Deepak K 717821E113

Quiz app:

<https://deepakkumaravelu.github.io/javascript-quiz-app/>

<https://github.com/deepakkumaravelu/javascript-quiz-app>

Script.js:

let timeLeft = document.querySelector(".time-left");

let quizContainer = document.getElementById("container");

let nextBtn = document.getElementById("next-button");

let countOfQuestion = document.querySelector(".number-of-question");

let displayContainer = document.getElementById("display-container");

let scoreContainer = document.querySelector(".score-container");

let restart = document.getElementById("restart");

let userScore = document.getElementById("user-score");

let startScreen = document.querySelector(".start-screen");

let startButton = document.getElementById("start-button");

let questionCount;

let scoreCount = 0;

let count = 11;

let countdown;

const quizArray = [

{

id: "0",

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

options: ["js",

"javascript",

"scripting",

"script"],

correct: "script",

},

{

id: "1",

question: "Javascript is a \_\_\_\_\_ language.",

options: ["Programming",

"Application",

"Scripting",

"None of the above"],

correct: "Scripting",

},

{

id: "2",

question: "JavaScript is a \_\_\_\_\_ Side Scripting Language.",

options: [ "Server",

"Browser",

"ISP",

"None of the above"],

correct: "Browser",

},

{

id: "3",

question: "Which of the following purpose, JavaScript is designed for ?",

options: [ "To Execute Query Related to DB on Server",

"To Style HTML Pages",

"To Perform Server Side Scripting Opertion",

"To add interactivity to HTML Pages"],

correct: "To add interactivity to HTML Pages",

},

];

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

initial();

displayContainer.classList.remove("hide");

scoreContainer.classList.add("hide");

});

nextBtn.addEventListener(

"click",

(displayNext = () => {

questionCount += 1;

if (questionCount == quizArray.length) {

displayContainer.classList.add("hide");

scoreContainer.classList.remove("hide");

userScore.innerHTML =

"Your score is " + scoreCount + " out of " + questionCount;

} else {

countOfQuestion.innerHTML =

questionCount + 1 + " of " + quizArray.length + " Question";

quizDisplay(questionCount);

count = 11;

clearInterval(countdown);

timerDisplay();

}

})

);

const timerDisplay = () => {

countdown = setInterval(() => {

count--;

timeLeft.innerHTML = `${count}s`;

if (count == 0) {

clearInterval(countdown);

displayNext();

}

}, 1000);

};

const quizDisplay = (questionCount) => {

let quizCards = document.querySelectorAll(".container-mid");

quizCards.forEach((card) => {

card.classList.add("hide");

});

quizCards[questionCount].classList.remove("hide");

};

function quizCreator() {

quizArray.sort(() => Math.random() - 0.5);

for (let i of quizArray) {

i.options.sort(() => Math.random() - 0.5);

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

div.classList.add("container-mid", "hide");

countOfQuestion.innerHTML = 1 + " of " + quizArray.length + " Question";

let question\_DIV = document.createElement("p");

question\_DIV.classList.add("question");

question\_DIV.innerHTML = i.question;

div.appendChild(question\_DIV);

div.innerHTML += `

<button class="option-div" onclick="checker(this)">${i.options[0]}</button>

<button class="option-div" onclick="checker(this)">${i.options[1]}</button>

<button class="option-div" onclick="checker(this)">${i.options[2]}</button>

<button class="option-div" onclick="checker(this)">${i.options[3]}</button>

<button class="option-div" onclick="checker(this)">${i.options[4]}</button>

`;

quizContainer.appendChild(div);

}

}

function checker(userOption) {

let userSolution = userOption.innerText;

let question =

document.getElementsByClassName("container-mid")[questionCount];

let options = question.querySelectorAll(".option-div");

if (userSolution === quizArray[questionCount].correct) {

userOption.classList.add("correct");

scoreCount++;

} else {

userOption.classList.add("incorrect");

options.forEach((element) => {

if (element.innerText == quizArray[questionCount].correct) {

element.classList.add("correct");

}

});

}

clearInterval(countdown);

options.forEach((element) => {

element.disabled = true;

});

}

function initial() {

quizContainer.innerHTML = "";

questionCount = 0;

scoreCount = 0;

count = 11;

clearInterval(countdown);

timerDisplay();

quizCreator();

quizDisplay(questionCount);

}

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

startScreen.classList.add("hide");

displayContainer.classList.remove("hide");

initial();

});

window.onload = () => {

startScreen.classList.remove("hide");

displayContainer.classList.add("hide");

};

Html:

<!DOCTYPE html>

<html lang="en">

<head>

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

<title>MCQ Quiz</title>

<link

href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;600&display=swap"

rel="stylesheet"

/>

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

</head>

<body>

<div class="start-screen">

<button id="start-button">Start</button>

</div>

<div id="display-container">

<div class="header">

<div class="number-of-count">

<span class="number-of-question">1 of 4 questions</span>

</div>

<div class="timer-div">

<img src="stopwatch.svg" />

<span class="time-left">10s</span>

</div>

</div>

<div id="container">

</div>

<button id="next-button">Next</button>

</div>

<div class="score-container hide">

<div id="user-score">Demo Score</div>

<button id="restart">Restart</button>

</div>

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

</body>

</html>

Style:

\* {

padding: 0;

margin: 0;

box-sizing: border-box;

font-family: "Poppins", sans-serif;

}

body {

background-color: #000000;

}

.start-screen,

.score-container {

position: absolute;

top: 0;

width: 100%;

height: 100%;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

}

button {

border: none;

outline: none;

cursor: pointer;

}

#start-button,

#restart {

font-size: 1.3em;

padding: 0.5em 1.8em;

border-radius: 0.2em;

}

#restart {

margin-top: 0.9em;

}

#display-container {

background-color: #ffffff;

padding: 3.1em 1.8em;

width: 80%;

max-width: 37.5em;

margin: 0 auto;

position: absolute;

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

top: 50%;

left: 50%;

border-radius: 0.6em;

}

.header {

margin-bottom: 1.8em;

display: flex;

justify-content: space-between;

align-items: center;

padding-bottom: 0.6em;

border-bottom: 0.1em solid #c0bfd2;

}

.timer-div {

background-color: #9b9b9b;

width: 7.5em;

border-radius: 1.8em;

display: flex;

align-items: center;

justify-content: space-between;

padding: 0.7em 1.8em;

}

.question {

margin-bottom: 1.25em;

font-weight: 600;

}

.option-div {

font-size: 0.9em;

width: 100%;

padding: 1em;

margin: 0.3em 0;

text-align: left;

outline: none;

background: transparent;

border: 1px solid #c0bfd2;

border-radius: 0.3em;

}

.option-div:disabled {

color: #000000;

cursor: not-allowed;

}

#next-button {

font-size: 1em;

margin-top: 1.5em;

background-color: #000000;

color: #ffffff;

padding: 0.7em 1.8em;

border-radius: 0.3em;

float: right;

}

.hide {

display: none;

}

.incorrect {

background-color: #ffdde0;

color: #d32f2f;

border-color: #d32f2f;

}

.correct {

background-color: #e7f6d5;

color: #689f38;

border-color: #689f38;

}

#user-score {

font-size: 1.5em;

color: #ffffff;

}