***A SOCIETAL RELATED Project on***

**WEB-BASED MULTI-USER ONLINE GAME**

**(ROCK-PAPER-SCISSORS)**

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

**BACHELOR OF TECHNOLOGY**

In

**CSE (DATA SCIENCE)**

By

**S Vikas (22AG1A6755)**

**M Govardhan Reddy (22AG1A6736)**

**V Vamshi Reddy (22AG1A6763)**

Under the guidance of

**P.Ashok Kumar**

**(Assistant Professor)**



**DEPARTMENT OF CSE (DATA SCIENCE)**

**ACE Engineering College**

**Ankushapur(V), Ghatkesar(M), Medchal Dist - 501 301**

***(An Autonomous Institution, Affiliated to JNTUH, Hyderabad)***

www.aceec.ac.in

**2024-2025**



**DEPARTMENT OF CSE (DATA SCIENCE)**

**CERTIFICATE**

This is to certify that the Societal Related project report entitled “**WEB-BASED MULTIUSER ONLINE GAME**” is a bonafide work done by **S.VIKAS(22AG1A6755), M.GOVARDHANREDDY(22AG1A6736), V.VAMSHIREDDY(22AG1A6763)**in partial fulfillment for the award of Degree of BACHELOR OF TECHNOLOGY in CSE (Data Science) from JNTU University, Hyderabad during the academic year 2023- 2024. This record of bonafide work carried out by them under our guidance and supervision.

The results embodied in this report have not been submitted by the student to any other University or Institution for the award of any degree or diploma.

**(P. Ashok kumar) (Dr. P. Chiranjeevi)**

Assistant Professor Associate Professor

Supervisor H.O.D., CSE-DS.

**ACKNOWLEDGEMENT**

We would like to express my gratitude to all the people behind the screen who have helped me transform an idea into a real time application.

We would like to express my heart-felt gratitude to my parents without whom we would not have been privileged to achieve and fulfill my dreams.

A special thanks to our General Secretary, **Prof. Y. V. Gopala Krishna Murthy,** for having founded such an esteemed institution. Sincere thanks to our Joint Secretary **Mrs. M. Padmavathi**, for support in doing project work. I am also grateful to our beloved principal, **Dr. B. L. RAJU** for permitting us to carry out this project.

We profoundly thank **Dr. P. Chiranjeevi**, Associate Professor and Head of the Department of Computer Science and Engineering (Data Science), who has been an excellent guide and also a great source of inspiration to my work.

We extremely thank **Mrs. B. Saritha** and **Mr. M. Hari Krishna,** Assistant Professors, Project coordinators, who helped us in all the way in fulfilling of all aspects in completion of our Mini-Project.

We are very thankful to my internal guide **P.Ashok kumar** who has been an excellent and also given continuous support for the Completion of my project work.

The satisfaction and euphoria that accompany the successful completion of the task would be great, but incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success. In this context, I would like to thank all the other staff members, both teaching and non-teaching, who have extended their timely help and eased my task.

**S.Vikas (22AG1A6755)**

**M.Govardhan Reddy (22AG1A6736)**

**V.Vamshi Reddy (22AG1A6763)**

**WEB-BASED MULTI-USER ONLINE GAME**

**(ROCK-PAPER-SCISSORS)**

**ABSTRACT**

This project aims to design and develop an online Rock, Paper, Scissors game as a web-based application. The game will provide users with the option to play against the computer or against other online players in real-time.

Key features include user registration and authentication, an intuitive game interface, robust game logic, scalability and performance optimization, multiplayer support with matchmaking capabilities, leaderboard and statistics tracking and security measures to safeguard user data.

Through thorough testing and documentation, the objective is to create an engaging and user-friendly platform for players to enjoy the classic game of Rock, Paper, Scissors while fostering interaction and competition within the online gaming community.

**CONTENT**

**S.NO CHAPTER NAME PAGE NO**

1 INTRODUCTION 7

2 EXISTING SYSTEM 8

3 LITERATURE SURVEY 9

4 PROPOSED MODEL / SYSTEM 12

5 REQUIREMENTS GATHERING

5.1 Software Requirements & 13

Hardware Requirements

5.2 Functional Requirements & 13

Non-Functional Requirements

6 SYSTEM ANALYSIS & DESIGN

6.1 Module Description 16

6.2 System Architecture 17

6.3 ER Diagram 18

6.4 UML Diagrams 19

7 IMPLEMENTATION

7.1 Sample Code 25

8 RESULTS 50

9 CONCLUSION & FUTURE

WORKS 57

**CHAPTER 1. INTRODUCTION**

What is ROCK-PAPER-SCISSORS???

Rock-paper-scissors is a simple hand game usually played between people, where each player simultaneously forms one of three shapes with an outstretched hand. These shapes are:

Rock (a fist)

Paper (an open hand)

Scissors (a fist with the index and middle fingers extended, forming a V)

The possible outcomes are:

Rock crushes scissors (rock wins)

Scissors cuts paper (scissors win)

Paper covers rock (paper wins)

If both players choose the same shape, the game is a tie and usually played again.

This project can have societal implications or applications in various contexts:

Decision-making

Conflict resolution

Education

Icebreaker activities

**CHAPTER 2. EXISTING SYSTEM**

Existing digital Rock-Paper-Scissors games provide a solid foundation and a variety of features that cater to both casual and serious players. However, there is always room for innovation and improvement, especially in enhancing user experience, adding advanced features, and creating more engaging gameplay. This project aims to address some of these limitations and offer a fresh take on the classic game.

Game Logic: Determines winner based on player and computer choices.

Computer Opponent: Player plays against computer, whose choice is randomly or AI-based.

Feedback and Results: Provides visual feedback on round outcomes.

Score Tracking: System can track and display score after each round

**CHAPTER 3. LITERATURE SURVEY**

### The Rock-Paper-Scissors (RPS) game, though simple in concept, has been the subject of various studies and implementations in both traditional and digital formats. This literature survey reviews existing research, development, and enhancements in RPS game implementations, exploring algorithmic strategies, user interaction, and technological advancements.

### Game Theory and Strategy

**Nash Equilibrium and Mixed Strategies:**

Research by Nash (1950) on equilibrium points in n-person games laid the foundation for understanding strategic decision-making in games like RPS. Nash's equilibrium concepts apply to RPS, where players use mixed strategies to randomize their choices to avoid predictability.

Selten (1975) expanded on this by studying subgame perfection and extensive form games, which are relevant in iterated RPS scenarios where history of play can influence strategy.

**Algorithmic Play:**

Studies by Rapoport and Boebel (1992) explored human strategies in repeated RPS games, identifying patterns and adaptive strategies. This research is crucial for developing AI opponents that can mimic human-like unpredictability and adapt to player behavior.

