PHASE-4 FINAL PROJECT-1 WRITEUP

**Online Test Application.**

DESCRIPTION

The Online Test Application system creates an application that enables users to provide online tests, review them, and display the results.

**System Details**  
This system contains three main modules: Quiz, Review, and Result. The quiz section of the online test application accepts the questions in JSON format. The JSON file can be easily shared from the server in the pre-defined format. The application renders the test at the client-side.  
The “Review and display result” section allows users to declare the results immediately. You can simply call another JSON with the answers in it and evaluate and display the results immediately.

Using the application

* Run the Application in Visual Studio Code.
* Home page will be displayed. Home page contains list of quizzes that user can choose from.
* Based on User’s Choice, Quiz is rendered in the Client Side by fetching data from Quiz Server or JSON file with Questions and Answers.
* User can select option for a question, Users also have the option to navigate through the questions using Navigation Button (First, Last, Next and Prev).
* Users can directly go to First and Last Questions by clicking on Navigation Buttons (First and Last).
* Users have the option to review their Quiz by clicking on Review button. Here user can see details of answered and not answered questions and navigate directly to the question which they want to Answer.
* After completing the quiz, User can submit the quiz where the Quiz Score is displayed on the Results page. As well as user can see their responses are right or wrong.
* There is timer set for quiz, if user exceeds quiz time, then the quiz is automatically submitted and results are displayed to User.

In this program [Quiz App with Timer], there are three layers or boxes, and these boxes shown one by one on a particular button clicked. At first, on the webpage, there is shown a button labelled as “Start Quiz” and when you clicked on that button, then the info box appears with popup animation.

In this info box, there are some rules of the quiz and two buttons labelled as “Exit” and “Continue”. When you clicked on the Exit button, the info box will be hidden but when you clicked on the Continue button, then the Quiz Box appears.

In the Quiz Box, there is a header with a title on the left side and a timer box on the right side. This timer starts decrement from 15 to 0 sec and there is also shown a timeline indicator that is sliding from the left to right side according to the timer. If the user selects an option between 15 to 0 sec, the timer will be stopped and all available options will be disabled.

If the user selected option is correct, the selected option colour, background colour changed to green and there is also shown the tick icon to inform the user that the selected answer is correct. If the user selects an option that is incorrect, the selected option colour, background colour changed to red and there is shown the cross icon to inform the user that the selected option is incorrect and the correct option will be automatically selected.

If the user doesn’t select an option between 15 to 0 sec, the timer will be stopped once it comes in 0 and the correct option of that question will be selected automatically. After that, there is the next button to show the next question, and there is a total of five questions on this Quiz.

In the end, the result box will be appeared and shown the user score and two buttons [Replay Quiz, Quit Quiz], if the user clicked on the replay quiz button, the quiz will again start with the number 1 question, and the score of the user will be 0 but if the user clicked on the quit quiz button, the current window will be reloaded and the quiz starts from the begin.

To create this program (Quiz Application with Timer). First, you need to create four Files one HTML File, CSS File and the other two are JavaScript files. After creating these files just paste the following codes in your file.

First, create an HTML file with the name of index.html and paste the given codes in your HTML file. Remember, you’ve to create a file with .html extension.

Index.html:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Awesome Quiz App </title>

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

<!-- FontAweome CDN Link for Icons -->

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css"/>

</head>

<body>

<!-- start Quiz button -->

<div class="start\_btn"><button>Start Quiz</button></div>

<!-- Info Box -->

<div class="info\_box">

<div class="info-title"><span>Some Rules of this Quiz</span></div>

<div class="info-list">

<div class="info">1. You will have only <span>15 seconds</span> per each question.</div>

<div class="info">2. Once you select your answer, it can't be undone.</div>

<div class="info">3. You can't select any option once time goes off.</div>

<div class="info">4. You can't exit from the Quiz while you're playing.</div>

<div class="info">5. You'll get points on the basis of your correct answers.</div>

