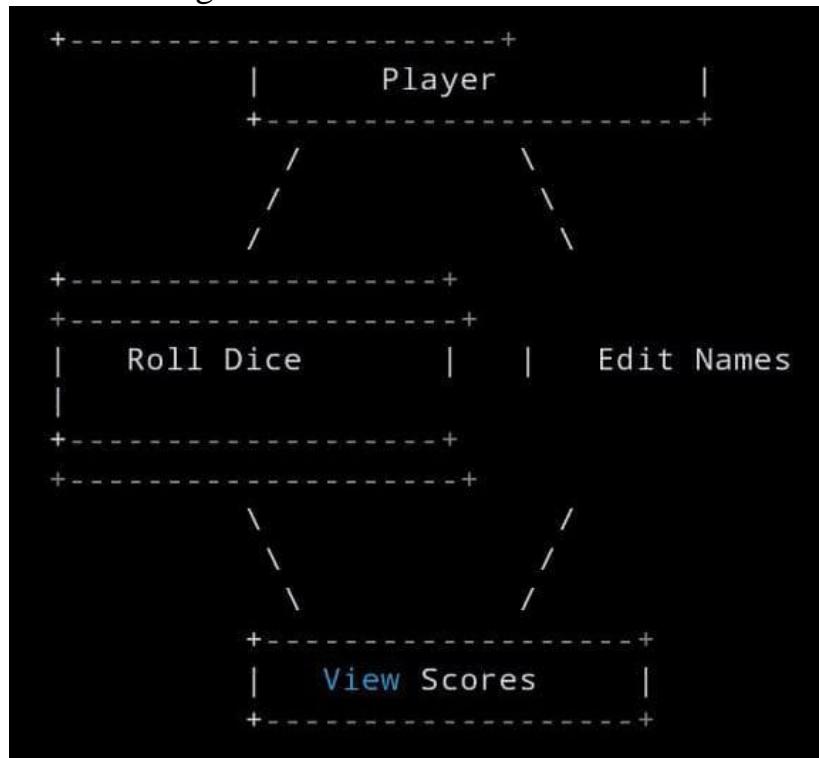
- This architecture provides a straightforward and modular approach to building and testing a dice game.
- It separates the concerns of structure, style, and logic, making it easier to maintain and test.

Why Do We Use UML in projects?

- As the strategic value of software increases for many companies, the industry looks for techniques to automate the production of software and to improve quality and reduce cost and time-to-market.
- These techniques include component technology, visual programming, patterns and frameworks.
- Businesses also seek techniques to manage the complexity of systems as they increase in scope and scale.
- In particular, they recognize the need to solve recurring architectural problems, such as physical distribution, concurrency, replication, security, load balancing and fault tolerance.
- Additionally, the development for the World Wide Web, while making some things simpler, has exacerbated these architectural problems.
- The Unified Modeling Language (UML) was designed to respond to these needs.
- Simply, Systems design refers to the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements which can be done easily through UML diagrams.

We can display the complete game play with the help of 3 different types of uml diagrams:

