**Online Test Application**

**Code**

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Online Test Application</title>

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

</head>

<body>

    <div class="container">

        <div class="leftSide">

           <img src="https://d1ymz67w5raq8g.cloudfront.net/Pictures/480xany/6/5/5/509655\_shutterstock\_1506580442\_769367.jpg"alt="quiz logo"class="tilt">

        </div>

    <div class="rightSide"><br>

        <h1>Welcome to Online Test Application</h1><br>

        <ul style="list-style-type:circle">

            <li><b>Consist of 5 questions</b></li>

        </ul>

        <a href="questions.html">Start Quiz</a>

        </div>

    </div>

</body>

<script src="js/jquery-3.5.1.min.js"></script>

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

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

</html>

**Questions.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Questions</title>

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

</head>

<body>

    <div class="box" id="questionScreen">

        <div class="title">

            Online Test Application

        </div>

        <div class="header">

            <div class="scoreBox" style="color:white; background-color: rgb(150, 198, 226);">Score: <span></span> </div>

        </div>

        <div class="questionBox">

        </div>

        <div class="optionBox" >

            <span onclick="checkAnswer(this)" data-opt="1"></span>

            <span onclick="checkAnswer(this)" data-opt="2"></span>

            <span onclick="checkAnswer(this)" data-opt="3"></span>

            <span onclick="checkAnswer(this)" data-opt="4"></span>

        </div>

        <div class="footer">

            <button onclick="showNext()">

                Next

            </button>

            <button onclick="showResult(1)">

                 Result

            </button>

        </div>

    </div>

    <div class="box" id="resultScreen" style="display: none;">

        <div class="title">

            Online Test Result

        </div>

        <div class="resultBox">

            <label>Questions : </label>

            <span id="titalQuestions">5</span>

            <label>Attempted : </label>

            <span id="attemptQuestion">0</span>

            <label>Correct : </label>

            <span id="correctAnswers">0</span>

            <label>Wrong : </label>

            <span id="wrongAnswers">0</span>

        </div>

        <div class="buttonBox">

            <a href="index.html">Start Again</a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

            <a href="review.html">Review Solutions</a>

        </div>

    </div>

</body>

<script src="js/jquery-3.5.1.min.js"></script>

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

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

</html>

**Review.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Review </title>

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

</head>

<body>

    <div class="container1">

        <div class="title">

        <h1><b>Solutions</b></h1></div>

        <h3>1. Which of the following JavaScript cannot do?</h3>

        <h4>Ans. All of the Above</h4>

        <br/>

        <br/>

        <h3>2. \_\_\_\_\_\_\_\_\_ keyword is used to declare variables in javascript.</h3>

        <h4>Ans. Var</h4>

        <br/>

        <br/>

        <h3>3. In JavaScript the x===y statement implies that:</h3>

        <h4>Ans. Both are equal in the value and data type.</h4>

        <br/>

        <br/>

        <h3>4. Whats so great about XML?</h3>

        <h4>Ans. Both</h4>

        <br/>

        <br/>

        <h3>5. In the JavaScript, which one of the following is not considered as an error:</h3>

        <h4>Ans. Division by zero</h4>

        <br>

        <br>

        <div class="buttonBox">

            <a href="index.html">Back to Home</a></div>

    </div>

</body>

</html>

**Style.css**

@import url("https://fonts.googleapis.com/css2?family=Open+Sans:wght@300&display=swap");

\*{

    margin: 0;

    padding:0;

    box-sizing: border-box;

}

body {

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

    background-image: url("https://media.istockphoto.com/photos/dark-blue-stained-grungy-background-or-texture-picture-id1132593892?k=6&m=1132593892&s=612x612&w=0&h=kdFpYAbe0jAnckEYkZdNWUVYbNdKXGxHa1rd8joRZEg=");

    background-size: cover;

    display: flex;

    justify-content: center;

    align-items: center;

    min-height: 100vh;

}

.container{

    width: 750px;

    background-color: rgb(102, 127, 251);

    box-shadow:0 0 50px 0 rgba(0,0, 0, 0.2);

    min-height: 350px;

    padding:50px 50px ;

    border-radius: 10px;

    display:flex;

}

.leftSide,

.rightSide{

    width: 50%;

}

.leftSide img{

    height:300px;

    width:300px;

}

.rightSide h1{

    color:#201414;

    font-size: 32px;

}

.rightSide h2{

    color: #444 ;

    margin:20px auto 10px;

    font-size:25px;