</div>

<div class="buttons">

<button class="quit">Exit Quiz</button>

<button class="restart">Continue</button>

</div>

</div>

<!-- Quiz Box -->

<div class="quiz\_box">

<header>

<div class="title">Awesome Quiz Application</div>

<div class="timer">

<div class="time\_left\_txt">Time Left</div>

<div class="timer\_sec">15</div>

</div>

<div class="time\_line"></div>

</header>

<section>

<div class="que\_text">

<!-- Here I've inserted question from JavaScript -->

</div>

<div class="option\_list">

<!-- Here I've inserted options from JavaScript -->

</div>

</section>

<!-- footer of Quiz Box -->

<footer>

<div class="total\_que">

<!-- Here I've inserted Question Count Number from JavaScript -->

</div>

<button class="next\_btn">Next Que</button>

</footer>

</div>

<!-- Result Box -->

<div class="result\_box">

<div class="icon">

<i class="fas fa-crown"></i>

</div>

<div class="complete\_text">You've completed the Quiz!</div>

<div class="score\_text">

<!-- Here I've inserted Score Result from JavaScript -->

</div>

<div class="buttons">

<button class="restart">Replay Quiz</button>

<button class="quit">Quit Quiz</button>

</div>

</div>

<!-- Inside this JavaScript file I've inserted Questions and Options only -->

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

<!-- Inside this JavaScript file I've coded all Quiz Codes -->

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

</body>

</html>

</html>

Second, create a CSS file with the name of style.css and paste the given codes in your CSS file. Remember, you’ve to create a file with .css extension.

style.css

/\* importing google fonts \*/

@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;600;700&display=swap');

\*{

margin: 0;

padding: 0;

box-sizing: border-box;

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

}

body{

background: #007bff;

}

::selection{

color: #fff;

background: #007bff;

}

.start\_btn,

.info\_box,

.quiz\_box,

.result\_box{

position: absolute;

top: 50%;

left: 50%;

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

box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2),

0 6px 20px 0 rgba(0, 0, 0, 0.19);

}

.info\_box.activeInfo,

.quiz\_box.activeQuiz,

.result\_box.activeResult{

opacity: 1;

z-index: 5;

pointer-events: auto;

transform: translate(-50%, -50%) scale(1);

}

.start\_btn button{

font-size: 25px;

font-weight: 500;

color: #007bff;

padding: 15px 30px;

outline: none;

border: none;

border-radius: 5px;

background: #fff;

cursor: pointer;

}

.info\_box{

width: 540px;

background: #fff;

border-radius: 5px;

transform: translate(-50%, -50%) scale(0.9);

opacity: 0;

pointer-events: none;

transition: all 0.3s ease;

}

.info\_box .info-title{

height: 60px;

width: 100%;

border-bottom: 1px solid lightgrey;

display: flex;

align-items: center;

padding: 0 30px;

border-radius: 5px 5px 0 0;

font-size: 20px;

font-weight: 600;

}

.info\_box .info-list{

padding: 15px 30px;

}

.info\_box .info-list .info{

margin: 5px 0;

font-size: 17px;

}

.info\_box .info-list .info span{

font-weight: 600;

color: #007bff;

}

.info\_box .buttons{

height: 60px;

display: flex;

align-items: center;

justify-content: flex-end;

padding: 0 30px;

border-top: 1px solid lightgrey;

}

.info\_box .buttons button{

margin: 0 5px;

height: 40px;

width: 100px;

font-size: 16px;

font-weight: 500;

cursor: pointer;

border: none;

outline: none;

border-radius: 5px;

border: 1px solid #007bff;

transition: all 0.3s ease;

}

.quiz\_box{

width: 550px;

background: #fff;

border-radius: 5px;

transform: translate(-50%, -50%) scale(0.9);

opacity: 0;

pointer-events: none;

transition: all 0.3s ease;

}

.quiz\_box header{

position: relative;

z-index: 2;

height: 70px;

padding: 0 30px;

background: #fff;

border-radius: 5px 5px 0 0;

display: flex;

align-items: center;

justify-content: space-between;

box-shadow: 0px 3px 5px 1px rgba(0,0,0,0.1);

