<!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">

<title>Quick Quiz</title>

<style>

/\* CSS styles \*/

:root {

background-color: #ecf5ff;

font-size: 62.5%;

}

\* {

box-sizing: border-box;

font-family: Arial, Helvetica, sans-serif;

margin: 0;

padding: 0;

color: #333;

}

h1,

h2,

h3,

h4 {

margin-bottom: 1rem;

}

h1 {

font-size: 5.4rem;

color: #56a5eb;

margin-bottom: 5rem;

}

h1>span {

font-size: 2.4rem;

font-weight: 500;

}

h2 {

font-size: 4.2rem;

margin-bottom: 4rem;

font-weight: 700;

}

h3 {

font-size: 2.8rem;

font-weight: 500;

}

/\* UTILITIES \*/

.container {

width: 100vw;

height: 100vh;

display: flex;

justify-content: center;

align-items: center;

max-width: 80rem;

margin: 0 auto;

padding: 2rem;

}

.container>\* {

width: 100%;

}

.flex-column {

display: flex;

flex-direction: column;

}

.flex-center {

justify-content: center;

align-items: center;

}

.justify-center {

justify-content: center;

}

.text-center {

text-align: center;

}

.hidden {

display: none;

}

/\* BUTTONS \*/

.btn {

font-size: 1.8rem;

padding: 1rem 0;

width: 20rem;

text-align: center;

border: 0.1rem solid #56a5eb;

margin-bottom: 1rem;

text-decoration: none;

color: #56a5eb;

background-color: white;

}

.btn:hover {

cursor: pointer;

box-shadow: 0 0.4rem 1.4rem 0 rgba(86, 185, 235, 0.5);

transform: translateY(-0.1rem);

transition: transform 150ms;

}

.btn[disabled]:hover {

cursor: not-allowed;

box-shadow: none;

transform: none;

}

/\* FORMS \*/

form {

width: 100%;

display: flex;

flex-direction: column;

align-items: center;

}

input {

margin-bottom: 1rem;

width: 20rem;

padding: 1.5rem;

font-size: 1.8rem;

border: none;

box-shadow: 0 0.1rem 1.4rem 0 rgba(86, 185, 235, 0.5);

}

input::placeholder {

color: #aaa;

}

/\* Additional CSS \*/

.container {

margin: auto;

max-width: 600px;

padding: 20px;

}

.flex-center {

display: flex;

justify-content: center;

align-items: center;

}

.flex-column {

flex-direction: column;

}

.btn {

display: block;

width: 200px;

padding: 10px;

margin: 10px auto;

text-align: center;

background-color: #56a5eb;

color: white;

text-decoration: none;

border-radius: 5px;

}

.choice-container {

display: flex;

margin-bottom: 5px;

width: 100%;

font-size: 18px;

border: 1px solid rgba(86, 165, 235, 0.25);

background-color: white;

}

.choice-container:hover {

cursor: pointer;

box-shadow: 0 0.4px 1.4px 0 rgba(86, 185, 235, 0.5);

transform: translateY(-0.1px);

transition: transform 150ms;

}

.choice-prefix {

padding: 15px 25px;

background-color: #56a5eb;

color: white;

}

.choice-text {

padding: 15px;

width: 100%;

}

.correct {

background-color: #28a745;

}

.incorrect {

background-color: #dc3545;

}

#hud {

display: flex;

justify-content: space-between;

}

.hud-prefix {

text-align: center;

font-size: 20px;

}

.hud-main-text {

text-align: center;

}

#progressBar {

width: 200px;

height: 40px;

border: 3px solid #56a5eb;

margin-top: 15px;

}

#progressBarFull {

height: 34px;

background-color: #56a5eb;

width: 0%;

}

</style>

</head>

<body>

<div class="container" id="homeContainer">

<div id="home" class="flex-center flex-column">

<h1>Quick Quiz</h1>

<button class="btn" id="playBtn">Play</button>

</div>

</div>

<div class="container hidden" id="gameContainer">

<div id="game" class="justify-center flex-column">

<div id="hud">

<div id="hud-item">

<p id="progressText" class="hud-prefix">Question</p>

<div id="progressBar">

<div id="progressBarFull"></div>

</div>

</div>

<div id="hud-item">

<p class="hud-prefix">Score</p>

<h1 class="hud-main-text" id="score">0</h1>

</div>

</div>

<h2 id="question">What is the answer to this question?</h2>

<div class="choice-container">

<p class="choice-prefix">A</p>

<p class="choice-text" data-number="1">Choice 1</p>

</div>

</div>

</div>

<div class="container">

<div id="end" class="flex-center flex-column hidden">

<h1 id="finalScore"></h1>

<button class="btn" id="playAgainBtn">Play Again</button>

</div>

</div>

<script>

// JavaScript code

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

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

const endContainer = document.getElementById("end");

