**Project Objective:**

As a Full Stack Developer, you have to develop an online test portal for assessment.

**Background of the problem statement:**

As the project is progressing, the management has asked you to create an online test application in order to help the recruitment team take online tests of the candidates coming for interviews.

Have used almost all the concepts which were taught in the phase 02 session.

Here is the github repository link

<https://github.com/muskanrafeeq04/phase1-> This link is continued with the phase 01 Github Link the folder name is AngularQuizApp

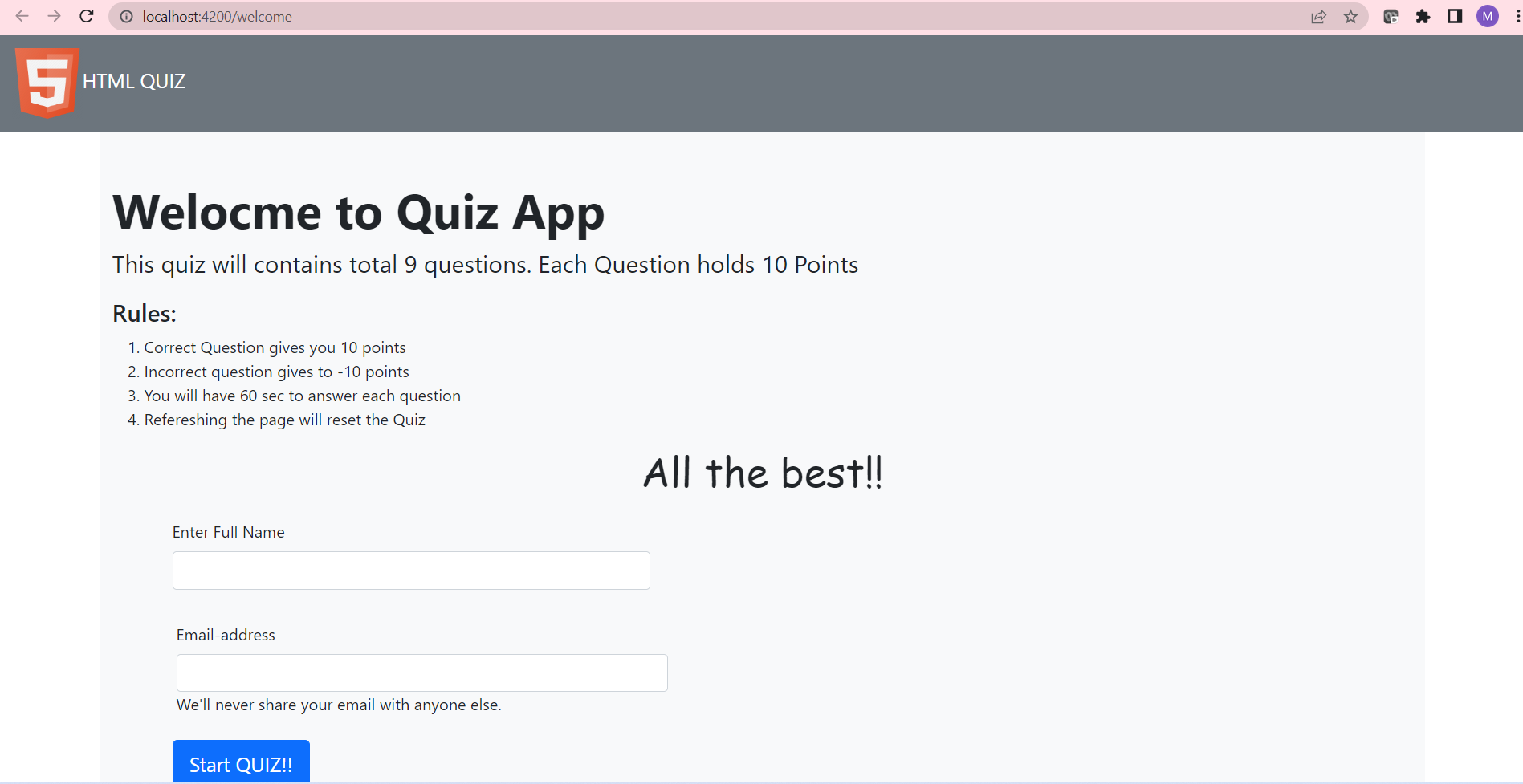
all the task, source code, images everything has been tracked in the github link.

I have created a company website where we can give online test ,

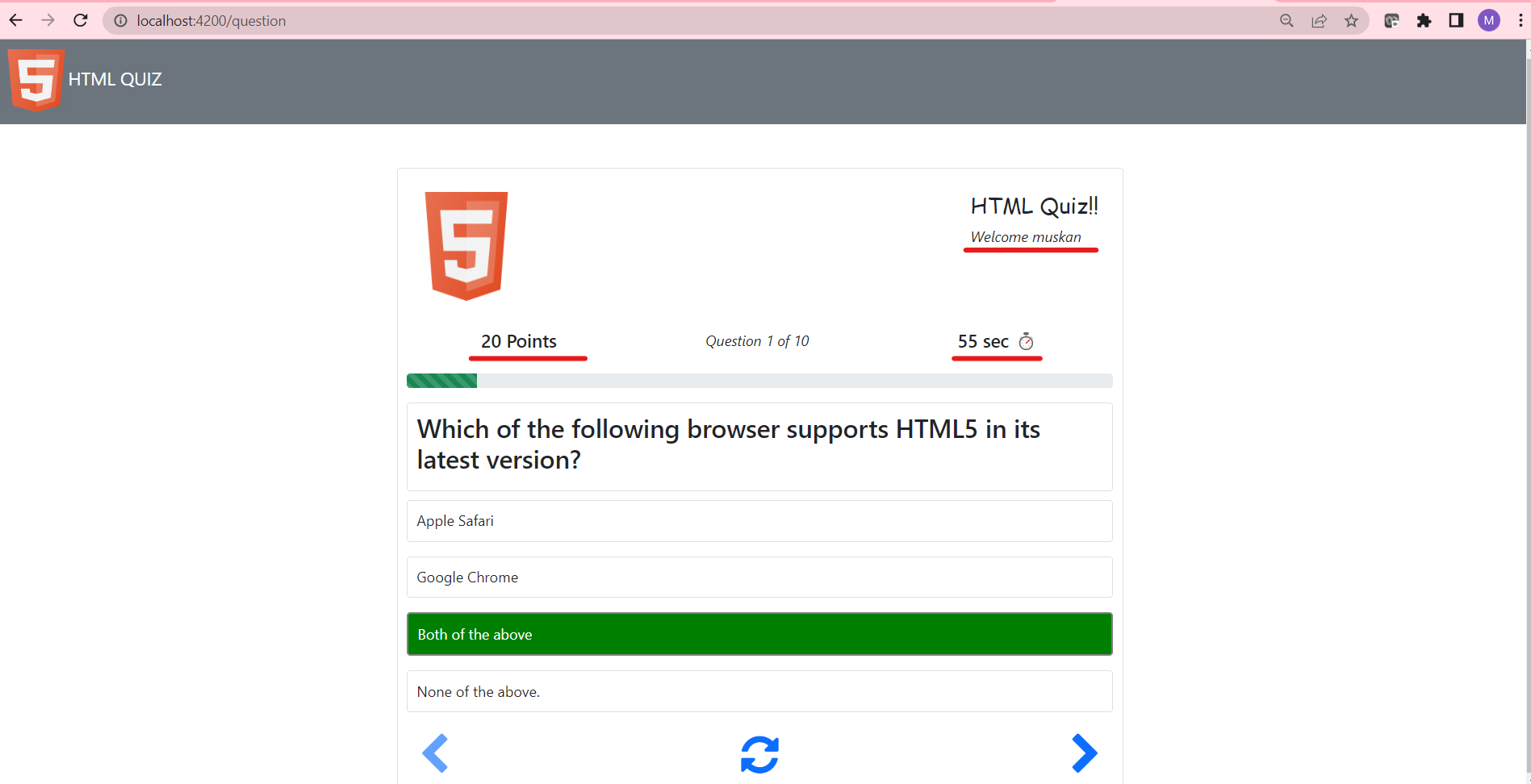
**The website has not been connected with a backend database.**