}

.quiz\_box header .title{

font-size: 20px;

font-weight: 600;

}

.quiz\_box header .timer{

color: #004085;

background: #cce5ff;

border: 1px solid #b8daff;

height: 45px;

padding: 0 8px;

border-radius: 5px;

display: flex;

align-items: center;

justify-content: space-between;

width: 145px;

}

.quiz\_box header .timer .time\_left\_txt{

font-weight: 400;

font-size: 17px;

user-select: none;

}

.quiz\_box header .timer .timer\_sec{

font-size: 18px;

font-weight: 500;

height: 30px;

width: 45px;

color: #fff;

border-radius: 5px;

line-height: 30px;

text-align: center;

background: #343a40;

border: 1px solid #343a40;

user-select: none;

}

.quiz\_box header .time\_line{

position: absolute;

bottom: 0px;

left: 0px;

height: 3px;

background: #007bff;

}

section{

padding: 25px 30px 20px 30px;

background: #fff;

}

section .que\_text{

font-size: 25px;

font-weight: 600;

}

section .option\_list{

padding: 20px 0px;

display: block;

}

section .option\_list .option{

background: aliceblue;

border: 1px solid #84c5fe;

border-radius: 5px;

padding: 8px 15px;

font-size: 17px;

margin-bottom: 15px;

cursor: pointer;

transition: all 0.3s ease;

display: flex;

align-items: center;

justify-content: space-between;

}

section .option\_list .option:last-child{

margin-bottom: 0px;

}

section .option\_list .option:hover{

color: #004085;

background: #cce5ff;

border: 1px solid #b8daff;

}

section .option\_list .option.correct{

color: #155724;

background: #d4edda;

border: 1px solid #c3e6cb;

}

section .option\_list .option.incorrect{

color: #721c24;

background: #f8d7da;

border: 1px solid #f5c6cb;

}

section .option\_list .option.disabled{

pointer-events: none;

}

section .option\_list .option .icon{

height: 26px;

width: 26px;

border: 2px solid transparent;

border-radius: 50%;

text-align: center;

font-size: 13px;

pointer-events: none;

transition: all 0.3s ease;

line-height: 24px;

}

.option\_list .option .icon.tick{

color: #23903c;

border-color: #23903c;

background: #d4edda;

}

.option\_list .option .icon.cross{

color: #a42834;

background: #f8d7da;

border-color: #a42834;

}

footer{

height: 60px;

padding: 0 30px;

display: flex;

align-items: center;

justify-content: space-between;

border-top: 1px solid lightgrey;

}

footer .total\_que span{

display: flex;

user-select: none;

}

footer .total\_que span p{

font-weight: 500;

padding: 0 5px;

}

footer .total\_que span p:first-child{

padding-left: 0px;

}

footer button{

height: 40px;

padding: 0 13px;

font-size: 18px;

font-weight: 400;

cursor: pointer;

border: none;

outline: none;

color: #fff;

border-radius: 5px;

background: #007bff;

border: 1px solid #007bff;

line-height: 10px;

opacity: 0;

pointer-events: none;

transform: scale(0.95);

transition: all 0.3s ease;

}

footer button:hover{

background: #0263ca;

}

footer button.show{

opacity: 1;

pointer-events: auto;

transform: scale(1);

}

.result\_box{

background: #fff;

border-radius: 5px;

display: flex;

padding: 25px 30px;

width: 450px;

align-items: center;

flex-direction: column;

justify-content: center;

transform: translate(-50%, -50%) scale(0.9);

opacity: 0;

pointer-events: none;

transition: all 0.3s ease;

}

.result\_box .icon{

font-size: 100px;

color: #007bff;

margin-bottom: 10px;

}

.result\_box .complete\_text{

font-size: 20px;

font-weight: 500;

}

.result\_box .score\_text span{

display: flex;

margin: 10px 0;

font-size: 18px;

font-weight: 500;

}

.result\_box .score\_text span p{

padding: 0 4px;

font-weight: 600;