A more recent study by Erev and Roth (1998) investigated reinforcement learning models in RPS, which can be applied to develop computer opponents that improve their strategy based on past outcomes.

### Computational Implementations

**Random Number Generation:**

Marsaglia and Zaman (1991) discussed RNG algorithms critical for ensuring unpredictability in computer-generated choices in RPS games. The effectiveness of these algorithms affects the fairness and challenge level of the game.

Knuth (1997) provided a comprehensive overview of random number generation techniques that are widely implemented in game development, including RPS.

**AI and Machine Learning:**

Development of AI for RPS has been explored through various approaches, including heuristic algorithms, machine learning, and neural networks. Research by Doya (2000) on reinforcement learning has significant implications for creating AI that adapts to player strategies.

### User Interaction and Interface Design

**Human-Computer Interaction (HCI):**

Shneiderman (1998) emphasized the importance of user-friendly interfaces in game design. Effective HCI principles ensure that RPS games are accessible and enjoyable for players of all ages.

Norman (2002) discussed design principles that enhance user engagement, which are crucial for creating an immersive RPS game experience.

**Usability Studies:**

Nielsen (1994) conducted usability studies that highlight the significance of intuitive design and ease of use in software applications, including games. These principles are essential for developing a seamless and engaging RPS game.

### Technological Advancements

**Web and Mobile Technologies:**

The advent of web technologies like HTML5, CSS3, and JavaScript has enabled the development of browser-based RPS games. Research by Flanagan (2006) on JavaScript frameworks highlights tools and libraries that simplify game development.

Mobile development frameworks such as React Native and Flutter have been explored for creating cross-platform RPS games, offering insights into responsive design and performance optimization (Sametinger, 1997).

**Networking and Multiplayer Gaming:**

Studies on networked gaming by Berners-Lee et al. (1992) provide foundational knowledge for implementing online multiplayer modes in RPS games. This includes considerations for latency, synchronization, and fair play.

Research on peer-to-peer gaming networks (Smed, 2001) is relevant for developing robust and scalable multiplayer RPS games.

**CHAPTER 4. PROPOSED SYSTEM**

Multiplayer Game Development Overview:

User Authentication: Implement secure authentication methods like username/password or OAuth.

Friend Management: Allow users to add and manage friends.

Game Lobby: Display the status of online and offline friends.

Invitations: Enable players to send game invitations to friends.

Real-Time Gameplay: Implement gameplay mechanics and establish a bidirectional communication channel.

Game Rooms: Create separate game rooms for each match.

Score Tracking: Keep track of each player's score in multiplayer matches.

Endgame Handling: Display final score and provide options for new matches or lobby return.

Security Considerations: Ensure security of player data and communications.

**CHAPTER 5. REQUIREMENTS GATHERING**

**5.1 SOFTWARE AND HARDWARE REQUIREMENTS**

Software: OS – Windows

Hardware : 320GB (Hard disk)

Minimum of 4GB RAM.

**5.2 FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS**

FUNCTIONAL REQUIRMENTS

1. The system will enable users to register, login, and store their data securely.
2. Users can choose a move (rock, paper, scissors) and validate their moves.
3. The system will determine the winner of a round based on traditional Rock Paper Scissors rules.
4. The system will display the outcome to all players, track player wins, losses, and ties, and notify users when they are matched with another player.
5. It will also provide a user-friendly interface for easy navigation.
6. The system will also track player wins, losses, and ties.

NON-FUNCTIONAL REQUIRMENTS

1. The system should have low latency to provide a responsive gameplay experience.
2. The system should be able to handle a high number of concurrent users (scalability).
3. User data (passwords) should be securely stored using hashing techniques.
4. Communication between client and server should be encrypted (HTTPS).
5. The user interface should be intuitive and easy to use.
6. The game should be accessible to users with different levels of technical expertise.
7. The system code should be well-documented and easy to maintain.
8. The system should be modular to allow for future enhancements.

**CHAPTER 6. SYSTEM ANALYSIS AND DESIGN**

**6.1 MODULE DESCRIPTION**

User Interface (UI) Module:

The UI module handles all aspects of user interaction, providing a graphical interface that allows players to easily navigate and play the game.

Game Logic Module:

This module contains the core logic of the game, determining the outcome of each round based on the choices made by the players or the computer.

### Randomization Module:

Ensures the computer's choices are unpredictable, enhancing the challenge and fairness of the game.

### AI Opponent Module:

Develops an intelligent opponent that can adapt its strategy based on the player's behavior, providing a more challenging experience.

### Multiplayer Module:

Enables two players to compete against each other either on the same device or over a network connection.

### Data Storage Module:

Manages the storage and retrieval of game data, such as scores and user preferences.

### Sound and Animation Module:

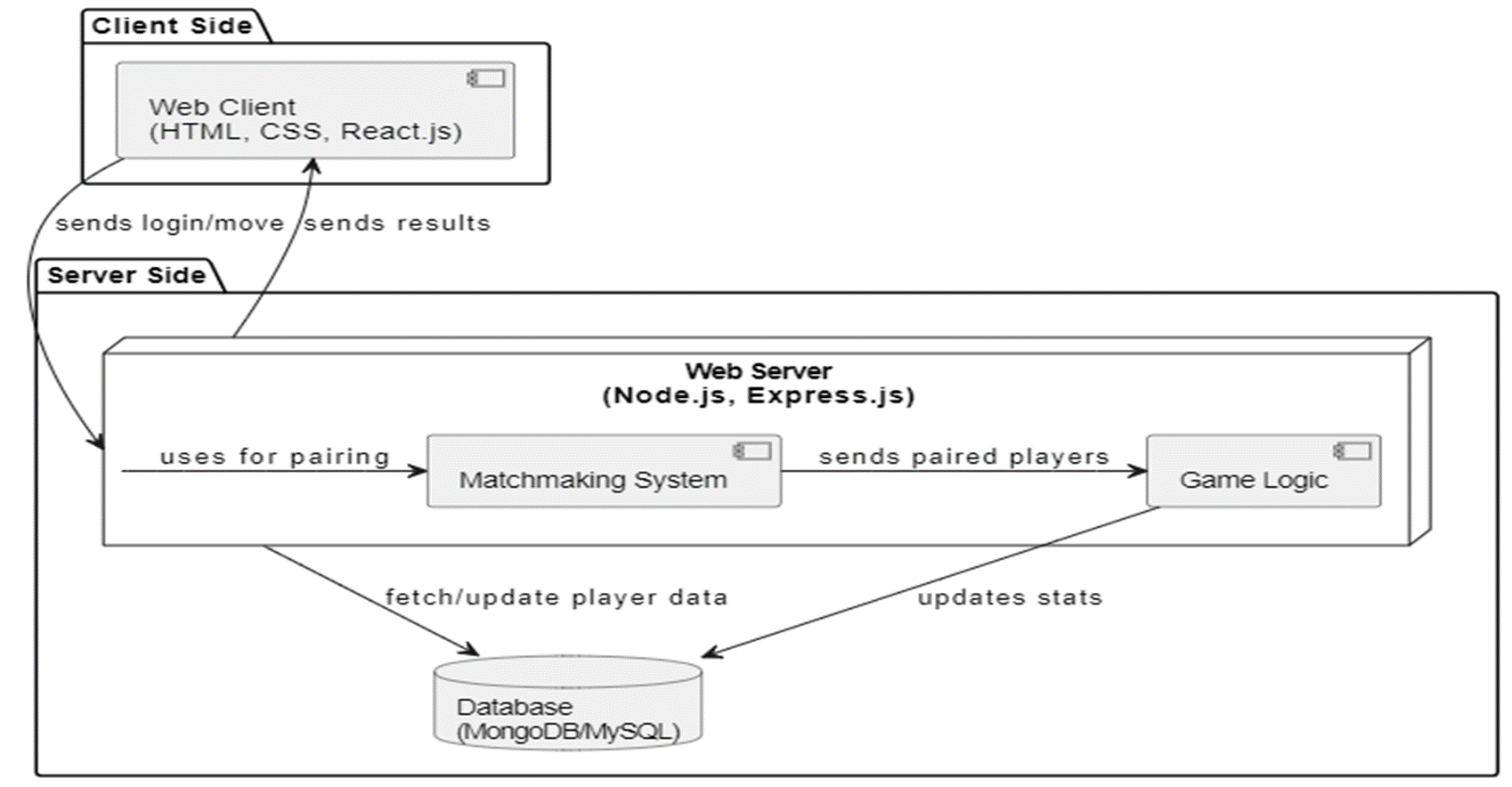
Enhances the user experience with audio-visual effects that make the game more engaging and enjoyable.