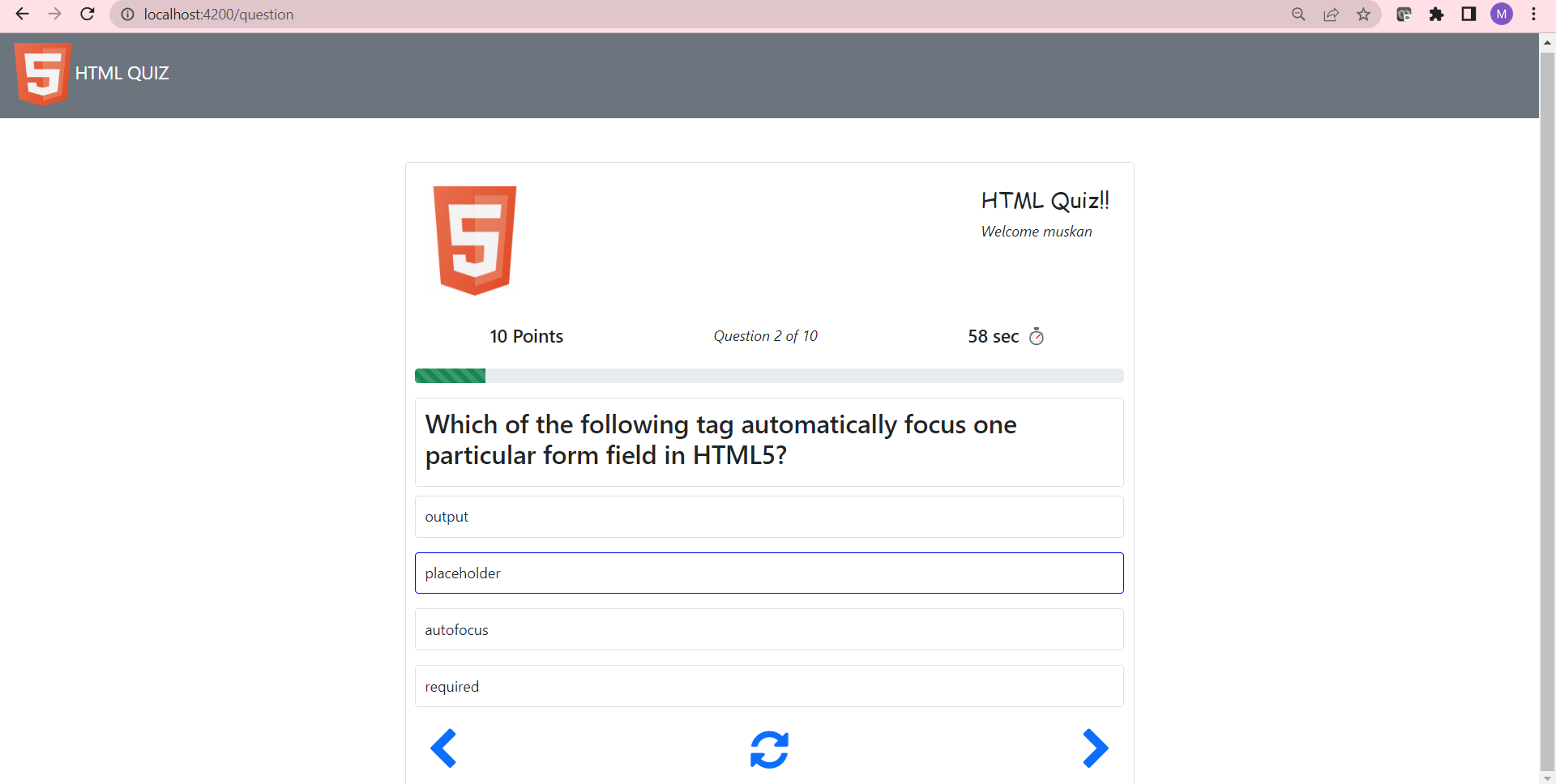
***ScreenShots of the website***

**First Header and welcome page**

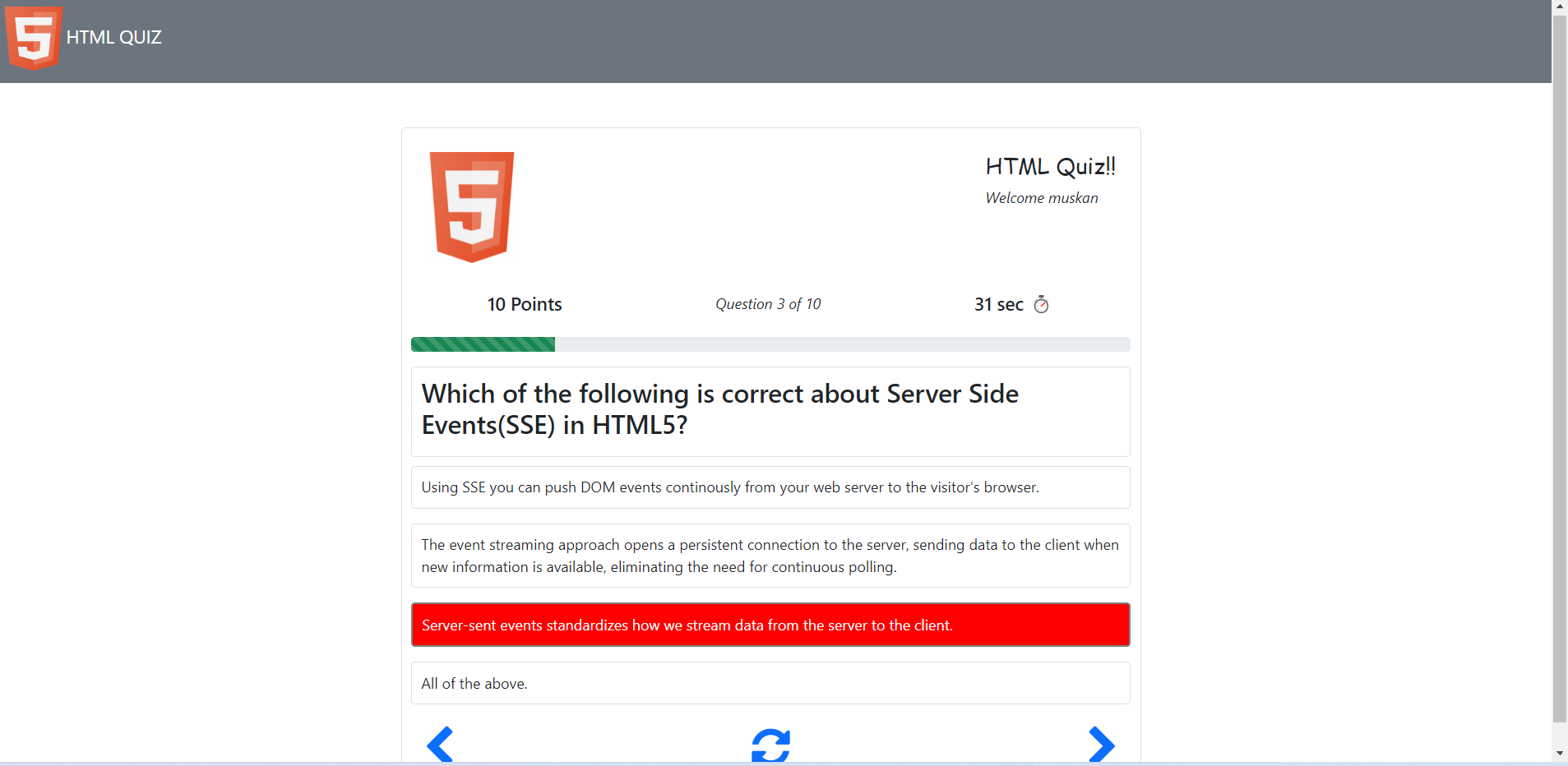
****

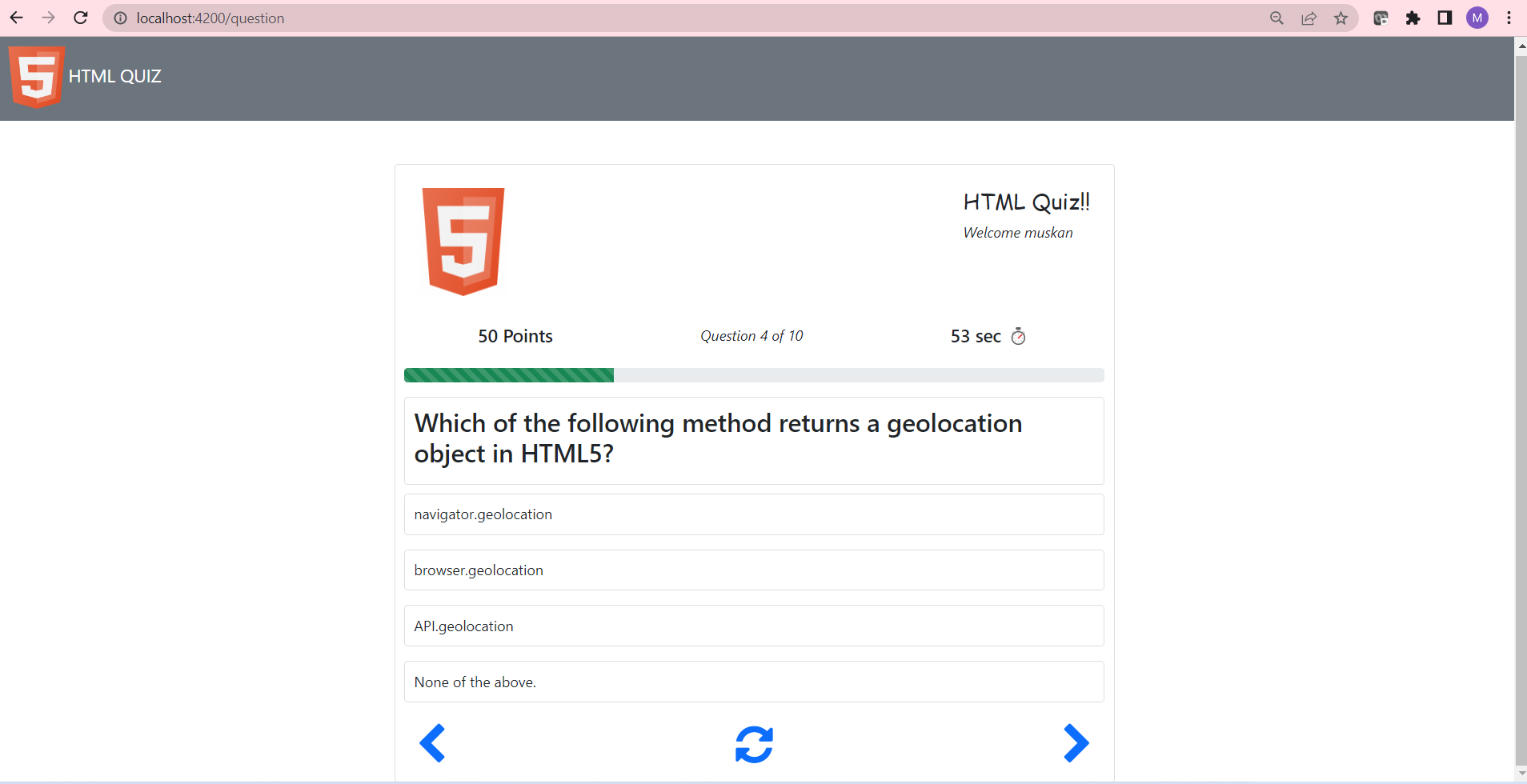
**Questions consist of the timer with add 60 secs to each question, and if the answer is correct it add 10 points to each… similarly it has 10 questions**