}

.result\_box .buttons{

display: flex;

margin: 20px 0;

}

.result\_box .buttons button{

margin: 0 10px;

height: 45px;

padding: 0 20px;

font-size: 18px;

font-weight: 500;

cursor: pointer;

border: none;

outline: none;

border-radius: 5px;

border: 1px solid #007bff;

transition: all 0.3s ease;

}

.buttons button.restart{

color: #fff;

background: #007bff;

}

.buttons button.restart:hover{

background: #0263ca;

}

.buttons button.quit{

color: #007bff;

background: #fff;

}

.buttons button.quit:hover{

color: #fff;

background: #007bff;

}

Next, create a JavaScript file with the name of questions.js and paste the given codes in your JavaScript file. Remember, you’ve to create a file with .js extension. In this file, we store all questions in an array.

questions.js:

// creating an array and passing the number, questions, options, and answers

let questions = [

{

numb: 1,

question: "What does HTML stand for?",

answer: "Hyper Text Markup Language",

options: [

"Hyper Text Preprocessor",

"Hyper Text Markup Language",

"Hyper Text Multiple Language",

"Hyper Tool Multi Language"

]

},

{

numb: 2,

question: "What does CSS stand for?",

answer: "Cascading Style Sheet",

options: [

"Common Style Sheet",

"Colorful Style Sheet",

"Computer Style Sheet",

"Cascading Style Sheet"

]

},

{

numb: 3,

question: "What does PHP stand for?",

answer: "Hypertext Preprocessor",

options: [

"Hypertext Preprocessor",

"Hypertext Programming",

"Hypertext Preprogramming",

"Hometext Preprocessor"

]

},

{

numb: 4,

question: "What does SQL stand for?",

answer: "Structured Query Language",

options: [

"Stylish Question Language",

"Stylesheet Query Language",

"Statement Question Language",

"Structured Query Language"

]

},

{

numb: 5,

question: "What does XML stand for?",

answer: "eXtensible Markup Language",

options: [

"eXtensible Markup Language",

"eXecutable Multiple Language",

"eXTra Multi-Program Language",

"eXamine Multiple Language"

]

},

// you can uncomment the below codes and make duplicate as more as you want to add question

// but remember you need to give the numb value serialize like 1,2,3,5,6,7,8,9.....

// {

// numb: 6,

// question: "Your Question is Here",

// answer: "Correct answer of the question is here",

// options: [

// "Option 1",

// "option 2",

// "option 3",

// "option 4"

// ]

// },

];

Last, create a JavaScript file with the name of script.js and paste the given codes in your JavaScript file. Remember, you’ve to create a file with .js extension.

script.js:

//selecting all required elements

const start\_btn = document.querySelector(".start\_btn button");

const info\_box = document.querySelector(".info\_box");

const exit\_btn = info\_box.querySelector(".buttons .quit");

const continue\_btn = info\_box.querySelector(".buttons .restart");

const quiz\_box = document.querySelector(".quiz\_box");

const result\_box = document.querySelector(".result\_box");

const option\_list = document.querySelector(".option\_list");

const time\_line = document.querySelector("header .time\_line");

const timeText = document.querySelector(".timer .time\_left\_txt");

const timeCount = document.querySelector(".timer .timer\_sec");

// if startQuiz button clicked

start\_btn.onclick = ()=>{

info\_box.classList.add("activeInfo"); //show info box

}

// if exitQuiz button clicked

exit\_btn.onclick = ()=>{

info\_box.classList.remove("activeInfo"); //hide info box

}

// if continueQuiz button clicked

continue\_btn.onclick = ()=>{

info\_box.classList.remove("activeInfo"); //hide info box

quiz\_box.classList.add("activeQuiz"); //show quiz box

showQuetions(0); //calling showQestions function

queCounter(1); //passing 1 parameter to queCounter

startTimer(15); //calling startTimer function

startTimerLine(0); //calling startTimerLine function

}

let timeValue = 15;

let que\_count = 0;

let que\_numb = 1;

let userScore = 0;

let counter;

let counterLine;

let widthValue = 0;

const restart\_quiz = result\_box.querySelector(".buttons .restart");

const quit\_quiz = result\_box.querySelector(".buttons .quit");