### Testing and Debugging Module:

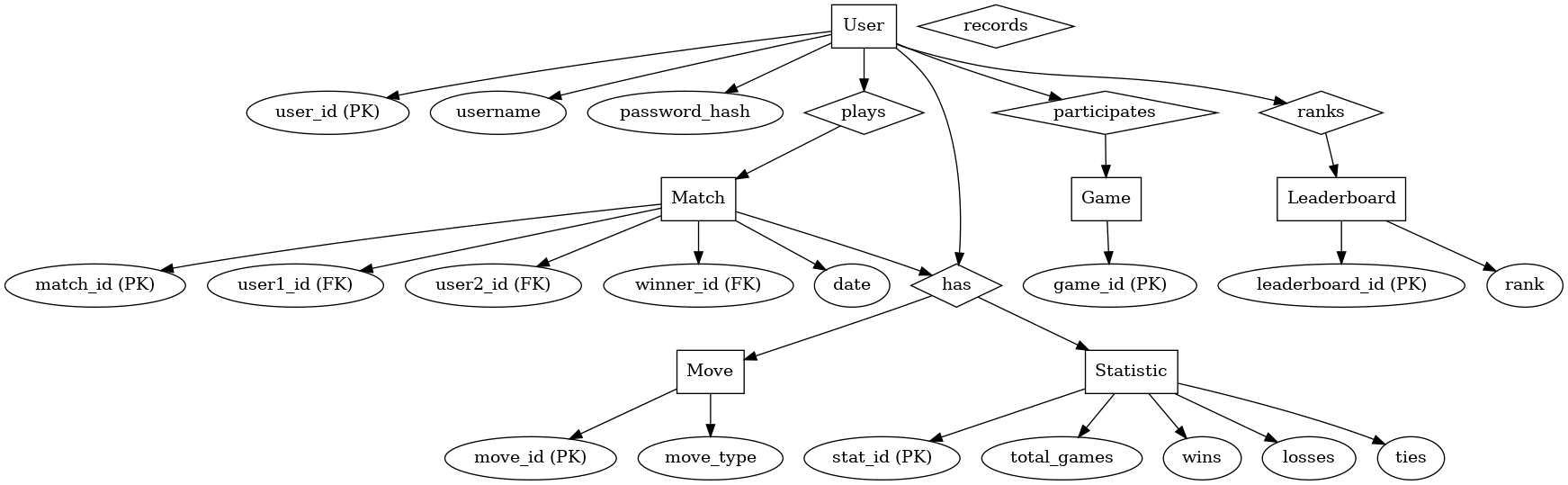
Ensures the game runs smoothly and is free of bugs through comprehensive testing and debugging.

### Documentation and Help Module:

Provides instructions and support for users to understand and play the game effectively.