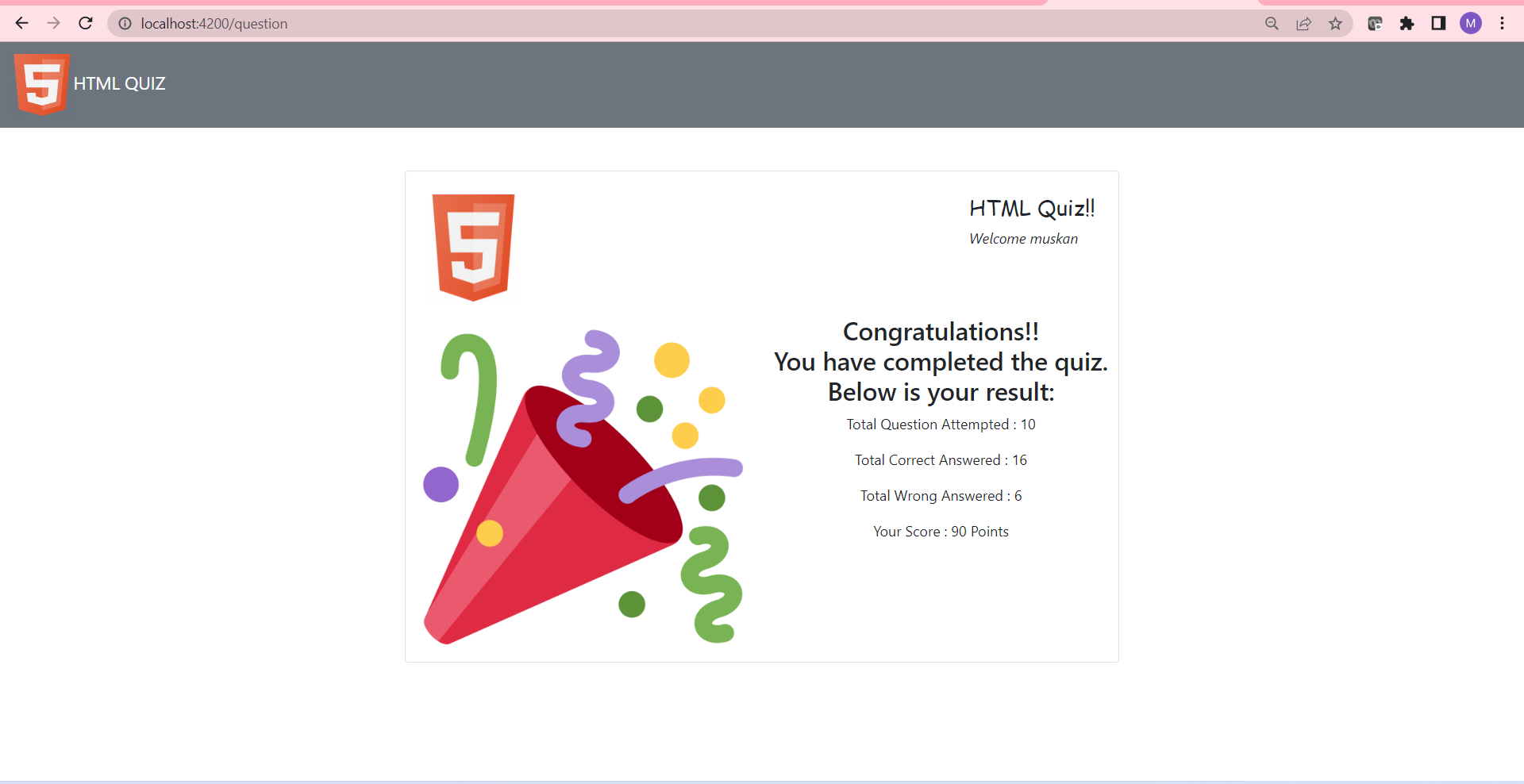
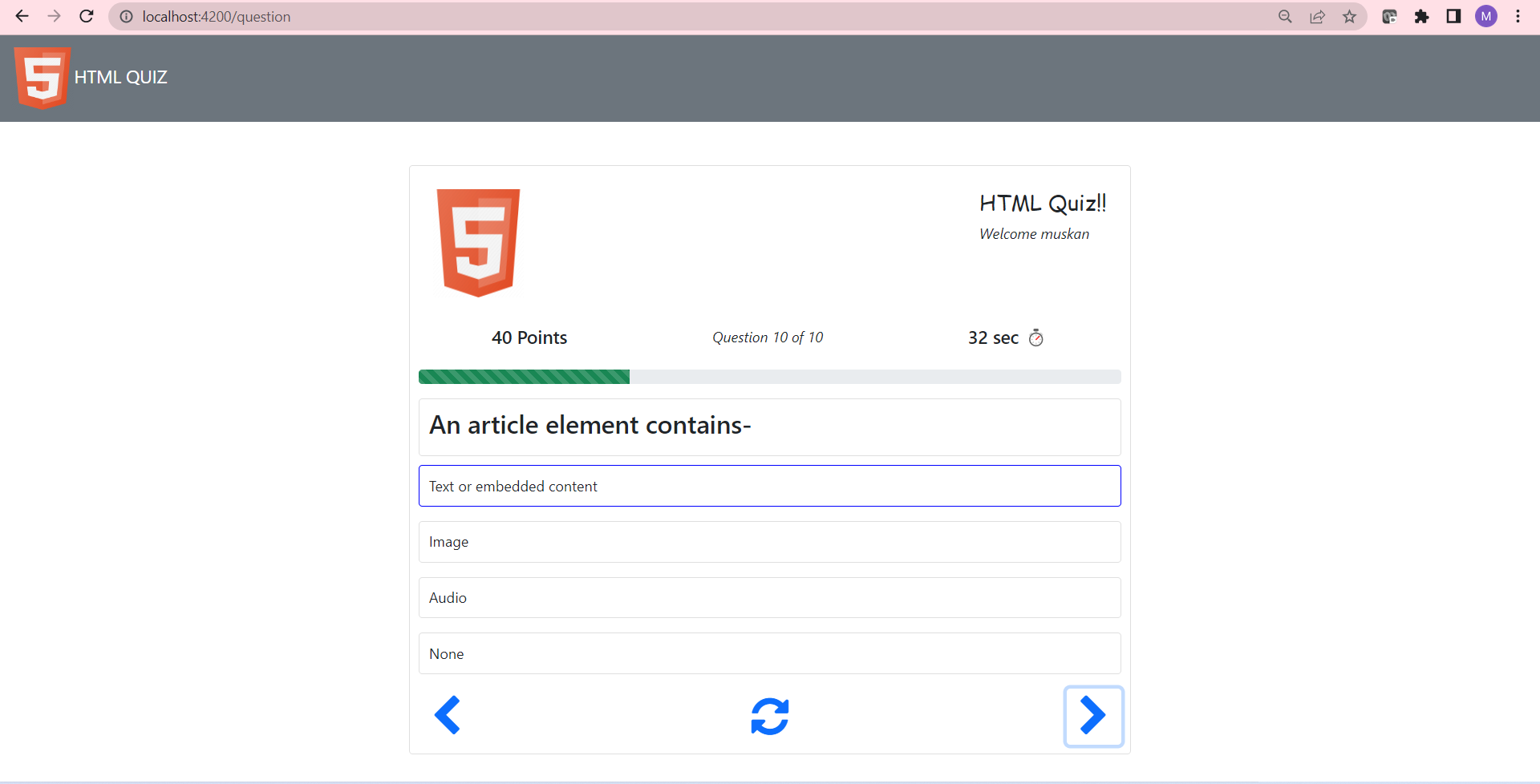