1. Use Case UML Diagram :



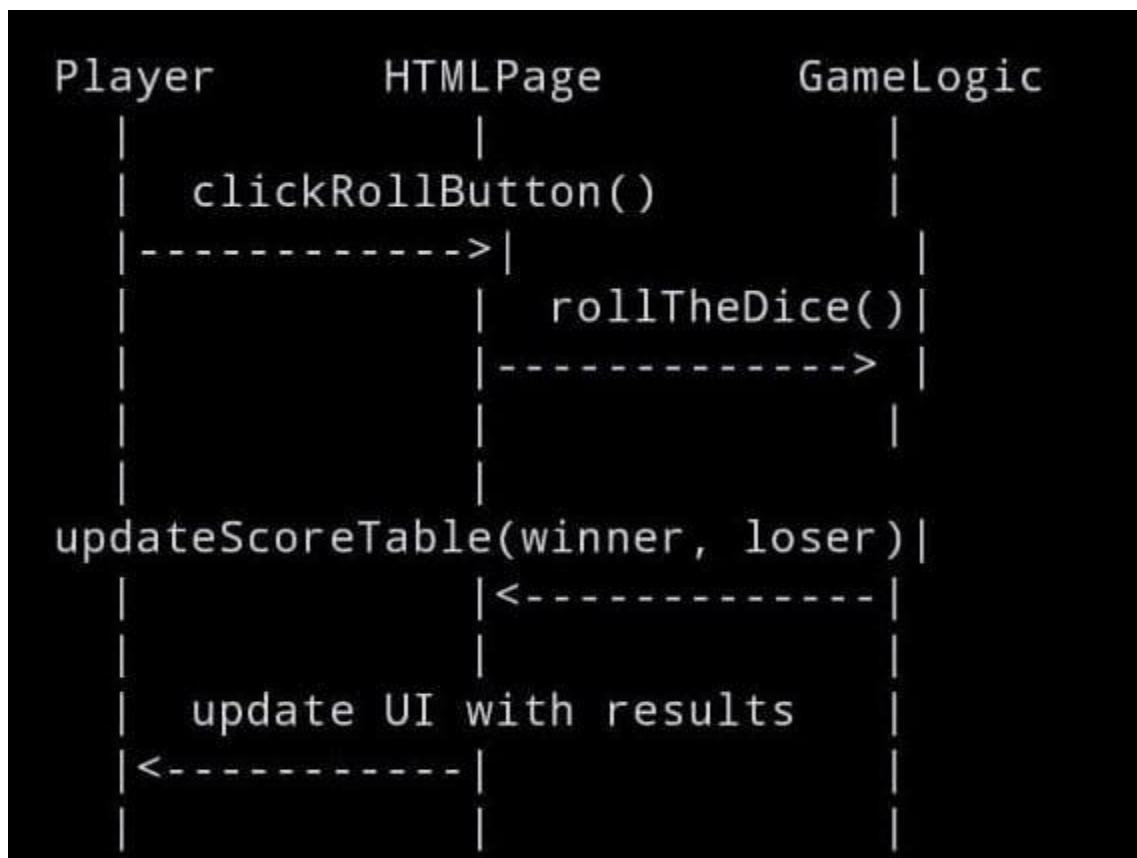
2. Class UML Diagram:

```

+-----+
+-----+
|   HTMLPage      |<-----+
| CSSStyles       |
+-----+
+-----+
| - container    |           | - 
| styles          |           |
| - dice          |           |
+-----+
| - buttons       |           | + 
applyStyles()|
| - scoreTable   |
+-----+
+-----+
| + render()     |
+-----+
|
|
|
v
+-----+
|   GameLogic    |
+-----+
| - player1      |
| - player2      |
| - gameCount    |
+-----+
| + editNames()  |
| + rollTheDice()|
| + updateScoreTable(winner: string,
loser: string)|
+-----+

```

3. Sequence Diagram:



Explanation

- **Use Case Diagram:** Shows the interactions between the Player and the system, highlighting the main functionalities: Roll Dice, Edit Names, and View Scores.
- **Class Diagram:** Displays the structure of the application, with classes representing HTML elements, CSS styles, and game logic, and their relationships
- **Sequence Diagram:** Illustrates the sequence of method calls when a player clicks the Roll Dice button, showing interactions between the Player, HTMLPage, and GameLogic.

CODE

//HTML CODE:

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

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <h1>EPIC ROLL SAGA</h1>

<style>
    .container {
        width: 70%;
        margin: auto;
        text-align: center;
    }
    .dice {
```

```
    text-align: center;
    display: inline-block;
    margin: 10px;
}

body {
    background-color: slateblue;
    margin: 0;
}

h1 {
    margin: 30px;
    font-family: "Lobster", cursive;
    text-shadow: 5px 0 #232931;
    font-size: 4.5rem;
    color: black;
    text-align: center;
}
h2 {
    margin: 30px;
    font-family: "Lobster", cursive;
    text-shadow: 5px 0 #232931;
    font-size: 4.5rem;
    color: tomato;
    text-align: center;
}

p {
    font-size: 2rem;
    color: pink;
    font-family: "Indie Flower", cursive;
}

img {
    width: 100%;
}

.bottom {
    padding-top: 30px;
}

.butn {
```

```
background: yellow;  
font-family: "Indie Flower", cursive;  
border-radius: 7px;  
color: blue;  
font-size: 30px;  
padding: 16px 25px 16px 25px;  
text-decoration: none;  
}  
  
