

Online Test Application(Source Code)

Developed By:G Rajya Lakshmi Suvarna HimaValli

Style.css:

```
@import url('https://fonts.googleapis.com/css?family=Montserrat:400,500,700,800&display=swap');
@import url('https://fonts.googleapis.com/css?family=Concert+One&display=swap');
body {
    margin: 0;
    padding: 0;
    background-attachment: fixed;
    font-family: 'Montserrat', sans-serif;
    overflow-x: hidden;
}

* {
    box-sizing: border-box;
}

.custom-box {
    max-width: 800px;
    background-color: #ffffff;
    margin: 30px auto;
    padding: 50px 40px;
    border-radius: 10px;
}

.test-home-box {
    text-align: center;
    display: none;
    margin-top: 13%;
    padding-top: 20px;
}
```

```
.test-home-box.show {  
  display: block;  
  animation: fadeInRight 1s ease;  
}
```

```
.test-home-box h4 {  
  font-size: 30px;  
  font-weight: 400;  
  color: #444444;  
  margin: 0 0 30px;  
}
```

```
.btn {  
  background-color: cornflowerblue;  
  border: 2px solid cornflowerblue;  
  padding: 14px 40px;  
  color: #ffffff;  
  font-size: 20px;  
  border-radius: 30px;  
  font-family: 'Montserrat', sans-serif;  
  cursor: pointer;  
  white-space: nowrap;  
  margin: 0 10px;  
}
```

```
.btn:focus {  
  outline: none;  
}
```

```
.test-box {  
  background-color: lavender;  
  display: none;  
  padding-top: 20px;  
  padding-bottom: 30px;  
}
```

```
.test-box.show {  
  display: block;  
  animation: fadeInRight 1s ease;  
}
```

```
.test-app-name {  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  color: black;  
}
```

```
.test-box .stats {  
  display: flex;  
  flex-wrap: wrap;  
  justify-content: space-between;  
  font-size: 30px;  
  text-transform: uppercase;  
  color: #000000;  
}
```

```
.test-box .stats .test-time {  
  flex-basis: calc(50% - 20px);  
  max-width: calc(50% - 20px);  
  text-align: left;  
}
```

```
.test-box .stats .test-time .time-up-text {  
  color: #d23723;  
  font-weight: 500;  
  display: none;  
}
```

```
.test-box .stats .test-time .time-up-text.show {
```

```
display: inline-block;
animation: fadeInOut 1s linear infinite;
}
```

```
@keyframes fadeInOut {
  0% {
    opacity: 0;
  }
  50% {
    opacity: 1;
  }
  100% {
    opacity: 0;
  }
}
```

```
.test-box .stats .test-time .remaining-time {
  height: 60px;
  width: 60px;
  color: #059e4c;
  border: 2px solid #059e4c;
  font-weight: 800;
  line-height: 56px;
  border-radius: 50%;
  text-align: center;
  display: inline-block;
}
```

```
.test-box .stats .test-time .remaining-time.less-time {
  color: #d23723;
  border-color: #d23723;
}
```

```
.test-box .stats .score-board {
  flex-basis: calc(50% - 100px);
}
```

```
max-width: calc(50% - 100px);
display: flex;
justify-content: space-evenly;
align-items: center;
}
```

```
.test-box .stats .score-board .correct-answers {
  font-weight: 800;
}
```

```
.test-box .question-box {
  background-color: #ffffff;
  padding: 40px 30px;
  margin-top: 40px;
  border-radius: 10px;
  font-size: 28px;
  border: 2px solid #734F96;
  text-align: center;
  position: relative;
}
```

```
.test-box .question-box .current-question-num {
  height: 70px;
  width: 70px;
  background-color: #ffffff;
  border: 2px solid #734F96;
  font-size: 25px;
  font-weight: 800;
  border-radius: 50%;
  line-height: 68px;
  text-align: center;
  position: absolute;
  top: -35px;
  left: 50%;
  margin-left: -35px;
}
```

```
    z-index: 1;
}

.test-box .question-box .question-text {
    font-size: 30px;
    font-family: 'Concert One', cursive;
    color: #665b49;
}

.test-box .option-box {
    display: flex;
    flex-wrap: wrap;
    justify-content: space-between;
}

.test-box .option-box .option {
    background-color: #ffffff;
    flex-basis: calc(50% - 20px);
    max-width: calc(50% - 20px);
    margin-top: 30px;
    padding: 14px 15px;
    text-align: center;
    font-size: 20px;
    text-transform: uppercase;
    font-weight: 500;
    color: #796746;
    border: 2px solid #734F96;
    border-radius: 30px;
    cursor: pointer;
    position: relative;
    opacity: 0;
    animation: zoomIn .3s linear forwards;
}
```

```
@keyframes zoomIn {
```

```
from {