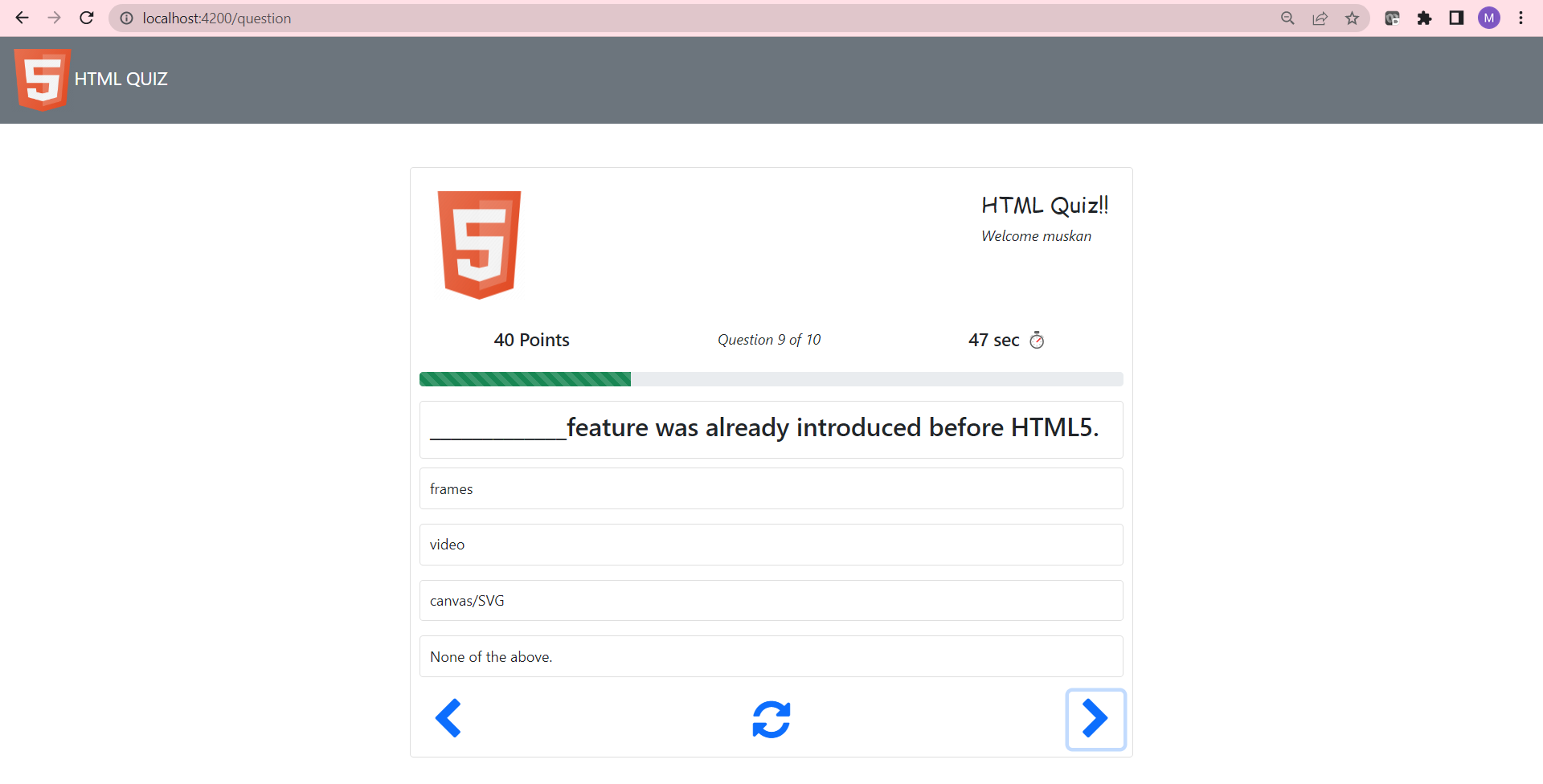
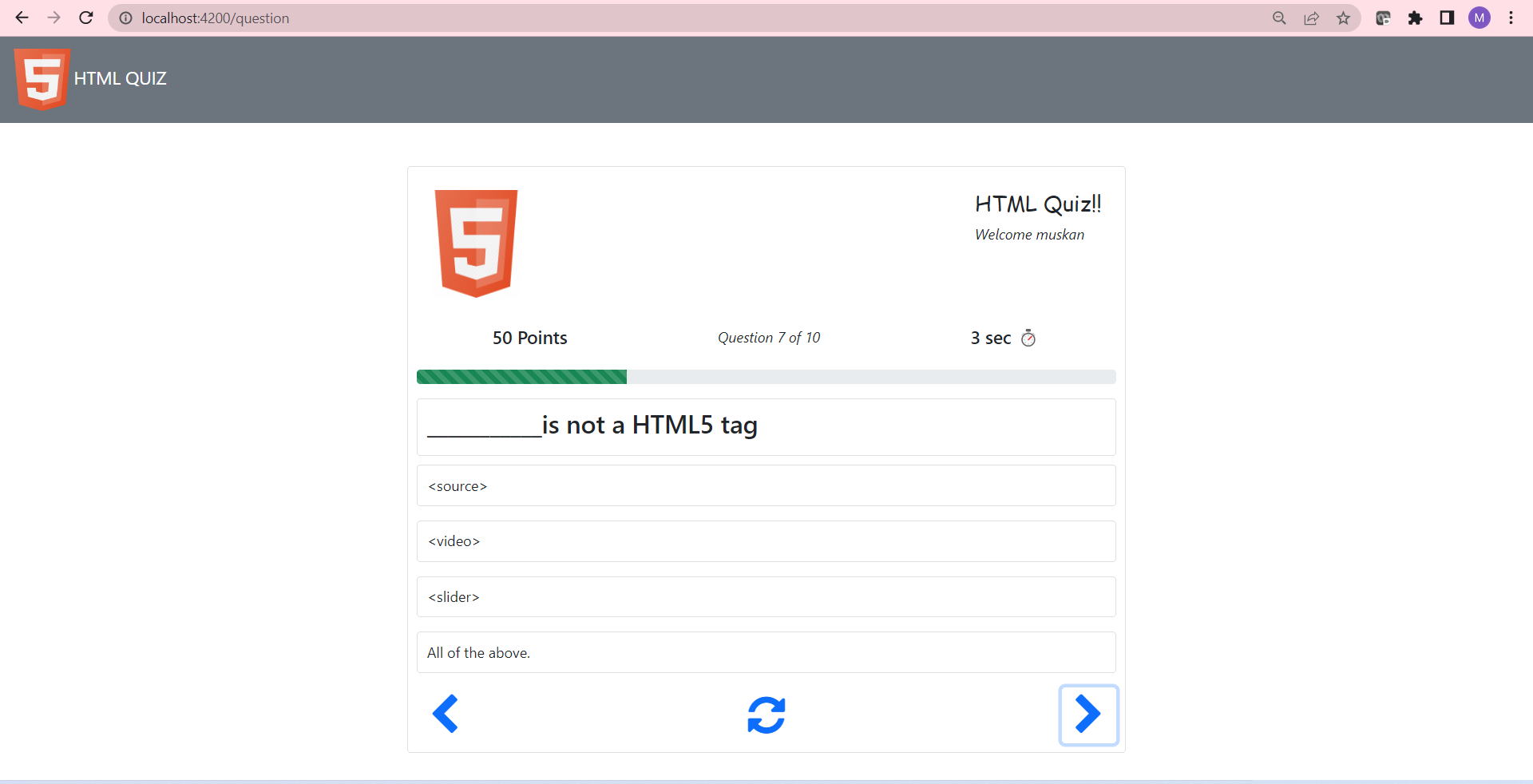
