**ADHIYAMAAN COLLEGE OF ENGINEERING (AUTONOMOUS)-HOSUR**

*DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING*

**SUBJECT: FULL STACK DEVELOPMENT**

**SUBJECT CODE: 618CIE02**

**TEAM MEMBERS:**

**SANJEEVITHA D**

**(6176AC21UCS120)**

**SHANJANA J**

**(6176AC21UCS125)**

**SHUBA S**

**(6176AC21UCS129)**

**PIG GAME**

** APP DEVELOPMENT**

***PROBLEM DESCRIPTION***

Pig game is a simple web designed game application using angular. Players take turns to roll a single die as many times as they wish, adding all roll results to a running total, but losing their gained score for the turn if they roll a 1.

***TOOLS:***

* HTML
* Javascript
* CSS

***PROGRAM:***

***INDEX.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>Pig Game</title>

</head>

<body>

<main>

<section class="player player--0 player--active">

<h2 class="name" id="name--0">Player 1</h2>

<p class="score" id="score--0">43</p>

<div class="current">

<p class="current-label">Current</p>

<p class="current-score" id="current--0">0</p>

</div>

</section>

<section class="player player--1">

<h2 class="name" id="name--1">Player 2</h2>

<p class="score" id="score--1">24</p>

<div class="current">

<p class="current-label">Current</p>

<p class="current-score" id="current--1">0</p>

</div>

</section>

<img src="images/dice-5.png" alt="Playing dice" class="dice" />

<button class="btn btn--new">🔄 New game</button>

<button class="btn btn--roll">🎲 Roll dice</button>

<button class="btn btn--hold">📥 Hold</button>

</main>

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

</body>

</html>

***SCRIPT.JS***

'use strict';

const player0El = document.querySelector('.player--0');

const player1El = document.querySelector('.player--1');

const score0El = document.querySelector('#score--0');

const score1El = document.getElementById('score--1');

const current0El = document.getElementById('current--0');

const current1El = document.getElementById('current--1');

const diceEl = document.querySelector('.dice');

const btnNew = document.querySelector('.btn--new');

const btnRoll = document.querySelector('.btn--roll');

const btnHold = document.querySelector('.btn--hold');

let scores, currentScore, activePlayer, playing;

const init = function () {

scores = [0, 0];

currentScore = 0;

activePlayer = 0;

playing = true;

score0El.textContent = 0;

score1El.textContent = 0;

current0El.textContent = 0;

current1El.textContent = 0;

diceEl.classList.add('hidden');

player0El.classList.remove('player--winner');

player1El.classList.remove('player--winner');

player0El.classList.add('player--active');

player1El.classList.remove('player--active');

};

init();

const switchPlayer = function () {

document.getElementById(`current--${activePlayer}`).textContent = 0;

currentScore = 0;

activePlayer = activePlayer === 0 ? 1 : 0;

player0El.classList.toggle('player--active');

player1El.classList.toggle('player--active');

};

btnRoll.addEventListener('click', function () {

if (playing) {

// 1. Generating a random dice roll

const dice = Math.trunc(Math.random() \* 6) + 1;

// 2. Display dice

diceEl.classList.remove('hidden');

diceEl.src = `images/dice-${dice}.png`;

// 3. Check for rolled 1

if (dice !== 1) {

// Add dice to current score

currentScore += dice;

document.getElementById(

`current--${activePlayer}`

).textContent = currentScore;

} else {

// Switch to next player

switchPlayer();

}

}

});

btnHold.addEventListener('click', function () {

if (playing) {

// 1. Add current score to active player's score

scores[activePlayer] += currentScore;

// scores[1] = scores[1] + currentScore

document.getElementById(`score--${activePlayer}`).textContent =

scores[activePlayer];

// 2. Check if player's score is >= 100

if (scores[activePlayer] >= 100) {

// Finish the game

playing = false;

diceEl.classList.add('hidden');

document

.querySelector(`.player--${activePlayer}`)

.classList.add('player--winner');

document

.querySelector(`.player--${activePlayer}`)

.classList.remove('player--active');

} else {

// Switch to the next player

switchPlayer();

}

}

});

btnNew.addEventListener('click', init);

***STYLE.CSS***

@import url('https://fonts.googleapis.com/css2?family=Nunito&display=swap');

\* {

margin: 0; padding: 0;box-sizing: inherit;}

html {

font-size: 62.5%;

box-sizing: border-box;

}

body {

font-family: 'Nunito', sans-serif;

font-weight: 400;

height: 100vh;

color: #333;

background-image: linear-gradient(to top left, #753682 0%, #bf2e34 100%);

display: flex;

align-items: center;

justify-content: center;

}

/\* LAYOUT \*/

main {

position: relative;

width: 100rem;

height: 60rem;

background-color: rgba(255, 255, 255, 0.35);

backdrop-filter: blur(200px);

filter: blur();

box-shadow: 0 3rem 5rem rgba(0, 0, 0, 0.25);

border-radius: 9px;

overflow: hidden;

display: flex;

