**Main.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Hangman Game</title>

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

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

</head>

<body>

    <div class="banner">

        <video autoplay loop muted playsinline>

            <source src="video1.mp4" type="video/mp4">

        </video>

        <div class="navbar">

            <img class="logo" src="logo.png" alt="Logo">

            <ul>

                <li><a href="#">Home</a></li>

                <li><a href="#">FRUITS</a></li>

                <li><a href="#">ANIMAL</a></li>

                <li><a href="#">COUNTRIES</a></li>

            </ul>

        </div>

        <div class="content">

            <h1>HANG-MAN</h1>

            <div>

                <button id="start-button" type="button">START</button>

            </div>

        </div>

    </div>

</body>

</html>

**Gamepage.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Hangman Game</title>

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

</head>

<body>

    <div class="game-container">

        <canvas id="canvas" width="200" height="150"></canvas>

        <div id="options-container"></div>

        <div id="user-input-section"></div>

        <div id="letter-container" class="hide"></div>

        <div id="result-text"></div>

        <div id="new-game-container">

            <button id="new-game-button">New Game</button>

        </div>

    </div>

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

</body>

</html>

**Gamestyle.css**

/\* General Reset \*/

\* {

  margin: 0;

  padding: 0;

  font-family: sans-serif;

  box-sizing: border-box;

}

/\* Game Container \*/

.game-container {

  display: flex;

  flex-direction: column;

  align-items: center;

  justify-content: center;

  height: 100vh;

  background: url('https://t4.ftcdn.net/jpg/02/62/24/45/360\_F\_262244537\_RjHfRBucxPyo7o6QetIAQYpd5O3h6cEN.jpg') no-repeat center center fixed;

  background-size: cover; /\* Ensure the image covers the entire container \*/

}

/\* Canvas Style \*/

#canvas {

  border: 10px solid #000;

  background-color: #fff; /\* White background for the canvas \*/

}

/\* User Input Section \*/

#user-input-section {

  color: #fff;

  margin: 20px;

  font-size: 20px; /\* Adjusted font size \*/

  font-weight: bold;

}

/\* Letter Container \*/

#letter-container {

  display: flex;

  flex-wrap: wrap;

  gap: 10px; /\* Increased gap between letters \*/

  margin: 20px;

}

/\* Letter Buttons \*/

.letters {

  font-size: 15px; /\* Adjusted font size for letter buttons \*/

  padding: 10px;   /\* Adjust padding if necessary \*/

  margin: 5px;     /\* Adjust margin if necessary \*/

  border: 1px solid #ddd;

  background-color: #f0f0f0;

  cursor: pointer;

  transition: background-color 0.3s ease;

}

.letters:hover {

  background-color: #ddd;

}

/\* Option Container \*/

#options-container {

  color: #fff;

  margin: 20px;

  font-size: 28px; /\* Increased font size for option buttons \*/

}

/\* Option Buttons \*/

.options {

  font-size: 28px; /\* Font size for option buttons \*/

  padding: 10px 20px; /\* Adjust padding as needed \*/

  border: 2px solid #fff; /\* Border color to ensure visibility \*/

  background-color: #bb9e5f; /\* Background color of option buttons \*/

  color: #fff; /\* Font color for option buttons \*/

  cursor: pointer;

  transition: background-color 0.3s ease;

  margin: 5px; /\* Space between buttons \*/

}

.options:hover {

  background-color: #0056b3; /\* Darker shade on hover \*/

}

/\* Result Text \*/

#result-text {

  color: #fff;

  font-size: 25px; /\* Increased font size for result text \*/

  font-weight: bold;

  margin: 80px;

}

/\* New Game Container \*/

#new-game-container {

  margin: 50px;

}

/\* New Game Button \*/

#new-game-button {

  padding: 1px 2px; /\* Increased padding \*/

  font-size: 24px; /\* Increased font size \*/

  border: none;

  border-radius: 5px;

  background-color: #007bff;

  color: #fff;

  cursor: pointer;

  transition: background-color 0.3s ease;