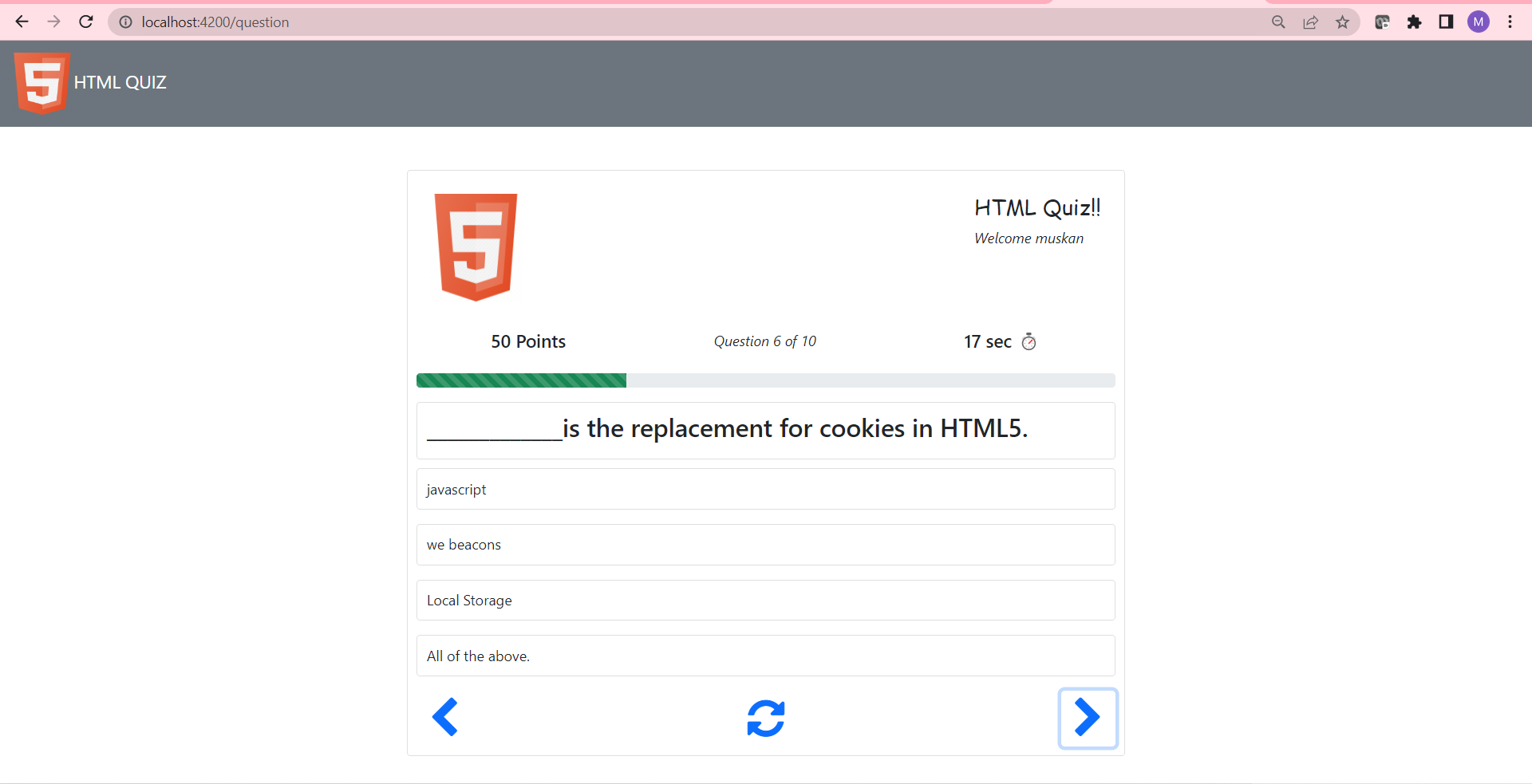
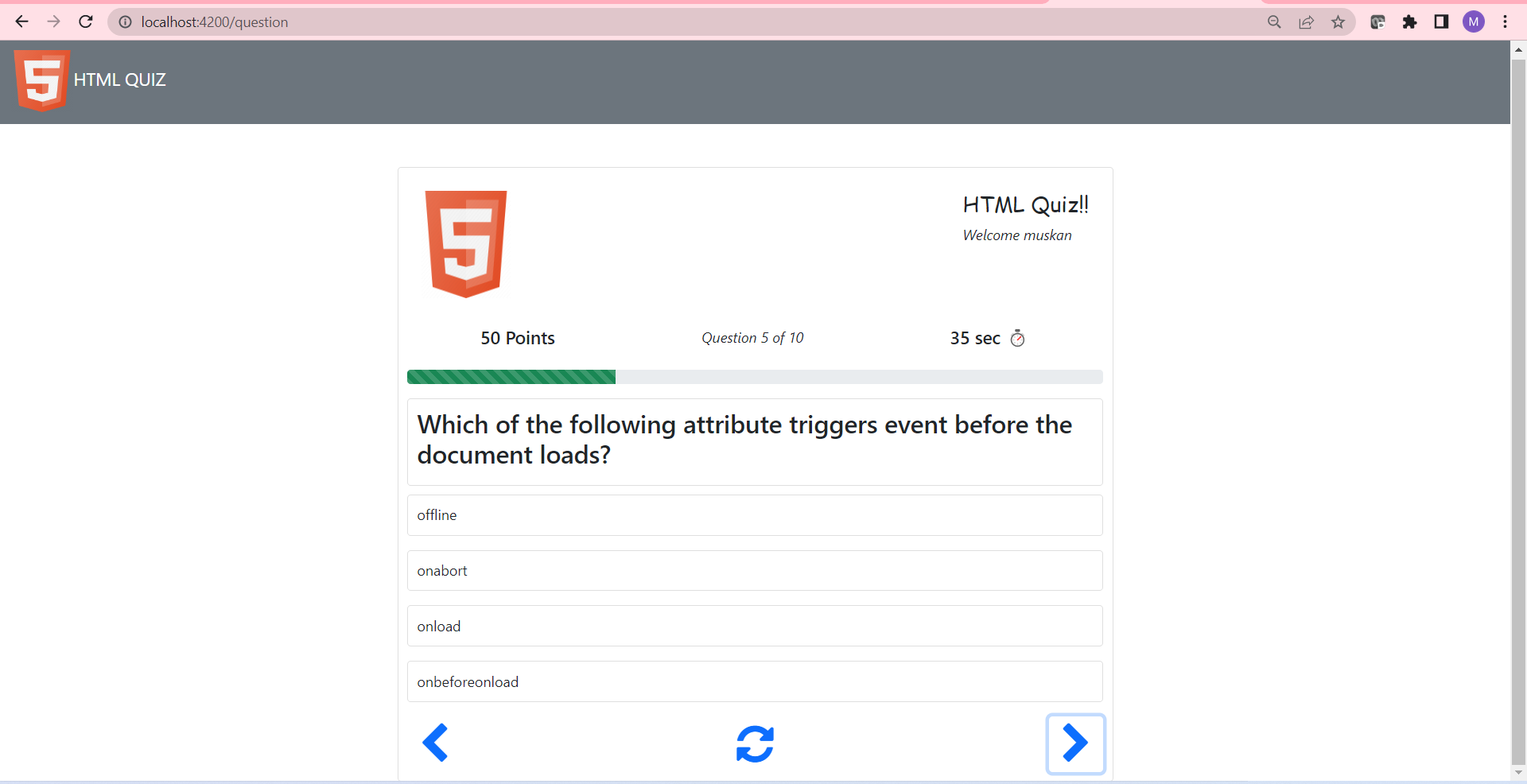
****