}

.player {

flex: 50%;

padding: 9rem;

display: flex;

flex-direction: column;

align-items: center;

transition: all 0.75s;

}

/\* ELEMENTS \*/

.name {

position: relative;

font-size: 4rem;

text-transform: uppercase;

letter-spacing: 1px;

word-spacing: 2px;

font-weight: 300;

margin-bottom: 1rem;

}

.score {

font-size: 8rem;

font-weight: 300;

color: #c7365f;

margin-bottom: auto;

}

.player--active {

background-color: rgba(255, 255, 255, 0.4);

}

.player--active .name {

font-weight: 700;

}

.player--active .score {

font-weight: 400;

}

.player--active .current {

opacity: 1;

}

.current {

background-color: #c7365f;

opacity: 0.8;

border-radius: 9px;

color: #fff;

width: 65%;

padding: 2rem;

text-align: center;

transition: all 0.75s;

}

.current-label {

text-transform: uppercase;

margin-bottom: 1rem;

font-size: 1.7rem;

color: #ddd;

}

.current-score {

font-size: 3.5rem;

}

/\* ABSOLUTE POSITIONED ELEMENTS \*/

.btn {

position: absolute;

left: 50%;

transform: translateX(-50%);

color: #444;

background: none;

border: none;

font-family: inherit;

font-size: 1.8rem;

text-transform: uppercase;

cursor: pointer;

font-weight: 400;

transition: all 0.2s;

background-color: white;

background-color: rgba(255, 255, 255, 0.6);

backdrop-filter: blur(10px);

padding: 0.7rem 2.5rem;

border-radius: 50rem;

box-shadow: 0 1.75rem 3.5rem rgba(0, 0, 0, 0.1);

}

.btn::first-letter {

font-size: 2.4rem;

display: inline-block;

margin-right: 0.7rem;

}

.btn--new {

top: 4rem;

}

.btn--roll {

top: 39.3rem;

}

.btn--hold {

top: 46.1rem;

}

.btn:active {

transform: translate(-50%, 3px);

box-shadow: 0 1rem 2rem rgba(0, 0, 0, 0.15);

}

.btn:focus {

outline: none;

}

.dice {

position: absolute;

left: 50%;

top: 16.5rem;

transform: translateX(-50%);

height: 10rem;

box-shadow: 0 2rem 5rem rgba(0, 0, 0, 0.2);

}

.player--winner {

background-color: #2f2f2f;

}

.player--winner .name {

font-weight: 700;

color: #c7365f;