}

.rightSide ul{

    color: #444 ;

    font-size:18px;

    margin-top:600;

    list-style-type: circle;

    list-style-position: inside;

}

.rightSide ul li{

    margin-top: 5px;

}

.rightSide a{

    background-color: #dbd9d9;

    border-radius: 100px;

    color:rgb(46, 40, 40);

    font-weight:600;

    width:100%;

    display:inline-block;

    text-align: center;

    padding:15px 0;

    margin-top:25px;

    text-decoration: none;

    outline:none;

}

.box{

    background-color:antiquewhite;

    border-radius: 10px;

    box-shadow:0 0 50px 0 rgba(0,0, 0, 0.2);

    min-height: 350px;

    width:540px;

    padding:50px;

}

.title{

    border-bottom: 1px solid #464646;

    color:#464646;

    padding-bottom:10px;

    margin-bottom:20px;

    font-weight:600;

    font-size:24px;

}

.optionBox span{

    background-color:rgb(219, 222, 223);

    border-radius: 10px;

    color:rgb(32, 28, 28);

    border:1px solid,#444;

    padding:10px 15px;

}

.header{

    margin-bottom: 30px;

    display:flex;

    justify-content:space-between ;

}

.scoreBox{

    border-radius: 100px;

    padding:10px 15px;

    border:1px solid #444;

    color:#444;

}

.questionBox{

    background-color: rgb(219, 222, 223);

    color:rgb(32, 28, 28);

    border-radius: 10px;

    padding:10px 15px;

}

.optionBox{

    display: grid;

    grid-template-columns: 1fr 1fr;

    margin: 30px 0;

    grid-gap:15px;

}

.footer{

    display: flex;

    justify-content: space-between;

}

.footer button{

    background-color: #1da3dd;

    border-radius: 5px;

    padding:7px 15px;

    color:#fff;

    border:0;

    outline:none;

    font-size: 20px;

}

.resultBox{

    margin-bottom: 30px;

    display: grid;

    grid-template-columns: 1fr 1fr;

    grid-row-gap:15px;

    font-size:20px

}

.resultBox \*:nth-child(odd){

    text-align:right;

}

.resultBox span{

    font-weight: 600;

}

.buttonBox{

    text-align:center;

}

.buttonBox a{

    background-color:#1da3dd;

    border-radius: 50px;

    border:0;

    text-decoration: none;

    color:#fff;

    outline: none;

    padding: 7px 15px;

    margin-top: 30px;

    display: inline-block;

}

.optionBox span.right{

    background-color: rgb(74, 206, 57);

    border-color: chartreuse;

    color:cornsilk;

}

.optionBox span.wrong{

    background-color: crimson;

    border-color:crimson;

    color:cyan;

}

.container1{

    background-color: antiquewhite;

    padding:50px;

    border-radius: 10px;

}

span:hover{

    background-color: rgb(241, 241, 79);

}

@media screen and (max-width:768px){

    .leftSide{

        display: none;

    }

    .container{

        width: 425px;

    }

    .rightSide{

        width:100px;

    }

    .optionBox{

        grid-template-columns: 1fr;

    }

}

**Quiz.js**

let quiz=[

    {

        question:"Which of the following JavaScript cannot do?",

        option:[

            "1.JavaScript can react to events",

            "2.JavaScript can manipulate HTML elements",

            "3.JavaScript can be use to validate data",

            "4.All of the Above",

        ],

        answer:4,

    },

    {

        question:" \_\_\_\_\_\_\_\_\_ keyword is used to declare variables in javascript.",

        option:[

            "1.Var",

            "2.Dim",

            "3.String",

            "4.None of the Above",

        ],

        answer:1,

    },

    {

        question:"In JavaScript the x===y statement implies that:",

        option:[

            "1.Both x and y are equal in value, type and reference address as well.",

            "2.Both are x and y are equal in value only.",

            "3.Both are equal in the value and data type.",

           "4.Both are not same at all.",

        ],

        answer:3,

    },

    {

        question:"Whats so great about XML?",

        option:[

            "1.Easy data exchange",

            "2.High speed on network ",

            "3.Both",

            "4.None",

        ],

        answer:3,

    },

    {

        question:"In the JavaScript, which one of the following is not considered as an error:",

        option:[

            "1.Syntax error",

            "2.Missing of semicolons",

            "3.Division by zero",

            "4.Missing of Bracket",

        ],

        answer:3,

    }

]

**Script.js**

let index=0;

let attempt= 0;

let score = 0;

let wrong =0;

let questions=quiz.sort(function(){

   return 0.5 - Math.random();