}

#new-game-button:hover {

  background-color: #0056b3;

}

/\* Hide class for hidden elements \*/

.hide {

  display: none;

}

**Style.css**

\* {

  margin: 0;

  padding: 0;

  font-family: sans-serif;

  box-sizing: border-box;

}

.banner {

  width: 100%;

  height: 100vh;

  background-image: linear-gradient(rgba(0, 0, 0, 0.40), rgba(0, 0, 0, 0.40));

  background-position: center;

  background-size: cover;

  position: relative;

}

.navbar {

  width: 90%;

  padding: 30px;

  margin: auto;

  display: flex;

  align-items: center;

  justify-content: space-between;

}

.logo {

  width: 160px;

  margin-top: -10px;

  cursor: pointer;

}

.navbar ul {

  list-style-type: none;

  display: flex;

  align-items: center;

}

.navbar ul li {

  margin: 0 20px;

}

.navbar ul li a {

  text-decoration: none;

  text-transform: uppercase;

  color: white;

  font-weight: 600;

  padding: 15px;

}

.navbar ul li a:hover {

  background: white;

  color: black;

  transition: 0.6s;

}

.content {

  position: absolute;

  top: 50%;

  left: 50%;

  transform: translate(-50%, -50%);

  text-align: center;

  color: white;

}

.content h1 {

  font-size: 3em;

  font-weight: 800;

}

button {

  width: 200px;

  padding: 15px;

  margin: 20px 5px;

  text-align: center;

  border-radius: 25px;

  color: black;

  border: 2px solid black;

  font-size: 20px;

  cursor: pointer;

  font-weight: 600;

  background-color: white;

}

button:hover {

  background-color: rgb(0, 192, 226);

  color: white;

  transition: 0.5s;

}

.banner video {

  position: absolute;

  right: 0;

  bottom: 0;

  min-width: 100%;

  min-height: 100%;

  z-index: -1;

}

@media (min-aspect-ratio: 16/9) {

  .banner video {

      width: 100%;

      height: auto;

  }

}

**Script.js**

// script.js

// Function to initialize the game

const initializer = () => {

// Perform any game initialization here

console.log("Game initialized");

// Redirect to the game page after initialization

window.location.href = "gamepage.html"; // Replace with your actual game page URL

};

// Event listener for the START button

document.getElementById("start-button").addEventListener("click", () => {

// Call the initializer function when START button is clicked

initializer();

});

**Gamescript.js**

// Initial References

const letterContainer = document.getElementById("letter-container");

const optionsContainer = document.getElementById("options-container");

const userInputSection = document.getElementById("user-input-section");

const newGameContainer = document.getElementById("new-game-container");

const newGameButton = document.getElementById("new-game-button");

const canvas = document.getElementById("canvas");

const resultText = document.getElementById("result-text");

// Options values for buttons

let options = {

Fruits: ["Apple", "Blueberry", "Orange", "Pineapple", "Pomegranate", "Watermelon"],

Animals: ["Lion", "Rhinoceros", "Squirrel", "Panther", "Wolf", "Zebra", "Tiger"],

Countries: ["India", "Hungary", "NewZealand", "Switzerland", "Italy", "SriLanka"]

};

// Count

let winCount = 0;

let count = 0;

let chosenWord = "";

// Display option buttons

const displayOptions = () => {

optionsContainer.innerHTML = `<h3>Please Select An Option</h3>`;

let buttonCon = document.createElement("div");

for (let value in options) {

buttonCon.innerHTML += `<button class="options" onclick="generateWord('${value}')">${value}</button>`;

}

optionsContainer.appendChild(buttonCon);

};

// Block all the Buttons

const blocker = () => {

let optionsButtons = document.querySelectorAll(".options");

let letterButtons = document.querySelectorAll(".letters");