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

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

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

homeContainer.classList.add("hidden");

gameContainer.classList.remove("hidden");

startGame(); // Start the quiz when Play button is clicked

});

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

endContainer.classList.add("hidden");

homeContainer.classList.remove("hidden");

});

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

const choicesContainer = document.getElementById("game");

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

const scoreText = document.getElementById("score");

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

let currentQuestion = {};

let acceptingAnswers = false;

let score = 0;

let questionCounter = 0;

let availableQuestions = [10];

let questions = [

{

question: "Inside which HTML element do we put the JavaScript?",

choice1: "<script>",

choice2: "<javascript>",

choice3: "<js>",

choice4: "<scripting>",

answer: 1

},

{

question: "What is the correct syntax for referring to an external script called 'script.js'?",

choice1: "<script href='script.js'>",

choice2: "<script name='script.js'>",

choice3: "<script src='script.js'>",

choice4: "<script file='script.js'>",

answer: 3

},

{

question: "How do you write 'Hello World' in an alert box?",

choice1: "msgBox('Hello World');",

choice2: "alertBox('Hello World');",

choice3: "msg('Hello World');",

choice4: "alert('Hello World');",

answer: 4

},

{

question: "What does HTML stand for?",

choice1: "hyper text markup language",

choice2: "hyperlinks and text markup language",

choice3: "home tool markup language",

choice4: "hyper transfer markup language",

answer: 1

},

{

question: "Which tag is used to define the structure of an HTML document?",

choice1: "<header>",

choice2: "<body>",

choice3: "<html>",

choice4: "<title>",

answer: 3

},

{

question: "Which attribute is used to specify alternative text for an image in HTML?",

choice1: "alt",

choice2: "title",

choice3: "src",

choice4: "img",

answer: 1

},

{

question: "In HTML, which attribute is used to specify the URL of an image?",

choice1: "href",

choice2: "url",

choice3: "src",

choice4: "link",

answer: 3

},

{

question: "Which tag is used to define a table row in HTML?",

choice1: "<td>",

choice2: "<tr>",

choice3: "<table>",

choice4: "<th>",

answer: 2

},

{

question: "Which tag is used to create a horizontal line in HTML?",

choice1: "<line>",

choice2: "<hr>",

choice3: "<break>",

choice4: "<hline>",

answer: 2

},

{

question: "What is the correct HTML element for inserting a line break?",

choice1: "<lb>",

choice2: "<break>",

choice3: "<br>",

choice4: "<newline>",

answer: 3

}

];

const CORRECT\_BONUS = 10;

const MAX\_QUESTIONS = 10;

startGame = () => {

questionCounter = 0;

score = 10;

availableQuestions = [...questions];

getNewQuestion();

};

getNewQuestion = () => {

if (availableQuestions.length === 0 || questionCounter >= MAX\_QUESTIONS) {

localStorage.setItem("mostRecentScore", score);

endContainer.classList.remove("hidden");

return;

}

questionCounter++;

progressText.innerText = Question ${questionCounter}/${MAX\_QUESTIONS};

progressBarFull.style.width = ${(questionCounter / MAX\_QUESTIONS) \* 100}%;

const questionIndex = Math.floor(Math.random() \* availableQuestions.length);

currentQuestion = availableQuestions[questionIndex];

question.innerText = currentQuestion.question;

// Clear choices before populating

choicesContainer.querySelectorAll('.choice-container').forEach(choice => {

choice.remove();

});

// Populate choices

for (let i = 1; i <= 4; i++) {

const choice = document.createElement('div');

choice.classList.add('choice-container');

const prefix = document.createElement('p');

prefix.classList.add('choice-prefix');

prefix.innerText = String.fromCharCode(64 + i);

const text = document.createElement('p');

text.classList.add('choice-text');

text.setAttribute('data-number', i);

text.innerText = currentQuestion['choice' + i];

choice.appendChild(prefix);

choice.appendChild(text);

choicesContainer.appendChild(choice);

}

availableQuestions.splice(questionIndex, 1);

acceptingAnswers = true;

};

choicesContainer.addEventListener('click', (e) => {

const selectedChoice = e.target;

if (!selectedChoice.classList.contains('choice-text') || !acceptingAnswers) return;

acceptingAnswers = false;

const selectedAnswer = selectedChoice.dataset['number'];

const classToApply = selectedAnswer == currentQuestion.answer ? 'correct' : 'incorrect';

if (classToApply === 'correct') {

incrementScore(CORRECT\_BONUS);

}

selectedChoice.parentElement.classList.add(classToApply);

setTimeout(() => {

selectedChoice.parentElement.classList.remove(classToApply);

getNewQuestion();

}, 1000);

});

incrementScore = (num) => {

score += num;

scoreText.innerText = score;

};

</script>

</body>

</html>

|  | ReplyForward |
| --- | --- |