.butn:hover {  
background: Gray;  
text-decoration: none;  
}
```

//Table creation

```
table {  
width: 100%;  
margin: 50px 0;  
border-collapse: collapse;  
}  
  
table, th, td {  
border: 5px solid maroon;  
color: black;  
font-family: "Lobster", cursive;  
}  
  
th, td {  
padding: 10px;  
text-align: center;  
}  
</style>  
</head>
```

//HTML Code

```
<body>  
<div class="container">  
  <h2>Let's Play</h2>  
  
  <div class="dice">  
    <p class="Player1">Player 1</p>  
    
```

```

</div>

<div class="dice">
    <p class="Player2">Player 2</p>
    
</div>
</div>

<div class="container bottom">
    <button type="button" class="butn" onClick="rollTheDice()">
        Roll the Dice
    </button>
</div>
<div class="container bottom">
    <button type="button" class="butn" onClick="editNames()">
        Edit Names
    </button>
</div>

<div class="container">
    //Table columns
    <table id="scoreTable">
        <thead>
            <tr>
                <th>Game</th>
                <th>Winner</th>
                <th>Loser</th>
            </tr>
        </thead>
        <tbody>
            <!-- Score rows will be dynamically added here -->
        </tbody>
    </table>
</div>

```

//JavaScript Code

```

<script>
    //Player name
    var player1 = "Player 1";
    var player2 = "Player 2";
    var gameCount = 0;

```

```

//Function to change the player name
function editNames() {
    player1 = prompt("Change Player1 name") || player1;
    player2 = prompt("Change player2 name") || player2;

    document.querySelector("p.Player1").innerHTML = player1;
    document.querySelector("p.Player2").innerHTML = player2;
}

//Function to roll the dice
function rollTheDice() {
    setTimeout(function () {
        var randomNumber1 = Math.floor(Math.random() * 6) + 1;
        var randomNumber2 = Math.floor(Math.random() * 6) + 1;

        document.querySelector(".img1").setAttribute("src",
            "dice" + randomNumber1 + ".png");

        document.querySelector(".img2").setAttribute("src",
            "dice" + randomNumber2 + ".png");

        var winner, loser;
        if (randomNumber1 === randomNumber2) {
            document.querySelector("h2").innerHTML = "Draw!";
            winner = "Draw";
            loser = "Draw";
        } else if (randomNumber1 < randomNumber2) {
            document.querySelector("h2").innerHTML = player2 + " WINS!";
            winner = player2;
            loser = player1;
        } else {
            document.querySelector("h2").innerHTML = player1 + " WINS!";
            winner = player1;
            loser = player2;
        }

        updateScoreTable(winner, loser);
    }, 2500);
}

function updateScoreTable(winner, loser) {
    gameCount++;
    var table =

```

```

document.getElementById("scoreTable").getElementsByTagName('tbody')[0];
    var newRow = table.insertRow();
    var cell1 = newRow.insertCell(0);
    var cell2 = newRow.insertCell(1);
    var cell3 = newRow.insertCell(2);

    cell1.innerHTML = gameCount;
    cell2.innerHTML = winner;
    cell3.innerHTML = loser;
}
</script>
</body>