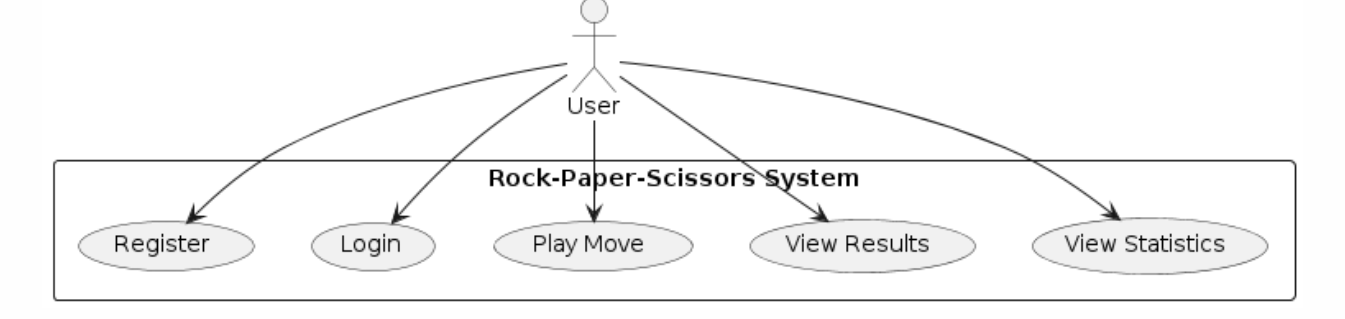
**6.2 SYSTEM ARCHITECTURE**

**6.3 ER DIAGRAM**

****

**6.4 UML DIAGRAMS**

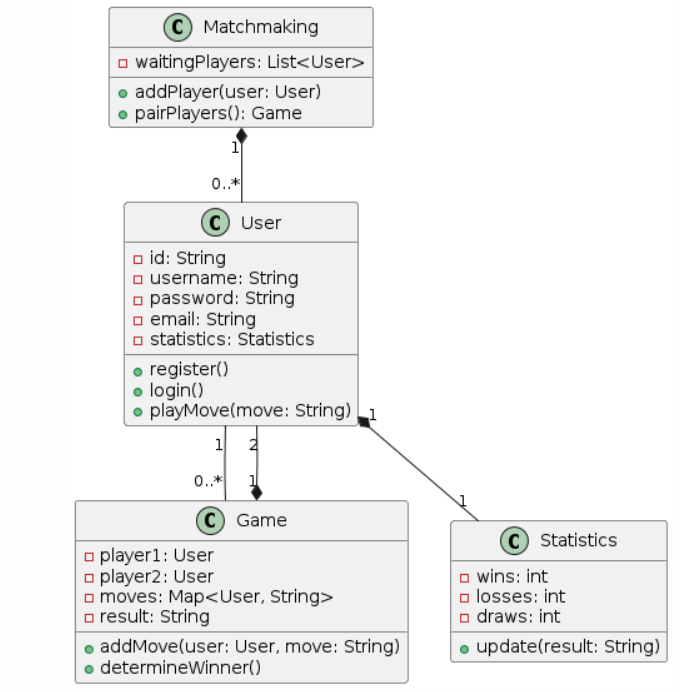
**1. USECASE DIAGRAM**

****

This diagram illustrates the interactions between the user and the system:

* Register: The user creates a new account.
* Login: The user logs into the system.
* Play Move: The user makes a move in the Rock-Paper-Scissors game.
* View Results: The user can view the outcome of the game.
* View Statistics: The user can view their game statistics, such as win/loss records.

**2. CLASS DIAGRAM**

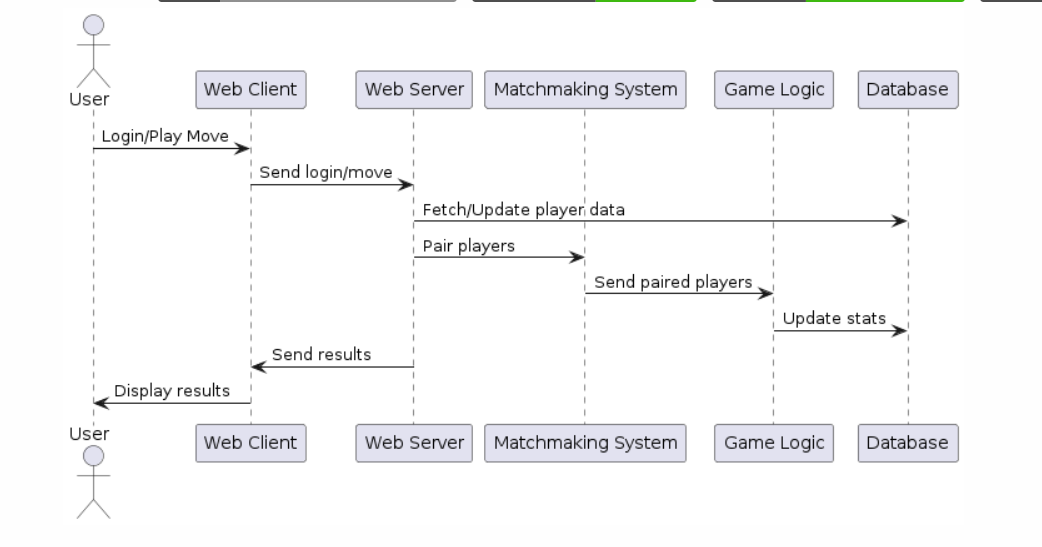
****

Purpose: The class diagram describes the static structure of the system, showing the system's classes, their attributes, operations, and relationships among the objects.

Key Components:

* Matchmaking Class:
  + waitingPlayers: List<User>: A list that holds users waiting to be paired for a game.
  + addPlayer(user: User): Adds a user to the waiting list.
  + pairPlayers(): Pairs two users for a game.
* User Class:
  + Attributes like id, username, password, email, and statistics which is a reference to the Statistics class.
  + Methods like register(), login(), and playMove(move: String).
* Game Class:
  + Holds references to player1 and player2.
  + Attributes include moves, which is a map of user moves, and result.
  + Methods include addMove() and determineWinner().
* Statistics Class:
  + Attributes for tracking wins, losses, and draws.
  + Method update(result: String) to update statistics based on the game result

**3. SEQUENCE DIAGRAM**

****

Purpose: The sequence diagram outlines the interaction between different components over time, focusing on the sequence of operations.

Key Sequence:

* User Interaction: Users log in or play a move through the Web Client.
* Server Communication: The Web Client sends login or move data to the Web Server.
* Data Processing: The Web Server interacts with the Matchmaking System to fetch or update player data, and with the Game Logic to process moves.
* Game Matching: The Matchmaking System pairs players, and the Game Logic updates the game state.
* Result Display: The results are sent back to the Web Client and displayed to the user. The system updates the database with the latest game statistics

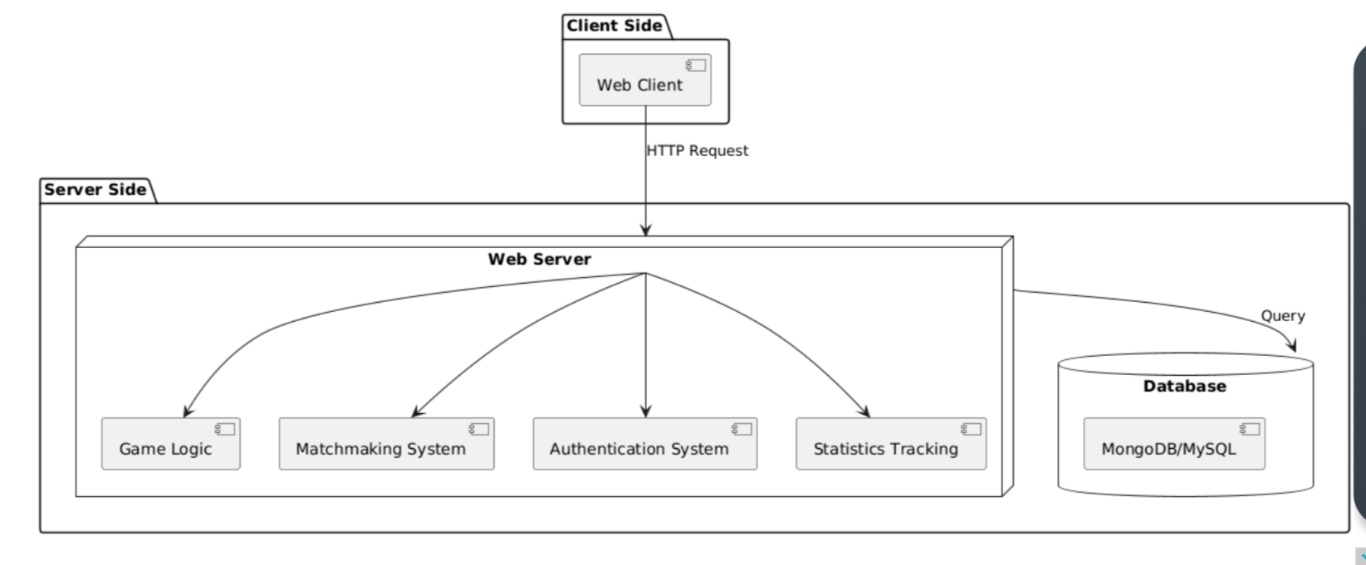
**4. ACTIVITY DIAGRAM**

****

This diagram illustrates the flow of activities in the Rock-Paper-Scissors game system:

* User Registration/Login: The user starts by either registering a new account or logging into an existing one.
* Game Selection: The user selects the Rock-Paper-Scissors game.
* Matchmaking: The system matches the user with another player.
* Game Play: The players make their moves in the game.
* Determine Winner: The system compares the moves and determines the winner.
* Update Statistics: The game result is recorded in the system’s statistics.
* Logout: The user logs out, ending their session.

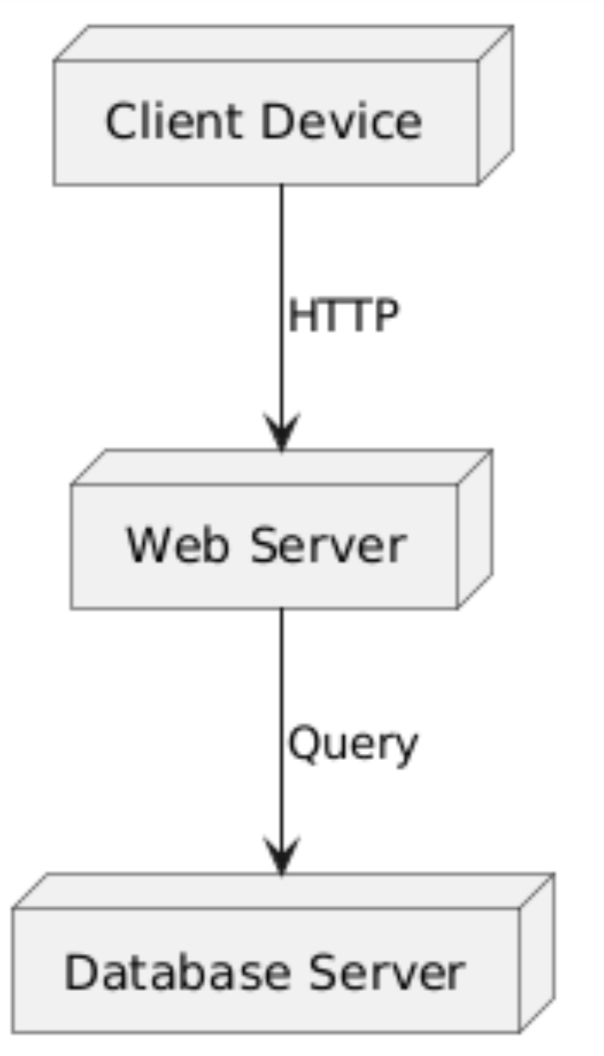
**5. COMPONENT DIAGRAM**

****Purpose: The component diagram shows the high-level architecture of the system, focusing on how different components interact with each other.

Key Components:

* Web Client: The front-end interface through which users interact with the system.
* Web Server: The central component that handles HTTP requests from the Web Client and coordinates the flow between other components.
* Game Logic: Handles the core game functions like move validation and determining winners.
* Matchmaking System: Pairs users to play against each other.
* Authentication System: Manages user login and registration processes.
* Statistics Tracking: Tracks and updates user game statistics.
* Database: Stores persistent data like user information and game statistics, which is managed via MongoDB/MySQL

**6. DEPLOYMENT DIAGRAM**

****

This diagram shows the physical arrangement of the system's components:

* Client Device: The user's device, which interacts with the web server.
* Web Server: Handles HTTP requests from the client and interacts with the database server.
* Database Server: Stores and retrieves game data and user information as required by the web server

**CHAPTER 7. IMPLEMENTATION**

**7.1 SAMPLE CODE**

App.js

import './App.css';

import ComputerGame from './computerGame';

import FriendGame from './friendGame';

import image from './images/bg.jpg';

import rock from './images/ROCK.png';

import scissors from './images/SCISSORS.png';

import { BrowserRouter as Router, Routes, Route, Link} from "react-router-dom";

import Login from './login';

import Signin from './signin';

import Start from './start';

// import HoverButton1 from './components/btn1';

// import HoverButton2 from './components/btn2';

const App= () => {

  const containerStyle = {

    position: 'relative',

    height: '100vh',

    width: '100vw',

    overflow: 'hidden'

  };

  const bg={

    backgroundImage: `url(${image})`,

    backgroundSize:'cover',

    backgroundRepeat: 'no-repeat',

    height: '100vh',

    width: '100%',

    backgroundPosition: 'center',

    position: 'absolute',

    filter: 'blur(3px)',

  };

  const contentStyle = {

    position: 'relative',

    zIndex: 1,

    textAlign: 'center',

    color: 'white',

    padding: '20px'

  };

  return (

    <Router>

      <div className="App" style={containerStyle}>

        <div style={bg}></div>

          <div style={contentStyle}>

            <Routes>

              <Route path="/signin" element={<Signin/>}></Route>

              <Route path="/login" element={<Login/>}></Route>

              <Route path="/computerGame" element={<ComputerGame/>}></Route>

              <Route path="/friendGame" element={<FriendGame/>}></Route>

              <Route path="/start" element={<Start/>}></Route>

            </Routes>

            <h1>ROCK PAPER SCISSORS</h1>

            <div className='cont'>

              <Link to="/signin" className='btn1'>SIGN UP</Link>

              <br></br>

              <br></br>

              <Link to="/login" className='btn2'>LOGIN</Link>

              <div className='images'>

                <img src={rock} className='rockimage' alt='rockimg'/>

                <img src={scissors} className='scissorsimage' alt='sciimg'/>

              </div>

            </div>

          </div>

        </div>

    </Router>

  );

}

export default App;