****

**Wrong answer gives the negative marking**

****

****

****

**Source Code**

**header.component.html:**

**<nav class="navbar navbar-dark bg-secondary">**

**<div class="container-fluid">**

**<a class="navbar-brand text-bold" href="#">**

**<img src="https://cdn1.iconfinder.com/data/icons/programing-development-7/24/html\_html5\_web\_programing\_developer-512.png" alt="logo" width="70" height="70">**

**<span>HTML QUIZ</span>**

**</a>**

**</div>**

**</nav>**

**header.component.spec.ts:**

**import { ComponentFixture, TestBed } from '@angular/core/testing';**

**import { HeaderComponent } from './header.component';**

**describe('HeaderComponent', () => {**

**let component: HeaderComponent;**

**let fixture: ComponentFixture<HeaderComponent>;**

**beforeEach(async () => {**

**await TestBed.configureTestingModule({**

**declarations: [ HeaderComponent ]**

**})**

**.compileComponents();**

**});**

**beforeEach(() => {**

**fixture = TestBed.createComponent(HeaderComponent);**

**component = fixture.componentInstance;**

**fixture.detectChanges();**

**});**

**it('should create', () => {**

**expect(component).toBeTruthy();**

**});**

**});**

**header.component.ts:**

**import { Component, OnInit } from '@angular/core';**

**@Component({**

**selector: 'app-header',**

**templateUrl: './header.component.html',**

**styleUrls: ['./header.component.scss']**

**})**

**export class HeaderComponent implements OnInit {**

**constructor() { }**

**ngOnInit(): void {**

**}**

**}**

**question.component.html:**

**<div class="container mt-5">**

**<div class="card">**

**<div class="d-flex justify-content-between p-3">**

**<div class="image">**

**<img src="https://cdn1.iconfinder.com/data/icons/programing-development-7/24/html\_html5\_web\_programing\_developer-512.png" width="100" height="120" alt="logo">**

**</div>**

**<div class="quiz-header">**

**<h4 style="font-family: cursive;">HTML Quiz!!</h4>**

**<span style="font-style: italic;">Welcome {{name}}</span>**

**</div>**

**</div>**

**<ng-container \*ngIf="!isQuizCompleted">**

**<div class="d-flex justify-content-around py-3">**

**<div class="score">**

**<h5>{{points}} Points</h5>**

**</div>**

**<div class="question-remain">**

**<span style="font-style: italic;">Question {{currentQuestion+1}} of {{questionList.length}}</span>**

**</div>**

**<div class="timer">**

**<h5>{{counter}} sec ⏱</h5>**

**</div>**

**</div>**

**<div class="progress mb-3">**

**<div class="progress-bar progress-bar-striped bg-success" role="progressbar" [ngStyle]="{'width':progress+'%'}" aria-valuenow="25" aria-valuemin="0" aria-valuemax="100"></div>**

**</div>**

**<div class="question">**

**<div class="card">**

**<h3>{{questionList[currentQuestion]?.questionText}}</h3>**

**</div>**

**</div>**

**<div class="options">**

**<ol \*ngFor="let option of questionList[currentQuestion]?.options">**

**<li (click)="answer(currentQuestion+1,option)">**

**<div appChangeBg [isCorrect]="option.correct" class="card">**

**{{option.text}}**

**</div>**

**</li>**

**</ol>**

**</div>**

**<div class="d-flex justify-content-between">**

**<button [disabled]="currentQuestion===0" class="btn" (click)="previousQuestion()"><i class="fa text-primary fa-chevron-left fa-3x" aria-hidden="true"></i></button>**

**<button class="btn" (click)="resetQuiz()"><i class="fa fa-refresh text-primary fa-3x" aria-hidden="true"></i></button>**

**<button class="btn" (click)="nextQuestion()"><i class="fa text-primary fa-chevron-right fa-3x" aria-hidden="true"></i></button>**

**</div>**

**</ng-container>**

**<ng-container \*ngIf="isQuizCompleted">**

**<div class="row d-flex justify-content-between">**

**<img style="width: 50%;" class="img-fluid col-sm-12 mx-auto" src="https://icon-library.com/images/celebration-icon-png/celebration-icon-png-7.jpg" alt="">**