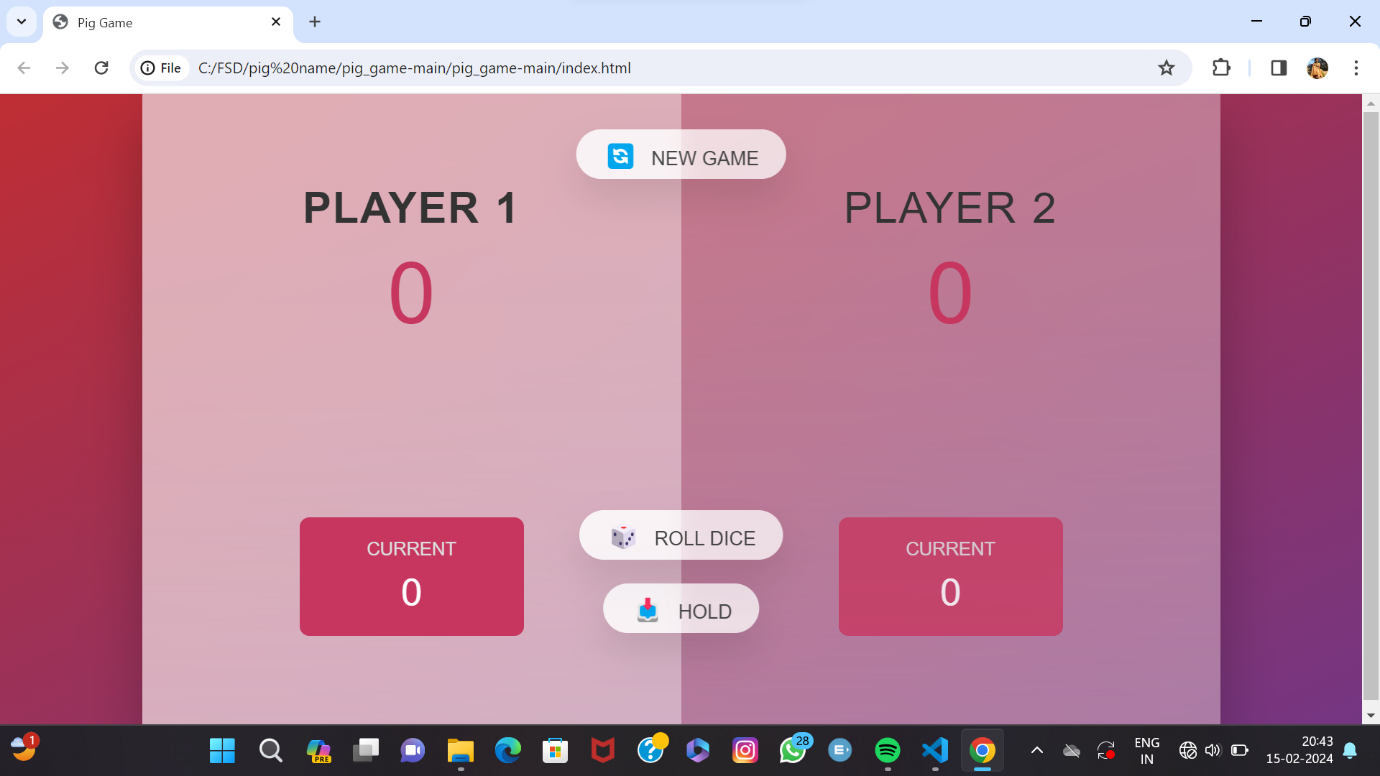
}

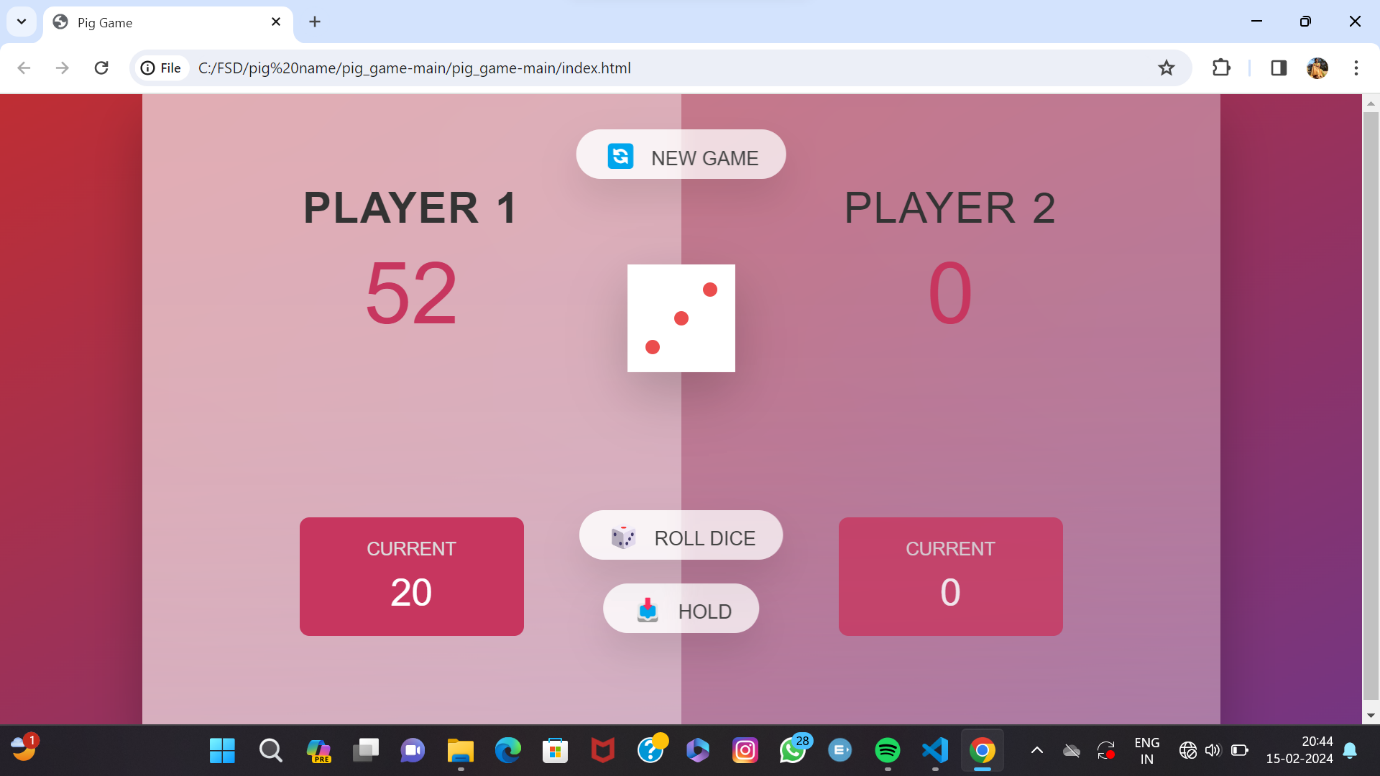
.hidden {

display: none;

}

**OUTPUT:**

****

****