</html>

```

TESTING

To ensure that our Dice Game project is thoroughly tested, we perform a variety of tests.

➤ **Unit Testing Purpose:**

- To verify that individual units or components of the code (functions, methods) work as expected.
- **What to Test:**
 - **editNames() function:** Ensure it correctly changes player names.
 - **rollTheDice() function:** Check if it generates valid random numbers for dice rolls and updates the UI appropriately.
 - **updateScoreTable(winner, loser) function:** Verify it correctly updates the score table with the results of each game.

➤ **Integration Testing Purpose:**

- To ensure that different services used by your application work well together.
- **What to Test:**
 - Ensure that changing player names using editNames() and then rolling the dice works seamlessly.

- Verify that rolling the dice updates the score table correctly.

➤ User Interface Testing Purpose:

- To ensure that the user interface behaves as expected.

What to Test:

- Button clicks (Roll the Dice, Edit Names) work as expected. Correct UI updates (dice images, winner display, score table updates).

➤ Functional Testing Purpose:

- To verify that the application functions as expected from the user's perspective.

● What to Test:

- The overall gameplay: Start the game, roll the dice, and update the score.
- Editing player names and ensuring they reflect correctly during the game.

➤ Regression Testing Purpose:

- To ensure that new code changes do not adversely affect the existing functionality.

● What to Test:

- Ensure that after implementing new features or bug fixes, the core functionality of rolling the dice and updating scores still works.

By performing these five types of testing, you can ensure that your Dice Game project is robust, user-friendly, and free from significant defects.

PROCESS OF EXECUTION

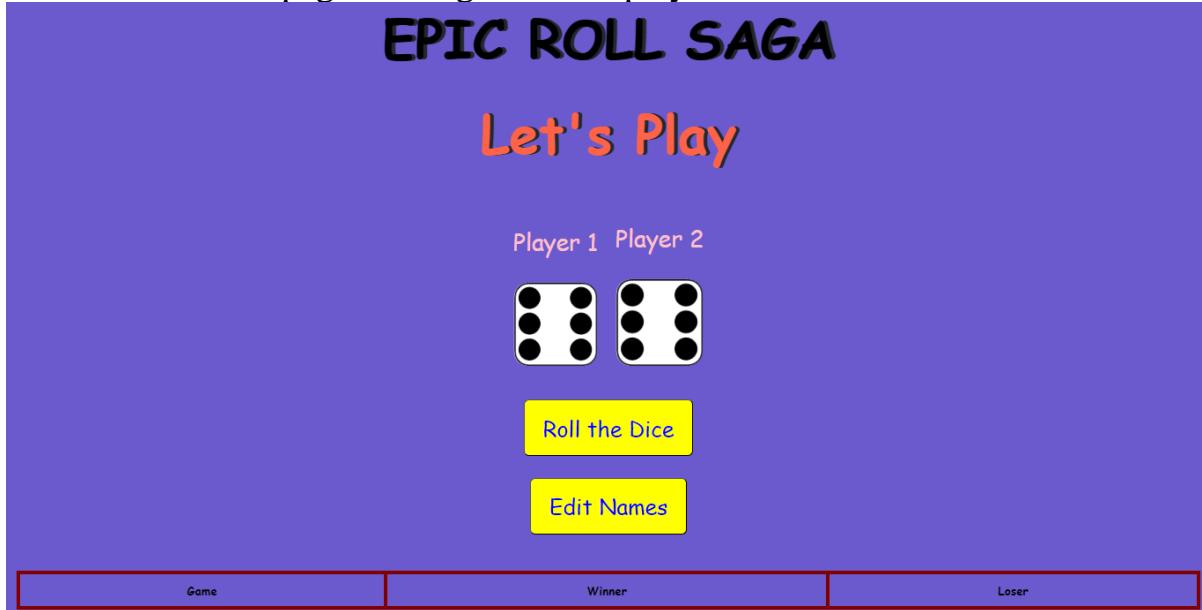
STEPS TO BE FOLLOWED:

- Create a folder named project.