**<div class="result text-center col-md-6 col-sm-12">**

**<h3>Congratulations!! <br>You have completed the quiz. <br>Below is your result:</h3>**

**<p>Total Question Attempted : {{questionList.length}} </p>**

**<p>Total Correct Answered : {{correctAnswer}} </p>**

**<p>Total Wrong Answered : {{inCorrectAnswer}} </p>**

**<p>Your Score : {{points}} Points </p>**

**</div>**

**</div>**

**</ng-container>**

**</div>**

**</div>**

**question.component.scss:**

**.card {**

**max-width: 800px;**

**margin: 0 auto;**

**padding: 10px;**

**}**

**li {**

**list-style-type: none;**

**cursor: pointer;**

**margin: 10px 0;**

**}**

**li .card:hover {**

**border: 1px solid blue;**

**}**

**ol {**

**padding: 0;**

**}**

**question.component.spec.ts:**

**import { ComponentFixture, TestBed } from '@angular/core/testing';**

**import { QuestionComponent } from './question.component';**

**describe('QuestionComponent', () => {**

**let component: QuestionComponent;**

**let fixture: ComponentFixture<QuestionComponent>;**

**beforeEach(async () => {**

**await TestBed.configureTestingModule({**

**declarations: [ QuestionComponent ]**

**})**

**.compileComponents();**

**});**

**beforeEach(() => {**

**fixture = TestBed.createComponent(QuestionComponent);**

**component = fixture.componentInstance;**

**fixture.detectChanges();**

**});**

**it('should create', () => {**

**expect(component).toBeTruthy();**

**});**

**});**

**question.component.ts:**

**import { Component, OnInit } from '@angular/core';**

**import { interval } from 'rxjs';**

**import { QuestionService } from '../service/question.service';**

**@Component({**

**selector: 'app-question',**

**templateUrl: './question.component.html',**

**styleUrls: ['./question.component.scss']**

**})**

**export class QuestionComponent implements OnInit {**

**public name: string = "";**

**public questionList: any = [];**

**public currentQuestion: number = 0;**

**public points: number = 0;**

**counter = 60;**

**correctAnswer: number = 0;**

**inCorrectAnswer: number = 0;**

**interval$: any;**

**progress: string = "0";**

**isQuizCompleted : boolean = false;**

**constructor(private questionService: QuestionService) { }**

**ngOnInit(): void {**

**this.name = localStorage.getItem("name")!;**

**this.getAllQuestions();**

**this.startCounter();**

**}**

**getAllQuestions() {**

**this.questionService.getQuestionJson()**

**.subscribe(res => {**

**this.questionList = res.questions;**

**})**

**}**

**nextQuestion() {**

**this.currentQuestion++;**

**}**

**previousQuestion() {**

**this.currentQuestion--;**

**}**

**answer(currentQno: number, option: any) {**

**if(currentQno === this.questionList.length){**

**this.isQuizCompleted = true;**

**this.stopCounter();**

**}**

**if (option.correct) {**

**this.points += 10;**

**this.correctAnswer++;**

**setTimeout(() => {**

**this.currentQuestion++;**

**this.resetCounter();**

**this.getProgressPercent();**

**}, 1000);**

**} else {**

**setTimeout(() => {**

**this.currentQuestion++;**

**this.inCorrectAnswer++;**

**this.resetCounter();**

**this.getProgressPercent();**

**}, 1000);**

**this.points -= 10;**

**}**

**}**

**startCounter() {**

**this.interval$ = interval(1000)**

**.subscribe(val => {**

**this.counter--;**

**if (this.counter === 0) {**

**this.currentQuestion++;**

**this.counter = 60;**

**this.points -= 10;**

**}**

**});**

**setTimeout(() => {**

**this.interval$.unsubscribe();**

**}, 600000);**

**}**

**stopCounter() {**

**this.interval$.unsubscribe();**

**this.counter = 0;**

**}**

**resetCounter() {**

**this.stopCounter();**

**this.counter = 60;**

**this.startCounter();**

**}**

**resetQuiz() {**

**this.resetCounter();**

**this.getAllQuestions();**

**this.points = 0;**

**this.counter = 60;**

**this.currentQuestion = 0;**

**this.progress = "0";**

**}**

**getProgressPercent() {**

**this.progress = ((this.currentQuestion / this.questionList.length) \* 100).toString();**

**return this.progress;**

**}**

**}**

**question.service.spec.ts:**

**import { TestBed } from '@angular/core/testing';**

**import { QuestionService } from './question.service';**

**describe('QuestionService', () => {**

**let service: QuestionService;**

**beforeEach(() => {**

**TestBed.configureTestingModule({});**

**service = TestBed.inject(QuestionService);**

**});**

**it('should be created', () => {**

**expect(service).toBeTruthy();**

**});**

**});**

**Question.service.ts:**

**import { Injectable } from '@angular/core';**

**import {HttpClient} from '@angular/common/http'**

**@Injectable({**

**providedIn: 'root'**

**})**

**export class QuestionService {**

**constructor(private http : HttpClient) { }**

**getQuestionJson(){**

**return this.http.get<any>("assets/questions.json");**

**}**

**}**

**welcome.component.html:**

**<div class="container bg-light py-5">**

**<h1 class="display-5 fw-bold">Welcome to Quiz App</h1>**

**<p class="col-md-8 fs-4">This quiz will contain a total of 10 questions. Each Question holds 10 Points</p>**

**<h4>Rules:</h4>**

**<ol>**

**<li>Correct Question gives you 10 points</li>**

**<li>Incorrect question gives to -10 points</li>**

**<li>You will have 60 sec to answer each question</li>**

**<li>Refreshing the page will reset the Quiz</li>**

**</ol>**

**<h1 style="font-family: cursive;text-align:center">All the best!!</h1>**

**<form class="row g-8 mx-5">**

**<div class="name col-md-5 my-3">**

**<label for="" class="form-label">Enter Full Name</label>**

**<input #name type="text" class="form-control" required>**

**</div>**

**<br>**

**<div class="mb-2 mx-1">**

**<div class="name col-md-5 my-3">**

**<label for="" class="form-label">Email-address</label>**

**<input type="email" class="form-control" required>**

**We'll never share your email with anyone else.**

**</div>**

**</div>**

**<div class="col-10">**

**<button class="btn btn-primary btn-lg" type="submit" routerLink="/question" (click)="startQuiz()">Start QUIZ!!</button>**

**</div>**

**<!-- <div class="mb-3">**

**<button class="btn btn-primary" type="submit" disabled routerLink="/question" (click)="startQuiz()">Submit form</button>**

**</div> -->**

**</form>**

**Welcome.component.spec.ts:**

**import { ComponentFixture, TestBed } from '@angular/core/testing';**

**import { WelcomeComponent } from './welcome.component';**

**describe('WelcomeComponent', () => {**

**let component: WelcomeComponent;**

**let fixture: ComponentFixture<WelcomeComponent>;**

**beforeEach(async () => {**

**await TestBed.configureTestingModule({**

**declarations: [ WelcomeComponent ]**

**})**

**.compileComponents();**

**});**

**beforeEach(() => {**

**fixture = TestBed.createComponent(WelcomeComponent);**

**component = fixture.componentInstance;**

**fixture.detectChanges();**

**});**

**it('should create', () => {**

**expect(component).toBeTruthy();**

**});**

**});**

**Welcome.component.ts:**

**import { Component, OnInit,ViewChild,ElementRef } from '@angular/core';**

**@Component({**

**selector: 'app-welcome',**

**templateUrl: './welcome.component.html',**

**styleUrls: ['./welcome.component.scss']**

**})**

**export class WelcomeComponent implements OnInit {**

**@ViewChild('name') nameKey!: ElementRef;**

**constructor() { }**

**ngOnInit(): void {**

**}**

**startQuiz(){**

**localStorage.setItem("name",this.nameKey.nativeElement.value);**

**}**

**}**

**App-routing.module.ts:**

import { NgModule } from '@angular/core';

import { RouterModule, Routes } from '@angular/router';

import { QuestionComponent } from './question/question.component';

import { WelcomeComponent } from './welcome/welcome.component';

const routes: Routes = [

{path:'', redirectTo:'welcome',pathMatch:"full"},

{path:"welcome", component:WelcomeComponent},

{path:"question", component:QuestionComponent

];