});

let totalQuestion = questions.length;

$(function(){

  //printing question

  printQuestion(index);

});

//this is the function to print question part

function printQuestion(i){

    $(".questionBox").text(questions[i].question);

    $(".optionBox span").eq(0).text(questions[i].option[0]);

    $(".optionBox span").eq(1).text(questions[i].option[1]);

    $(".optionBox span").eq(2).text(questions[i].option[2]);

    $(".optionBox span").eq(3).text(questions[i].option[3]);

}

function checkAnswer(option) {

  attempt ++;

  let optionClicked = $(option).data("opt");

  if(optionClicked == questions[index].answer) {

    $(option).addClass("right");

    score++;

  }

  else{

    $(option).addClass("wrong");

    wrong++;

  }

  $(".scoreBox span").text(score);

  $(".optionBox span").attr("onclick","");

}

function showNext(){

  if(index >= questions.length-1) {

    showResult(0);

    return;

  }

  index++;

  $(".optionBox span").removeClass();

  $(".optionBox span").attr("onclick","checkAnswer(this)");

  printQuestion(index);

}

function showResult(j){

  if(j==1 && index < questions.length-1 && !confirm(

      "Test is not yet finished. Press OK to end the test")

  ) {

    return;

  }

result();

}

function result() {

  $("#questionScreen").hide();

  $("#resultScreen").show();

  $("#totalQuestion").text(totalQuestion);

  $("#attemptQuestion").text(attempt);

  $("#correctAnswers").text(score);

  $("#wrongAnswers").text(wrong);

}

**Tilt.min.js**

"use strict"; var \_typeof = "function" == typeof Symbol && "symbol" == typeof Symbol.iterator ? function (t) {

    return typeof t

}

    : function (t) {

        return t && "function" == typeof Symbol && t.constructor === Symbol && t !== Symbol.prototype ? "symbol" : typeof t

    };

!function (t) {

    "function" == typeof define && define.amd ? define(["jquery"], t) : "object" === ("undefined" == typeof module ? "undefined" : \_typeof(module)) && module.exports ? module.exports = function (i, s) {

        return void 0 === s && (s = "undefined" != typeof window ? require("jquery") : require("jquery")(i)), t(s), s

    } :

        t(jQuery)

}

    (

        function (t) {

            return t.fn.tilt = function (i) {

                var s = function () {

                    this.ticking || (requestAnimationFrame(g.bind(this)), this.ticking = !0)

                },

                    e = function () {

                        var i = this; t(this).on("mousemove", o), t(this).on("mouseenter", a), this.settings.reset && t(this).on("mouseleave", l), this.settings.glare && t(window).on("resize", d.bind(i))

                    },

                    n = function () {

                        var i = this; void 0 !== this.timeout && clearTimeout(this.timeout), t(this).css({ transition: this.settings.speed + "ms " + this.settings.easing }),

                            this.settings.glare && this.glareElement.css(

                                {

                                    transition: "opacity " + this.settings.speed + "ms " + this.settings.easing

                                }),

                            this.timeout = setTimeout(function () {

                                t(i).css({ transition: "" }), i.settings.glare && i.glareElement.css({

                                    transition: ""

                                })

                            },

                                this.settings.speed)

                    },

                    a = function (i) {

                        this.ticking = !1, t(this).css({

                            "will-change": "transform"

                        }),

                            n.call(this), t(this).trigger("tilt.mouseEnter")

                    },

                    r = function (i) {

                        return "undefined" == typeof i && (i = { pageX: t(this).offset().left + t(this).outerWidth() / 2, pageY: t(this).offset().top + t(this).outerHeight() / 2 }), { x: i.pageX, y: i.pageY }