// Disable all options

optionsButtons.forEach(button => button.disabled = true);

// Disable all letters

letterButtons.forEach(button => button.disabled = true);

newGameContainer.classList.remove("hide");

};

// Word Generator

const generateWord = (optionValue) => {

let optionsButtons = document.querySelectorAll(".options");

optionsButtons.forEach(button => {

if (button.innerText.toLowerCase() === optionValue.toLowerCase()) {

button.classList.add("active");

}

button.disabled = true;

});

// Initially hide letters, clear previous word

letterContainer.classList.remove("hide");

userInputSection.innerText = "";

let optionArray = options[optionValue];

chosenWord = optionArray[Math.floor(Math.random() \* optionArray.length)].toUpperCase();

let displayItem = chosenWord.split("").map(char => `<span class="dashes">${char === " " ? "&nbsp;" : "\_"}</span>`).join(" ");

userInputSection.innerHTML = displayItem;

};

// Initial Function (Called when page loads/user presses new game)

const initializer = () => {

winCount = 0;

count = 0;

userInputSection.innerHTML = "";

optionsContainer.innerHTML = "";

letterContainer.classList.add("hide");

newGameContainer.classList.add("hide");

letterContainer.innerHTML = "";

for (let i = 65; i < 91; i++) {

let button = document.createElement("button");

button.classList.add("letters");

button.innerText = String.fromCharCode(i);

button.addEventListener("click", () => {

let charArray = chosenWord.split("");

let dashes = document.getElementsByClassName("dashes");

if (charArray.includes(button.innerText)) {

charArray.forEach((char, index) => {

if (char === button.innerText) {

dashes[index].innerText = char;

winCount += 1;

if (winCount === charArray.length) {

resultText.innerHTML = `<h2 class='win-msg'>You Win!!</h2><p>The word was <span>${chosenWord}</span></p>`;

blocker();

}

}

});

} else {

count += 1;

drawMan(count);

if (count === 6) {

resultText.innerHTML = `<h2 class='lose-msg'>You Lose!!</h2><p>The word was <span>${chosenWord}</span></p>`;

blocker();

}

}

button.disabled = true;

});

letterContainer.append(button);

}

displayOptions();

let { initialDrawing } = canvasCreator();

initialDrawing();

};

// Canvas

const canvasCreator = () => {

let context = canvas.getContext("2d");

context.beginPath();

context.strokeStyle = "#000";

context.lineWidth = 2;

const drawLine = (fromX, fromY, toX, toY) => {

context.moveTo(fromX, fromY);

context.lineTo(toX, toY);

context.stroke();

};

const head = () => {

context.beginPath();

context.arc(70, 30, 10, 0, Math.PI \* 2, true);

context.stroke();

};

const body = () => drawLine(70, 40, 70, 80);

const leftArm = () => drawLine(70, 50, 50, 70);

const rightArm = () => drawLine(70, 50, 90, 70);

const leftLeg = () => drawLine(70, 80, 50, 110);

const rightLeg = () => drawLine(70, 80, 90, 110);

const initialDrawing = () => {

context.clearRect(0, 0, context.canvas.width, context.canvas.height);

drawLine(10, 130, 130, 130);

drawLine(10, 10, 10, 131);

drawLine(10, 10, 70, 10);

drawLine(70, 10, 70, 20);

};

return { initialDrawing, head, body, leftArm, rightArm, leftLeg, rightLeg };

};

// Draw the man

const drawMan = (count) => {

let { head, body, leftArm, rightArm, leftLeg, rightLeg } = canvasCreator();

switch (count) {

case 1: head(); break;

case 2: body(); break;

case 3: leftArm(); break;

case 4: rightArm(); break;

case 5: leftLeg(); break;

case 6: rightLeg(); break;

default: break;

}

};

// New Game

newGameButton.addEventListener("click", initializer);

// Initialize the game when the page loads

document.addEventListener("DOzMContentLoaded", initializer);