  transform: scale(0.8);
}
to {
  transform: scale(1);
  opacity: 1;
}
}
```

```
.test-box .option-box .option.already-answered {
  pointer-events: none;
}
```

```
.test-box .option-box .option.show-correct {
  background-color: #11c466;
  border-color: #059e4c;
  color: #ffffff;
  transition: all 0.3s ease;
}
```

```
.test-box .option-box .option.correct {
  background-color: #11c466;
  border-color: #059e4c;
  color: #ffffff;
  opacity: 1;
  animation: pulse 1s linear;
}
```

```
@keyframes pulse {
  0% {
    transform: scale(1);
  }
  50% {
    transform: scale(1.1);
  }
}
```

```
100% {  
    transform: scale(1);  
}  
}  
  
.test-box .option-box .option.correct::before {  
    content: "";  
    position: absolute;  
    height: 30px;  
    width: 30px;  
    right: 15px;  
    top: 10px;  
    background-image: url("../img/correct.png");  
    background-size: 20px;  
    background-position: center;  
    background-repeat: no-repeat;  
    animation: fadeInRight .5s ease;  
}  
  
.test-box .option-box .option.wrong {  
    background-color: #f85943;  
    border-color: #d23723;  
    color: #ffffff;  
    opacity: 1;  
    animation: shake 1s linear;  
}  
  
.test-box .option-box .option.wrong::before {  
    content: "";  
    position: absolute;  
    height: 30px;  
    width: 30px;  
    right: 15px;  
    top: 10px;  
    background-image: url("../img/wrong.png");
```



```
background-size: 20px;  
background-position: center;  
background-repeat: no-repeat;  
animation: fadeInRight .5s ease;  
}
```

```
@keyframes shake {  
  0%,  
  30%,  
  50% {  
    transform: translateX(10px);  
  }  
  20%,  
  40% {  
    transform: translateX(-10px);  
  }  
  60% {  
    transform: translateX(-7px);  
  }  
  70% {  
    transform: translateX(7px);  
  }  
  80% {  
    transform: translateX(-4px);  
  }  
  90% {  
    transform: translateX(4px);  
  }  
  100% {  
    transform: translateX(0px);  
  }  
}
```

```
@keyframes fadeInRight {  
  from {
```

```
    opacity: 0;
    transform: translateX(40px);
  }
  to {
    opacity: 1;
    transform: translateX(0px);
  }
}
```

```
.test-box .next-question {
  margin-top: 20px;
  text-align: center;
}
```

```
.next-question-btn,
.see-result-btn {
  display: none;
}
```

```
.next-question-btn.show,
.see-result-btn.show {
  display: inline-block;
  animation: fadeInRight 1s ease;
}
```

```
.test-over-box {
  text-align: center;
  display: none;
  margin-top: 5%;
}
```

```
.test-over-box.show {
  display: block;
  animation: fadeInRight 1s ease;
}
```

```
.test-over-box h1 {  
  font-size: 50px;  
  font-family: 'Concert One', cursive;  
  color: #f85943;  
  margin: 0 0 20px;  
}
```

```
.test-over-box h4 {  
  font-size: 25px;  
  font-weight: normal;  
  color: #444444;  
  margin: 15px 0 20px;  
}
```

```
.test-over-box h4 span {  
  font-weight: 800;  
  color: #111111;  
}
```

```
.start-again-test-btn,  
.go-to-home-btn {  
  margin-top: 15px;  
}
```

```
@keyframes line-anime {  
  to {  
    stroke-dashoffset: 0;  
    /* stroke-dasharray: 0; */  
  }  
}
```

```
@keyframes fill {  
  from {
```

```
    fill: transparent;
  }
  to {
    fill: white;
  }
}
```

```
/*----- Responsive -----*/
```

```
@media only screen and (max-width: 767px) {
  body {
    overflow-y: scroll;
  }
  .custom-box {
    padding: 10px 15px 20px 15px;
    margin: 15px 10px 0 10px;
  }
  .test-home-box {
    margin-top: 60%;
  }
  .test-home-box h4 {
    font-size: 25px;
  }
  .test-box .question-box {
    padding: 35px 15px;
    margin-bottom: 30px;
  }
  .test-box .option-box {
    flex-direction: column;
  }
  .test-box .option-box .option {
    flex-basis: 100%;
    max-width: 100%;
    margin-top: 10px;
  }
}
```

```

}
.test-box .stats {
  font-size: 20px;
}
.test-box .stats .test-time .remaining-time {
  height: 50px;
  width: 50px;
  line-height: 46px;
  font-weight: 600;
}
.test-box .stats .test-time {
  flex-basis: calc(60% - 10px);
  max-width: calc(60% - 10px);
}
.test-box .stats .score-board {
  flex-basis: calc(40% - 10px);
  max-width: calc(40% - 10px);
}
.test-over-box {
  margin-top: 25%;
}
.loader {
  margin-top: 50%;
}
}