@NgModule({

imports: [RouterModule.forRoot(routes)],

exports: [RouterModule]

})

export class AppRoutingModule { }

**App.component.spec.ts:**

**import { TestBed } from '@angular/core/testing';**

**import { RouterTestingModule } from '@angular/router/testing';**

**import { AppComponent } from './app.component';**

**describe('AppComponent', () => {**

**beforeEach(async () => {**

**await TestBed.configureTestingModule({**

**imports: [**

**RouterTestingModule**

**],**

**declarations: [**

**AppComponent**

**],**

**}).compileComponents();**

**});**

**it('should create the app', () => {**

**const fixture = TestBed.createComponent(AppComponent);**

**const app = fixture.componentInstance;**

**expect(app).toBeTruthy();**

**});**

**it(`should have as title 'AngularQuizApp'`, () => {**

**const fixture = TestBed.createComponent(AppComponent);**

**const app = fixture.componentInstance;**

**expect(app.title).toEqual('AngularQuizApp');**

**});**

**it('should render title', () => {**

**const fixture = TestBed.createComponent(AppComponent);**

**fixture.detectChanges();**

**const compiled = fixture.nativeElement as HTMLElement;**

**expect(compiled.querySelector('.content span')?.textContent).toContain('AngularQuizApp app is running!');**

**});**

**});**

**App.component.ts:**

**import { Component } from '@angular/core';**

**@Component({**

**selector: 'app-root',**

**templateUrl: './app.component.html',**

**styleUrls: ['./app.component.scss']**

**})**

**export class AppComponent {**

**title = 'AngularQuizApp';**

**}**

**App.module.ts:**

**import { NgModule } from '@angular/core';**

**import { BrowserModule } from '@angular/platform-browser';**

**import { AppRoutingModule } from './app-routing.module';**

**import { AppComponent } from './app.component';**

**import { WelcomeComponent } from './welcome/welcome.component';**

**import { QuestionComponent } from './question/question.component';**

**import { HeaderComponent } from './header/header.component';**

**import { HttpClientModule } from '@angular/common/http';**

**import { ChangeBgDirective } from './change-bg.directive';**

**@NgModule({**

**declarations: [**

**AppComponent,**

**WelcomeComponent,**

**QuestionComponent,**

**HeaderComponent,**

**ChangeBgDirective**

**],**

**imports: [**

**BrowserModule,**

**AppRoutingModule,**

**HttpClientModule**

**],**

**providers: [],**

**bootstrap: [AppComponent]**

**})**

**export class AppModule { }**

**Questions.json:**

**{**

**"questions": [{**

**"questionText": " Which of the following browser supports HTML5 in its latest version?",**

**"options": [{**

**"text": " Apple Safari"**

**},**

**{**

**"text": "Google Chrome"**

**},**

**{**

**"text": "Both of the above",**

**"correct": true**

**},**

**{**

**"text": "None of the above."**

**}**

**],**

**"explanation": "Both of the above browsers supports HTML5."**

**},**

**{**

**"questionText": "Which of the following tag automatically focus one particular form field in HTML5?",**

**"options": [{**

**"text": "output"**

**},**

**{**

**"text": "placeholder"**

**},**

**{**

**"text": "autofocus",**

**"correct": true**

**},**

**{**

**"text": "required"**

**}**

**],**

**"explanation": "This is a simple one-step pattern, easily programmed in JavaScript at the time of document load, automatically focus one particular form field."**

**},**

**{**

**"questionText": "Which of the following is correct about Server Side Events(SSE) in HTML5?",**

**"options": [{**

**"text": "Using SSE you can push DOM events continously from your web server to the visitor's browser."**

**},**

**{**

**"text": "The event streaming approach opens a persistent connection to the server, sending data to the client when new information is available, eliminating the need for continuous polling."**

**},**

**{**

**"text": " Server-sent events standardizes how we stream data from the server to the client."**

**},**

**{**

**"text": "All of the above.",**

**"correct": true**

**}**

**],**

**"explanation": "All of the above options are correct."**

**},**

**{**

**"questionText": "Which of the following method returns a geolocation object in HTML5?",**

**"options": [{**

**"text": "navigator.geolocation",**

**"correct": true**

**},**

**{**

**"text": "browser.geolocation"**

**},**

**{**

**"text": "API.geolocation"**

**},**

**{**

**"text": " None of the above."**

**}**

**],**

**"explanation": "navigator.geolocation returns geolocation object."**

**},**

**{**

**"questionText": "Which of the following attribute triggers event before the document loads?",**

**"options": [{**

**"text": "offline"**

**},**

**{**

**"text": "onabort"**

**},**

**{**

**"text": "onload"**

**},**

**{**

**"text": "onbeforeonload",**

**"correct": true**

**}**

**],**

**"explanation": "onbeforeonload − Triggers event before the document loads."**

**},**

**{**

**"questionText": "\_\_\_\_\_\_\_\_\_\_\_\_\_is the replacement for cookies in HTML5.",**

**"options": [{**

**"text": "javascript"**

**},**

**{**

**"text": "we beacons"**

**},**

**{**

**"text": "Local Storage",**

**"correct": true**

**},**

**{**

**"text": "All of the above."**

**}**

**],**

**"explanation": "Local Storage is the replacement for cookies in HTML5."**

**},**

**{**

**"questionText": "\_\_\_\_\_\_\_\_\_\_\_is not a HTML5 tag",**

**"options": [{**

**"text": "<source>"**

**},**

**{**

**"text": "<video>"**

**},**

**{**

**"text": "<slider>",**

**"correct": true**

**},**

**{**

**"text": "All of the above."**

**}**

**],**

**"explanation": "<slider> is not a HTML5 tag ."**

**},**

**{**

**"questionText": "Which of the following element may contain not just links but also other interactive items, including the newly introduced command element.",**

**"options": [{**

**"text": "meter"**

**},**

**{**

**"text": "menu",**

**"correct": true**

**},**

**{**

**"text": "details"**

**},**

**{**

**"text": "progress"**

**}**

**],**

**"explanation": "menu element may contain not just links but also other interactive items, including the newly introduced command element."**

**},**

**{**

**"questionText": "\_\_\_\_\_\_\_\_\_\_\_\_\_feature was already introduced before HTML5.",**

**"options": [{**

**"text": "frames",**

**"correct": true**

**},**

**{**

**"text": "video"**

**},**

**{**

**"text": "canvas/SVG"**

**},**

**{**

**"text": "None of the above."**

**}**

**],**

**"explanation": "Frames feature was already introduced before HTML5."**

**},**

**{**

**"questionText": "An article element contains-",**

**"options": [{**

**"text": "Text or embedded content",**

**"correct": true**

**},**

**{**

**"text": "Image"**

**},**

**{**

**"text": "Audio"**

**},**

**{**

**"text": "None"**

**}**

**],**

**"explanation": "An article element contains Text or embedded content."**

**}**

**]**

**}**

**index.html:**

**<!doctype html>**

**<html lang="en">**

**<head>**

**<meta charset="utf-8">**

**<title>AngularQuizApp</title>**

**<base href="/">**

**<meta name="viewport" content="width=device-width, initial-scale=1">**

**<link rel="icon" type="image/x-icon" href="favicon.ico">**

**<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">**

**<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">**

**</head>**

**<body>**

**<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p" crossorigin="anonymous"></script>**

**<app-root></app-root>**

**</body>**

**</html>**

****