**CHAPTER 8. RESULTS**

**8.1 RESULT ANALYSIS**

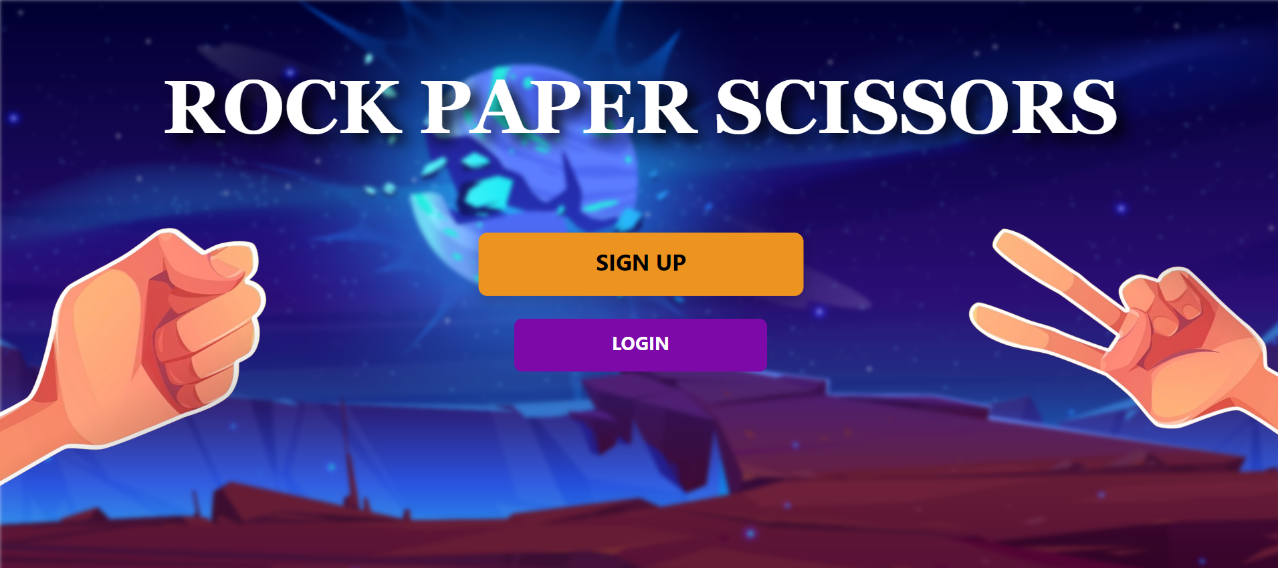
The development and implementation of the web-based multi-user Rock-Paper-Scissors game were successful, meeting the primary objectives of the project. The key features of the system, including user registration and authentication, intuitive game interface, robust game logic, scalability, and performance optimization, were effectively integrated and tested. The multiplayer support with matchmaking capabilities, leaderboard, statistics tracking, and security measures to safeguard user data were also implemented as planned.

Key Results:

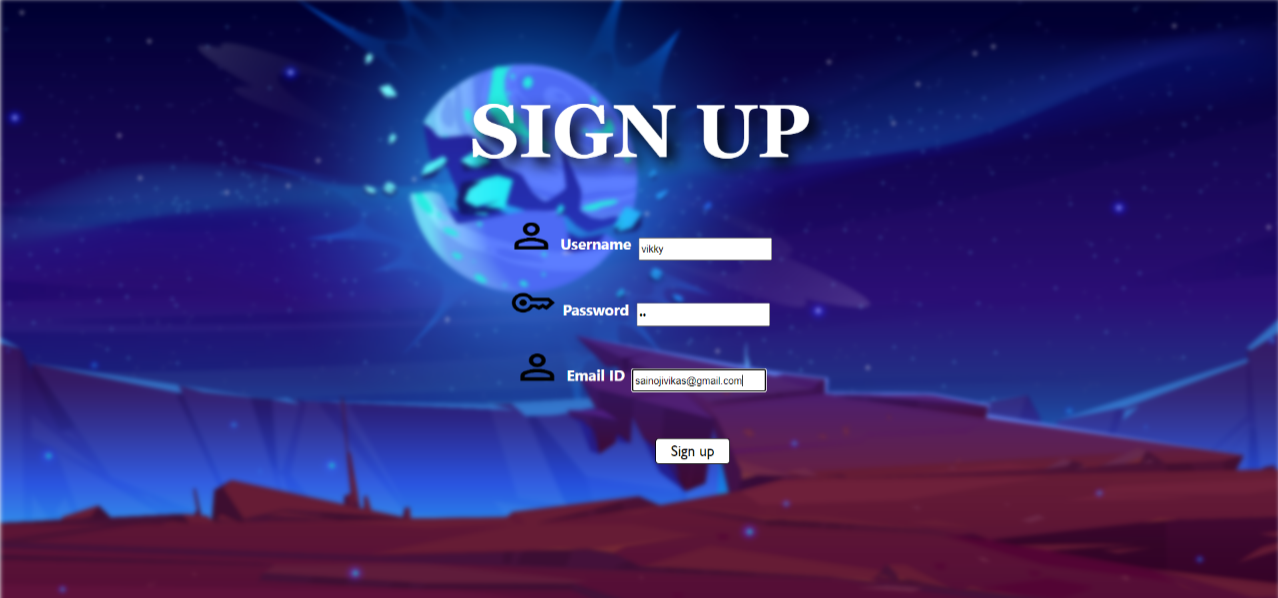
* User Interaction: The user interface was found to be intuitive and easy to navigate, providing a seamless user experience.
* Game Logic: The game logic accurately determined the winner of each round based on the traditional rules of Rock-Paper-Scissors, and handled ties appropriately.
* Performance: The system demonstrated low latency, providing a responsive gameplay experience even under high concurrent user loads.
* Security: User data, including passwords, were securely stored using hashing techniques, and communication between client and server was encrypted using HTTPS.
* Scalability: The system successfully handled a high number of concurrent users without performance degradation.
* Multiplayer Experience: The matchmaking system effectively paired users for gameplay, and the leaderboard and statistics tracking provided a competitive environment**.**