```

Script.js:

```

const questionText = document.querySelector('.question-text');
const optionBox = document.querySelector('.option-box');
const currentQuestionNum = document.querySelector('.current-question-num');
const nextQuestionBtn = document.querySelector('.next-question-btn');
const correctAnswers = document.querySelector('.correct-answers');
const seeResultsBtn = document.querySelector('.see-result-btn');
const remainingTime = document.querySelector('.remaining-time');

```

```

const timeUpText = document.querySelector('.time-up-text');
const testBox = document.querySelector('.test-box');
const testOverBox = document.querySelector('.test-over-box');
const testHomeBox = document.querySelector('.test-home-box');
const startAgainTestBtn = document.querySelector('.start-again-test-btn');
const goHomeBtn = document.querySelector('.go-to-home-btn');
const startTestBtn = document.querySelector('.start-test-btn');

let attempt = 0;
let myArray = [];
let questionIndex = 0;
let score = 0;
let number = 0;
let interval;

let myApp = [];

window.onload = () => {
  fetch("testdata.json")
    .then(function(resp) {
      return resp.json();
    })
    .then(function(data) {
      for (let i = 0; i < data.length; i++) {
        // console.log(data[i]);
        myApp.push({ question: data[i].question, options: data[i].options, answer: data[i].answer });
      }
    });
  testHomeBox.classList.add('show');
};

function generateRandomQuestion() {
  const randomNumber = Math.floor(Math.random() * myApp.length);
  let hitDuplicate = 0;
  if (myArray.length == 0) {

```

```

        questionIndex = randomNumber;
    } else {
        for (let i = 0; i < myArray.length; i++) {
            if (randomNumber == myArray[i]) {
                hitDuplicate = 1;
            }
        }
        if (hitDuplicate == 1) {
            generateRandomQuestion();
            return;
        } else {
            questionIndex = randomNumber;
        }
    }
    myArray.push(randomNumber);
    load();
}

```

```

function resetttest() {
    attempt = 0;
    myArray = [];
    score = 0;
    number = 0;
}

```

```

function timeIsUp() {
    showTimeUpText();
    for (let i = 0; i < optionBox.children.length; i++) {
        if (optionBox.children[i].id == myApp[questionIndex].answer) {
            optionBox.children[i].classList.add("show-correct");
        }
    }
    disableOptions();
    showNextQuestionBtn();
}

```

```
}
```

```
function startTimer() {  
  let timeLimit = 15;  
  remainingTime.innerHTML = timeLimit;  
  remainingTime.classList.remove('less-time');  
  interval = setInterval(() => {  
  
    if (timeLimit < 6) {  
      remainingTime.classList.add('less-time');  
    }  
    if (timeLimit < 10) {  
      timeLimit = "0" + timeLimit;  
    }  
    remainingTime.innerHTML = timeLimit;  
    if (timeLimit == 0) {  
      clearInterval(interval);  
      timesUp();  
    }  
    timeLimit--;  
  }, 1000);  
}
```

```
function stopTimer() {  
  clearInterval(interval);  
}
```

```
function testOver() {  
  nextQuestionBtn.classList.remove("show");  
  seeResultsBtn.classList.add("show");  
}
```

```
function scoreBoard() {  
  correctAnswers.innerHTML = score;  
}
```



```
function showTimeUpText() {  
  timeUpText.classList.add('show');  
}
```

```
function hideTimeUpText() {  
  timeUpText.classList.remove('show');  
}
```

```
function showNextQuestionBtn() {  
  nextQuestionBtn.classList.add('show');  
}
```

```
function hideNextQuestionBtn() {  
  nextQuestionBtn.classList.remove('show');  
}
```

```
function disableOptions() {  
  for (let i = 0; i < optionBox.children.length; i++) {  
    optionBox.children[i].removeAttribute("onclick");  
    optionBox.children[i].classList.add("already-answered");  
  }  
}
```

```
function check(element) {  
  const id = element.id;  
  if (id == myApp[questionIndex].answer) {  
    element.classList.add("correct");  
    score++;  
    scoreBoard();  
  } else {  
    element.classList.add("wrong");  
    for (let i = 0; i < optionBox.children.length; i++) {  
      if (optionBox.children[i].id == myApp[questionIndex].answer) {
```

```

        optionBox.children[i].classList.add("show-correct");
    }
}

attempt++;
disableOptions();
showNextQuestionBtn();
stopTimer();
if (number == myApp.length) {
    testOver();
}
}

function createOptions() {
    optionBox.innerHTML = "";
    let animationDelay = 0.2;
    for (let i = 0; i < myApp[questionIndex].options.length; i++) {
        const option = document.createElement('div');
        option.innerHTML = myApp[questionIndex].options[i];
        option.classList.add("option");
        option.id = i;
        option.style.animationDelay = animationDelay + "s";
        animationDelay = animationDelay + 0.2;
        option.setAttribute("onclick", "check(this)");
        optionBox.appendChild(option);
    }
}

function load() {
    number++;
    scoreBoard();
    questionText.innerHTML = myApp[questionIndex].question;
    createOptions();
    currentQuestionNum.innerHTML = number + '/' + myApp.length;
}