- Copy the dice images from google chrome and paste it in the project folder.
- Write the code for epic roll saga which includes JavaScript , CSS and HTML
- Save the code with the .html extension

- Save the code with the .HTML extension in the same folder where the dice images are saved (as shown below 👇).

Name	Status	Date modified	Type	Size
DICE	✓	6/20/2024 12:15 PM	Chrome HTML Do...	6 KB
dice1	✓	4/30/2024 10:02 PM	PNG File	2 KB
dice2	✓	4/30/2024 10:02 PM	PNG File	2 KB
dice3	✓	4/30/2024 10:03 PM	PNG File	3 KB
dice4	✓	4/30/2024 10:03 PM	PNG File	3 KB
dice5	✓	4/30/2024 10:03 PM	PNG File	3 KB
dice6	✓	4/30/2024 10:04 PM	PNG File	3 KB

- After saving the code, right click on it in the folder and select open.
- The static page of the game is displayed :



- Click on “Edit Names” and change the names Player1, Player2:
(as shown below )

This page says

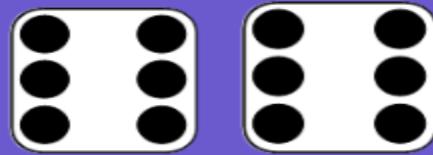
Change Player1 name

Aaron

OK

Cancel

Player 1 Player 2



Roll the Dice

Edit Names

- After editing the names by clicking on "Edit Names" , the new names of the players are displayed on the top of their respective dices (as shown below ↗):

EPIC ROLL SAGA

Let's Play

Aaron Ana



Roll the Dice

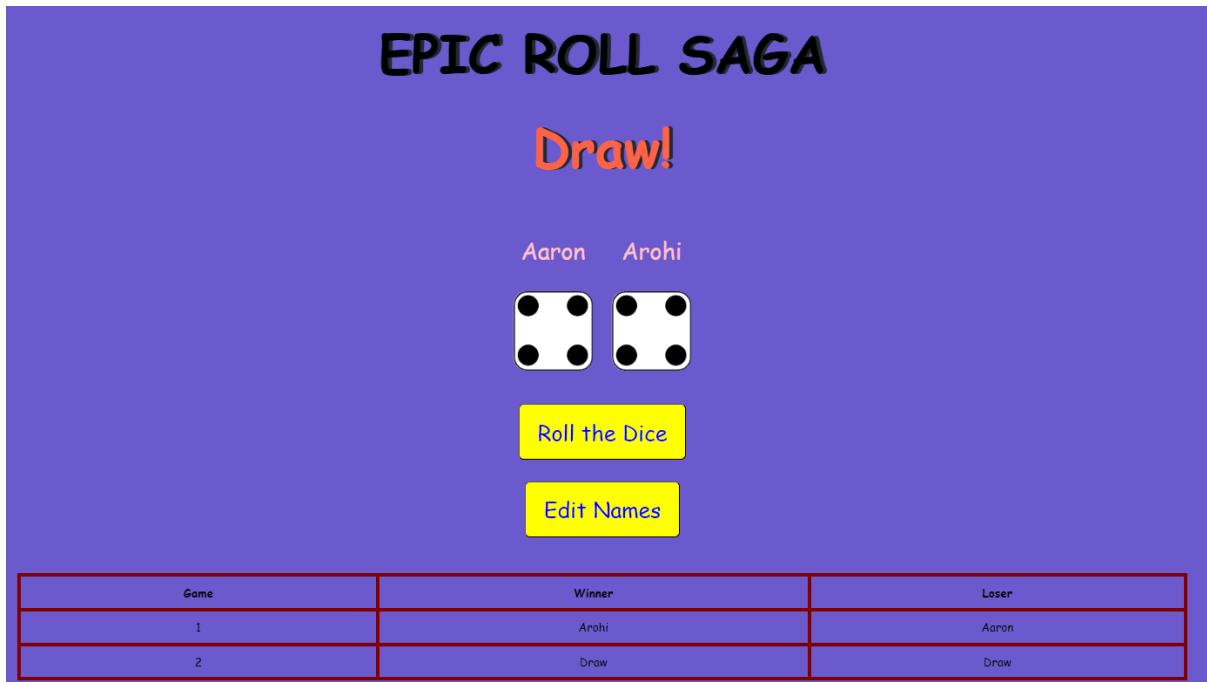
Edit Names

Game

Winner

Loser

- After Editing the player names, click on “Roll the Dice”.
- The dice of both players are rolled and the result is displayed in the table
- The scores of the players are updated in the score table indicating the game count, winner and loser after each round (as shown below 👇):



CONCLUSION

- The dice game provides a robust framework for understanding the basics of web development and offers a platform for exploring more advanced features and enhancements.
- Its simplicity makes it an excellent starting point for beginners, while its potential for expansion makes it a valuable learning tool for more experienced developers.
- Through continuous iteration and improvement, this game can evolve into a more complex and engaging application.