**8.2 OUTPUT SCREENS**

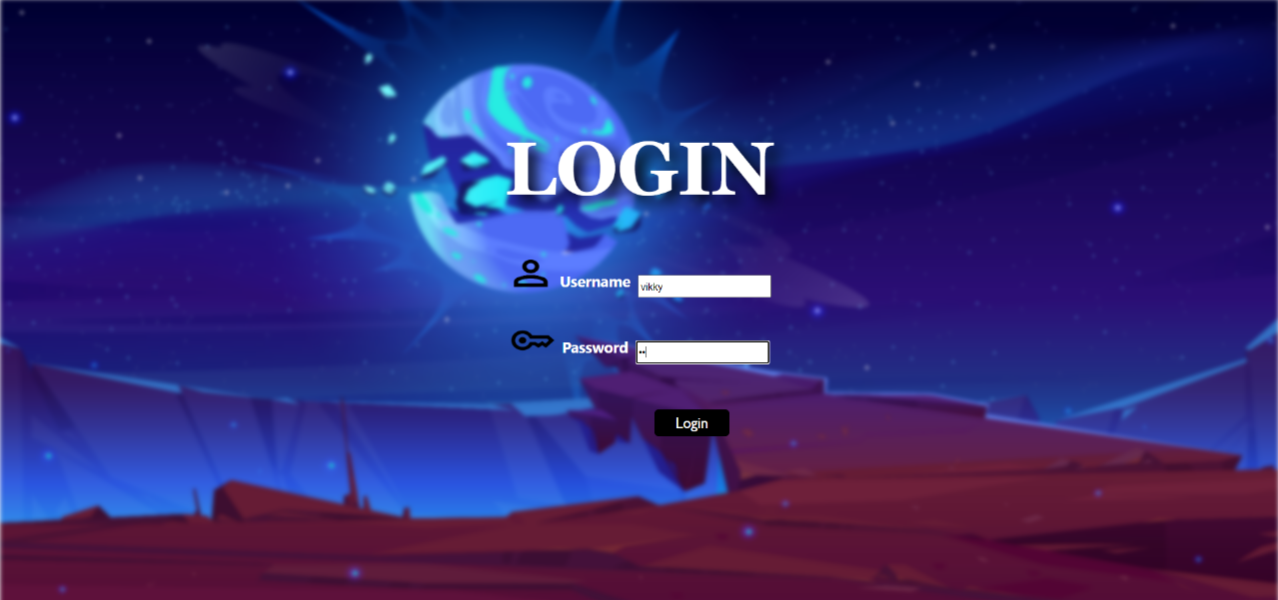
**FIRST PAGE TO SIGNUP OR LOGIN(if already signed):**