// if restartQuiz button clicked

restart\_quiz.onclick = ()=>{

quiz\_box.classList.add("activeQuiz"); //show quiz box

result\_box.classList.remove("activeResult"); //hide result box

timeValue = 15;

que\_count = 0;

que\_numb = 1;

userScore = 0;

widthValue = 0;

showQuetions(que\_count); //calling showQestions function

queCounter(que\_numb); //passing que\_numb value to queCounter

clearInterval(counter); //clear counter

clearInterval(counterLine); //clear counterLine

startTimer(timeValue); //calling startTimer function

startTimerLine(widthValue); //calling startTimerLine function

timeText.textContent = "Time Left"; //change the text of timeText to Time Left

next\_btn.classList.remove("show"); //hide the next button

}

// if quitQuiz button clicked

quit\_quiz.onclick = ()=>{

window.location.reload(); //reload the current window

}

const next\_btn = document.querySelector("footer .next\_btn");

const bottom\_ques\_counter = document.querySelector("footer .total\_que");

// if Next Que button clicked

next\_btn.onclick = ()=>{

if(que\_count < questions.length - 1){ //if question count is less than total question length

que\_count++; //increment the que\_count value

que\_numb++; //increment the que\_numb value

showQuetions(que\_count); //calling showQestions function

queCounter(que\_numb); //passing que\_numb value to queCounter

clearInterval(counter); //clear counter

clearInterval(counterLine); //clear counterLine

startTimer(timeValue); //calling startTimer function

startTimerLine(widthValue); //calling startTimerLine function

timeText.textContent = "Time Left"; //change the timeText to Time Left

next\_btn.classList.remove("show"); //hide the next button

}else{

clearInterval(counter); //clear counter

clearInterval(counterLine); //clear counterLine

showResult(); //calling showResult function

}

}

// getting questions and options from array

function showQuetions(index){

const que\_text = document.querySelector(".que\_text");

//creating a new span and div tag for question and option and passing the value using array index

let que\_tag = '<span>'+ questions[index].numb + ". " + questions[index].question +'</span>';

let option\_tag = '<div class="option"><span>'+ questions[index].options[0] +'</span></div>'

+ '<div class="option"><span>'+ questions[index].options[1] +'</span></div>'

+ '<div class="option"><span>'+ questions[index].options[2] +'</span></div>'

+ '<div class="option"><span>'+ questions[index].options[3] +'</span></div>';

que\_text.innerHTML = que\_tag; //adding new span tag inside que\_tag

option\_list.innerHTML = option\_tag; //adding new div tag inside option\_tag

const option = option\_list.querySelectorAll(".option");

// set onclick attribute to all available options

for(i=0; i < option.length; i++){

option[i].setAttribute("onclick", "optionSelected(this)");

}

}

// creating the new div tags which for icons

let tickIconTag = '<div class="icon tick"><i class="fas fa-check"></i></div>';

let crossIconTag = '<div class="icon cross"><i class="fas fa-times"></i></div>';

//if user clicked on option

function optionSelected(answer){

clearInterval(counter); //clear counter

clearInterval(counterLine); //clear counterLine

let userAns = answer.textContent; //getting user selected option

let correcAns = questions[que\_count].answer; //getting correct answer from array

const allOptions = option\_list.children.length; //getting all option items

if(userAns == correcAns){ //if user selected option is equal to array's correct answer

userScore += 1; //upgrading score value with 1

answer.classList.add("correct"); //adding green color to correct selected option

answer.insertAdjacentHTML("beforeend", tickIconTag); //adding tick icon to correct selected option

console.log("Correct Answer");

console.log("Your correct answers = " + userScore);