                    }, o = function (t) { this.mousePositions = r(t), s.call(this) }, l = function () { n.call(this), this.reset = !0, s.call(this), t(this).trigger("tilt.mouseLeave") }, h = function () { var i = t(this).outerWidth(), s = t(this).outerHeight(), e = t(this).offset().left, n = t(this).offset().top, a = (this.mousePositions.x - e) / i, r = (this.mousePositions.y - n) / s, o = (this.settings.maxTilt / 2 - a \* this.settings.maxTilt).toFixed(2), l = (r \* this.settings.maxTilt - this.settings.maxTilt / 2).toFixed(2), h = Math.atan2(this.mousePositions.x - (e + i / 2), -(this.mousePositions.y - (n + s / 2))) \* (180 / Math.PI); return { tiltX: o, tiltY: l, percentageX: 100 \* a, percentageY: 100 \* r, angle: h } }, g = function () { return this.transforms = h.call(this), this.reset ? (this.reset = !1, t(this).css("transform", "perspective(" + this.settings.perspective + "px) rotateX(0deg) rotateY(0deg)"), void (this.settings.glare && (this.glareElement.css("transform", "rotate(180deg) translate(-50%, -50%)"), this.glareElement.css("opacity", "0")))) : (t(this).css("transform", "perspective(" + this.settings.perspective + "px) rotateX(" + ("x" === this.settings.disableAxis ? 0 : this.transforms.tiltY) + "deg) rotateY(" + ("y" === this.settings.disableAxis ? 0 : this.transforms.tiltX) + "deg) scale3d(" + this.settings.scale + "," + this.settings.scale + "," + this.settings.scale + ")"), this.settings.glare && (this.glareElement.css("transform", "rotate(" + this.transforms.angle + "deg) translate(-50%, -50%)"), this.glareElement.css("opacity", "" + this.transforms.percentageY \* this.settings.maxGlare / 100)), t(this).trigger("change", [this.transforms]), void (this.ticking = !1)) }, c = function () { var i = this.settings.glarePrerender; if (i || t(this).append('<div class="js-tilt-glare"><div class="js-tilt-glare-inner"></div></div>'), this.glareElementWrapper = t(this).find(".js-tilt-glare"), this.glareElement = t(this).find(".js-tilt-glare-inner"), !i) { var s = { position: "absolute", top: "0", left: "0", width: "100%", height: "100%" }; this.glareElementWrapper.css(s).css({ overflow: "hidden", "pointer-events": "none" }), this.glareElement.css({ position: "absolute", top: "50%", left: "50%", "background-image": "linear-gradient(0deg, rgba(255,255,255,0) 0%, rgba(255,255,255,1) 100%)", width: "" + 2 \* t(this).outerWidth(), height: "" + 2 \* t(this).outerWidth(), transform: "rotate(180deg) translate(-50%, -50%)", "transform-origin": "0% 0%", opacity: "0" }) } }, d = function () { this.glareElement.css({ width: "" + 2 \* t(this).outerWidth(), height: "" + 2 \* t(this).outerWidth() }) }; return t.fn.tilt.destroy = function () { t(this).each(function () { t(this).find(".js-tilt-glare").remove(), t(this).css({ "will-change": "", transform: "" }), t(this).off("mousemove mouseenter mouseleave") }) }, t.fn.tilt.getValues = function () { var i = []; return t(this).each(function () { this.mousePositions = r.call(this), i.push(h.call(this)) }), i }, t.fn.tilt.reset = function () { t(this).each(function () { var i = this; this.mousePositions = r.call(this), this.settings = t(this).data("settings"), l.call(this), setTimeout(function () { i.reset = !1 }, this.settings.transition) }) }, this.each(function () {

                        var s = this; this.settings = t.extend({ maxTilt: t(this).is("[data-tilt-max]") ? t(this).data("tilt-max") : 20, perspective: t(this).is("[data-tilt-perspective]") ? t(this).data("tilt-perspective") : 300, easing: t(this).is("[data-tilt-easing]") ? t(this).data("tilt-easing") : "cubic-bezier(.03,.98,.52,.99)", scale: t(this).is("[data-tilt-scale]") ? t(this).data("tilt-scale") : "1", speed: t(this).is("[data-tilt-speed]") ? t(this).data("tilt-speed") : "400", transition: !t(this).is("[data-tilt-transition]") || t(this).data("tilt-transition"), disableAxis: t(this).is("[data-tilt-disable-axis]") ? t(this).data("tilt-disable-axis") : null, axis: t(this).is("[data-tilt-axis]") ? t(this).data("tilt-axis") : null, reset: !t(this).is("[data-tilt-reset]") || t(this).data("tilt-reset"), glare: !!t(this).is("[data-tilt-glare]") && t(this).data("tilt-glare"), maxGlare: t(this).is("[data-tilt-maxglare]") ? t(this).data("tilt-maxglare") : 1 }, i), null !== this.settings.axis && (console.warn("Tilt.js: the axis setting has been renamed to disableAxis. See https://github.com/gijsroge/tilt.js/pull/26 for more information"), this.settings.disableAxis = this.settings.axis), this.init = function () { t(s).data("settings", s.settings), s.settings.glare && c.call(s), e.call(s) },

                            this.init()

                    })

            }, t("[data-tilt]").tilt(), !0

        });