****

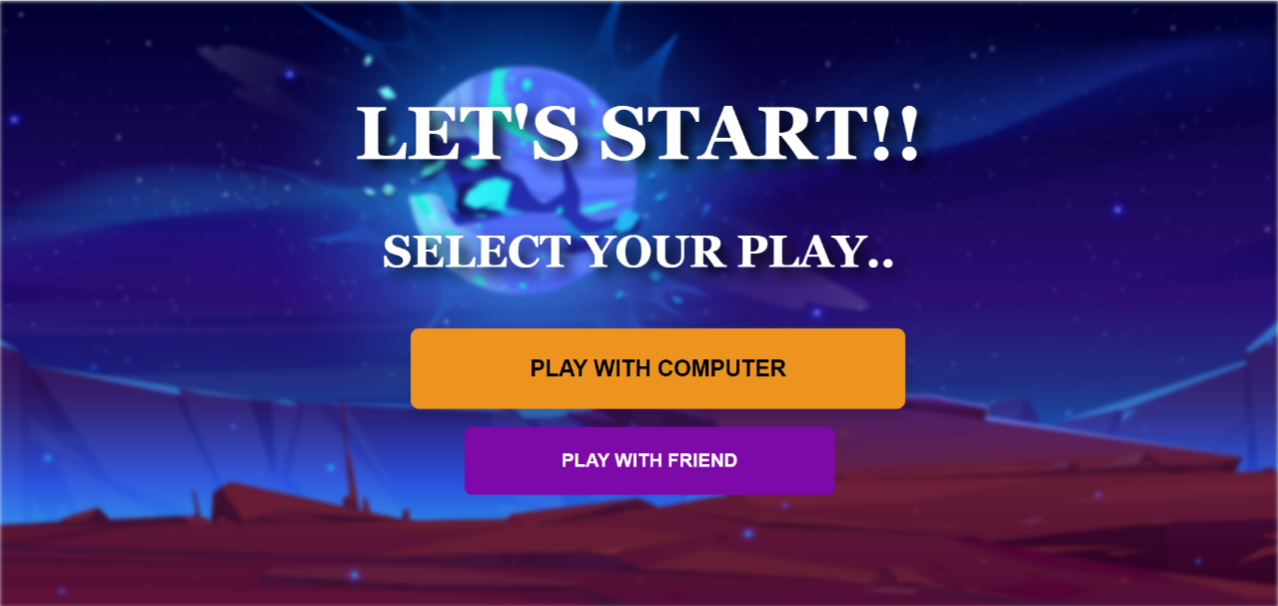
**SIGNUP PAGE:**

****

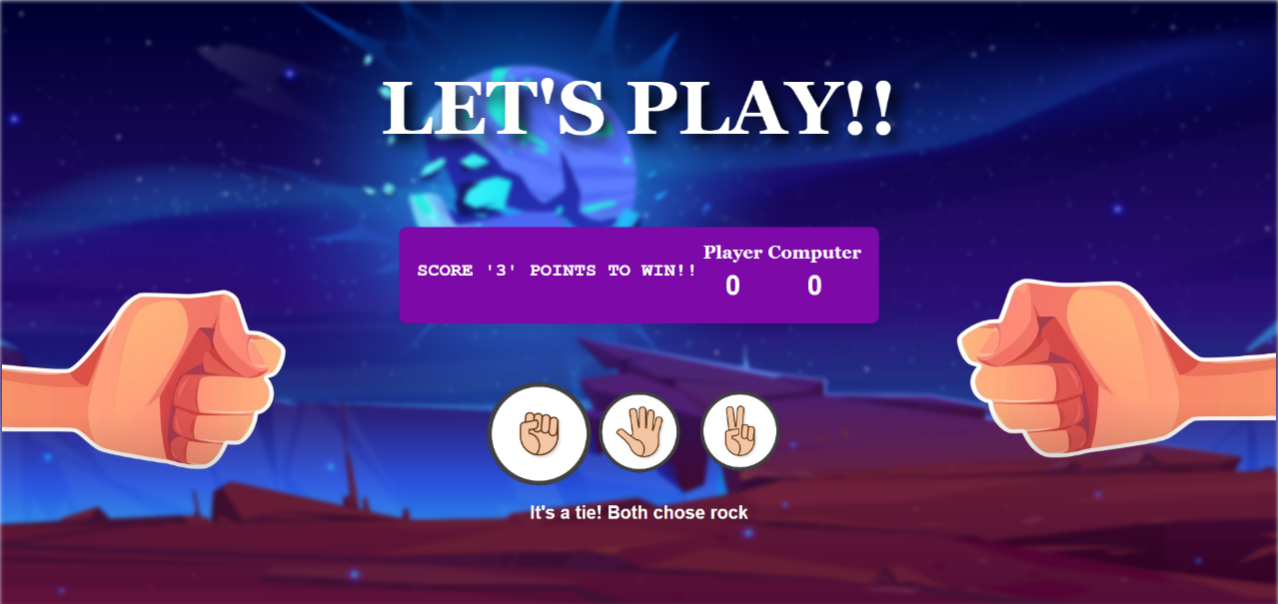
**LOGIN PAGE AFTER SIGNING UP:**

****

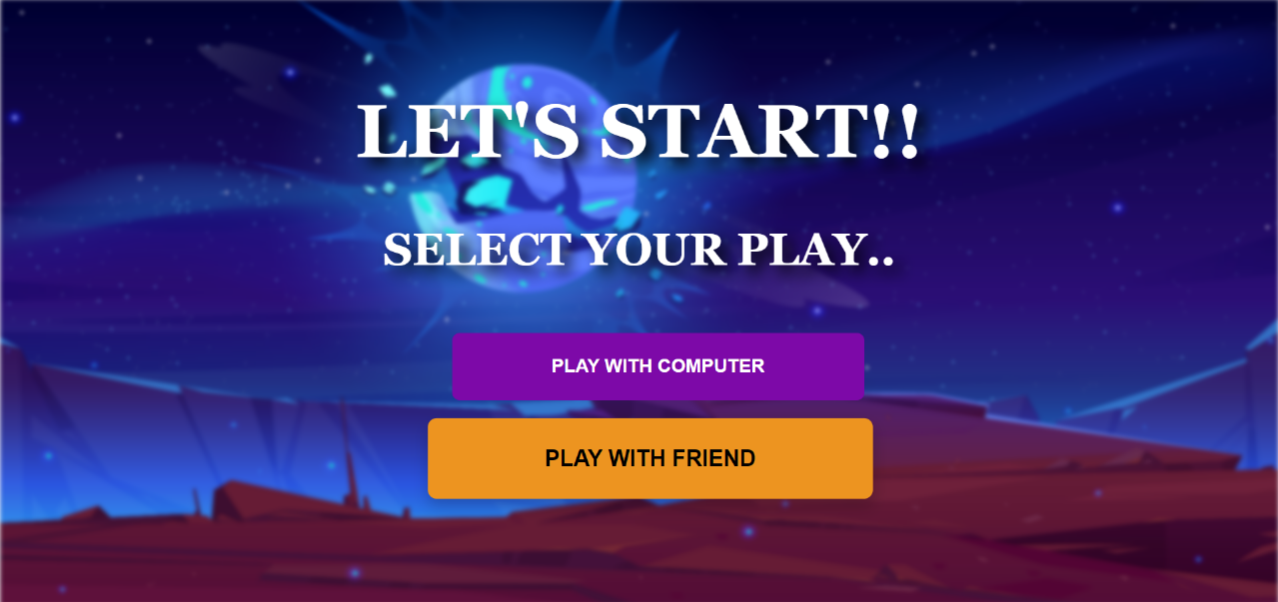
**START PAGE TO SELECT THE PLAY(either with friend or computer):**

****

**COMPUTER GAME:**

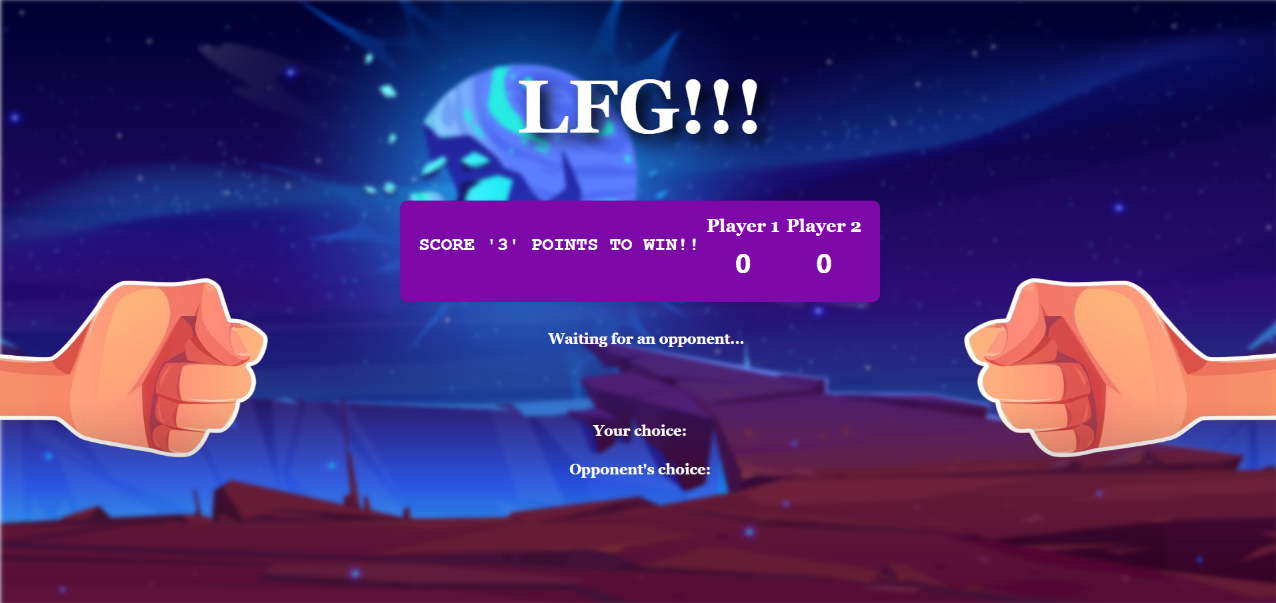
****

**START PAGE TO SELECT THE PLAY(either with friend or computer):**

****

**FRIEND GAME:**

**i) AFTER STARTING THE SERVER WITHOUT OPPONENT:**

****

**ii) WHEN THE OPPONENT HAS FOUND:**

****

**iii) GAMEPLAY:**

**SCREEN 1:**

****

**SCREEN 2:**

****

**DETERMINING WINNER:**

****

**CHAPTER 9. CONCLUSION AND FUTURE WORKS**

The project to develop a web-based multi-user online Rock-Paper-Scissors game has been successfully completed. The final product meets all specified requirements and offers a secure, scalable, and user-friendly platform for players to engage in the classic game. The implementation of advanced features, such as AI-based computer opponents, real-time multiplayer support, and comprehensive security measures, sets this game apart from existing solutions.

Future Work:

* Enhanced AI: Further development of the AI opponent to incorporate machine learning techniques for an even more challenging gameplay experience.
* Mobile Compatibility: Expanding the game's compatibility to mobile devices to reach a broader audience.
* Additional Features: Incorporating additional game modes, social features, and customizable user profiles to enhance player engagement.

The successful implementation of this project demonstrates the potential for creating engaging and interactive online gaming experiences that can foster community interaction and competition.

**BIBLIOGRAPHY:**

A bibliography is a list of sources or references that were used to research and write a document, essay, report, or book. It usually includes books, articles, websites, and other types of publications

These are the list of references according to the derived project:

1. https://www.online-stopwatch.com/chance-games/rock-paper-scissors/