}else{

answer.classList.add("incorrect"); //adding red color to correct selected option

answer.insertAdjacentHTML("beforeend", crossIconTag); //adding cross icon to correct selected option

console.log("Wrong Answer");

for(i=0; i < allOptions; i++){

if(option\_list.children[i].textContent == correcAns){ //if there is an option which is matched to an array answer

option\_list.children[i].setAttribute("class", "option correct"); //adding green color to matched option

option\_list.children[i].insertAdjacentHTML("beforeend", tickIconTag); //adding tick icon to matched option

console.log("Auto selected correct answer.");

}

}

}

for(i=0; i < allOptions; i++){

option\_list.children[i].classList.add("disabled"); //once user select an option then disabled all options

}

next\_btn.classList.add("show"); //show the next button if user selected any option

}

function showResult(){

info\_box.classList.remove("activeInfo"); //hide info box

quiz\_box.classList.remove("activeQuiz"); //hide quiz box

result\_box.classList.add("activeResult"); //show result box

const scoreText = result\_box.querySelector(".score\_text");

if (userScore > 3){ // if user scored more than 3

//creating a new span tag and passing the user score number and total question number

let scoreTag = '<span>and congrats! , You got <p>'+ userScore +'</p> out of <p>'+ questions.length +'</p></span>';

scoreText.innerHTML = scoreTag; //adding new span tag inside score\_Text

}

else if(userScore > 1){ // if user scored more than 1

let scoreTag = '<span>and nice , You got <p>'+ userScore +'</p> out of <p>'+ questions.length +'</p></span>';

scoreText.innerHTML = scoreTag;

}

else{ // if user scored less than 1

let scoreTag = '<span>and sorry , You got only <p>'+ userScore +'</p> out of <p>'+ questions.length +'</p></span>';

scoreText.innerHTML = scoreTag;

}

}

function startTimer(time){

counter = setInterval(timer, 1000);

function timer(){

timeCount.textContent = time; //changing the value of timeCount with time value

time--; //decrement the time value

if(time < 9){ //if timer is less than 9

let addZero = timeCount.textContent;

timeCount.textContent = "0" + addZero; //add a 0 before time value

}

if(time < 0){ //if timer is less than 0

clearInterval(counter); //clear counter

timeText.textContent = "Time Off"; //change the time text to time off

const allOptions = option\_list.children.length; //getting all option items

let correcAns = questions[que\_count].answer; //getting correct answer from array

for(i=0; i < allOptions; i++){

if(option\_list.children[i].textContent == correcAns){ //if there is an option which is matched to an array answer

option\_list.children[i].setAttribute("class", "option correct"); //adding green color to matched option

option\_list.children[i].insertAdjacentHTML("beforeend", tickIconTag); //adding tick icon to matched option

console.log("Time Off: Auto selected correct answer.");

}

}

for(i=0; i < allOptions; i++){

option\_list.children[i].classList.add("disabled"); //once user select an option then disabled all options

}

next\_btn.classList.add("show"); //show the next button if user selected any option

}

}

}

function startTimerLine(time){

counterLine = setInterval(timer, 29);

function timer(){

time += 1; //upgrading time value with 1

time\_line.style.width = time + "px"; //increasing width of time\_line with px by time value

if(time > 549){ //if time value is greater than 549

clearInterval(counterLine); //clear counterLine

}

}

}

function queCounter(index){

//creating a new span tag and passing the question number and total question

let totalQueCounTag = '<span><p>'+ index +'</p> of <p>'+ questions.length +'</p> Questions</span>';

bottom\_ques\_counter.innerHTML = totalQueCounTag; //adding new span tag inside bottom\_ques\_counter

}

Conclusion:

That’s all, now you’ve successfully created a Create a Quiz App with Timer using HTML CSS & JavaScript.

**Pushing the code to your GitHub repositories:**

* Open your command prompt and navigate to the folder where you have created your files.

**cd <folder path>**

* Initialize your repository using the following command:

**git init**

* Add all the files to your git repository using the following command:

**git add .**

* Commit the changes using the following command:

**git commit . -m “Changes have been committed.”**

* Push the files to the folder you initially created using the following command:

**git push -u origin master**