```

```
}
```

```
function nextQuestion() {  
  startTimer();  
  generateRandomQuestion()  
  hideNextQuestionBtn();  
  hideTimeUpText()  
  
}
```

```
function testResult() {  
  document.querySelector('.total-questions').innerHTML = myApp.length;  
  document.querySelector('.total-attempt').innerHTML = attempt;  
  document.querySelector('.total-correct').innerHTML = score;  
  document.querySelector('.total-wrong').innerHTML = attempt - score;  
  const percentage = (score / myApp.length) * 100;  
  document.querySelector('.percentage').innerHTML = percentage.toFixed(2) + "%";  
}  
seeResultsBtn.addEventListener('click', () => {  
  testBox.classList.remove('show');  
  seeResultsBtn.classList.remove('show');  
  testOverBox.classList.add('show');  
  testResult();  
})  
nextQuestionBtn.addEventListener("click", nextQuestion);
```

```
startAgaintestBtn.addEventListener('click', () => {  
  location.replace("./index.html")  
})
```

```
goHomeBtn.addEventListener('click', () => {  
  testOverBox.classList.remove('show');  
  testHomeBox.classList.add('show');  
  resettest();  
})
```

```
starttestBtn.addEventListener('click', () => {  
  testHomeBox.classList.remove('show');  
  testBox.classList.add('show');  
  nextQuestion();  
})
```

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="/css/style.css">  
  </head>  
  
  <body>  
  
    <div class="test-home-box custom-box">  
      <div class="test-app-name">  
        <h2>Online Test</h2>  
      </div>  
      <h4>You'll have 15 seconds to answer each question.</h4>  
      <button class="start-test-btn btn" type="button">Start The Test</button>  
    </div>
```

```
<div class="test-box custom-box">
  <div class="test-app-name">
    <h2>Online Test</h2>
  </div>
  <div class="stats">
    <div class="test-time">
      <div class="remaining-time"></div>
      <span class="time-up-text">Time's up</span>
    </div>
    <div class="score-board">
      <span class="score-text">Score: </span>
      <span class="correct-answers"></span>
    </div>
  </div>
  <div class="question-box">
    <div class="current-question-num">

    </div>
    <div class="question-text">

    </div>
  </div>
  <div class="option-box">

  </div>

  <div class="next-question">
    <button type="button" class="next-question-btn btn">Next Question</button>
    <button type="button" class="see-result-btn btn">See Your Result</button>
  </div>
</div>

<div class="test-over-box custom-box">
  <h1>Test Results</h1>
  <h4>Total Question: <span class="total-questions"></span></h4>
```

```
<h4>Attempt: <span class="total-attempt"></span></h4>
<h4>Correct: <span class="total-correct"></span></h4>
<h4>Wrong: <span class="total-wrong"></span></h4>
<h4>Percentage: <span class="percentage"></span></h4>
<button type="button" class="start-again-test-btn btn">End The Test</button>
<button type="button" class="go-to-home-btn btn">Go To Home</button>
</div>

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

</html>
```

Testdata.json:

```
[{
  "question": "Which of the following is NOT a fruit?",
  "options": ["Rhubarb", "Tomatoes", "Avocados", "Potatos"],
  "answer": 0
},
{
  "question": "Where was the first example of paper money used?",
  "options": ["Turkey", "China", "Greece", "India"],
  "answer": 1
},
{
```

```
"question": "Our human body has?",
"options": ["206 bones", "210 bones", "306 bones", "706 bones"],
"answer": 0
},
{
  "question": "Which of the following languages has the longest alphabet?",
  "options": ["Greek", "Arabic", "Russian", "Chinese"],
  "answer": 2
},
{
  "question": "Which horoscope sign is a fish?",
  "options": ["Aquarius", "Libra", "Cancer", "Pisces"],
  "answer": 3
}
]
```