Header Component.html

<nav class="navbar navbar-dark bg-light  bg-image"

style="background-image: url('assets/img/images xs.jpg');background-repeat:no-repeat;background-size:1490px 100px ;">

    <div class="container-fluid">

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

        <img src="assets/img/Qui2.jpg" width="50" >

      </a>

    </div>

</nav>

Header component.ts

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

@Component({

  selector: 'app-header',

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

  styleUrls: ['./header.component.css']

})

export class HeaderComponent implements OnInit {

  constructor() { }

  ngOnInit(): void {

  }

}

Welcome component Html

<div class="container  bg-dark text-light my-1 py-5 bg-image" style="background-image: url('assets/img/backround.jpg'); height: 100%;opacity:5 ;">

 <h1 class="display-4 fw-bold ">Welcome to  quiz app </h1>

 <p class="col-md-8 fs-4 ">Quiz contain 10 question & each hold 10 points.</p>

 <h4>Rules for this quiz:</h4>

 <ol >

    <li>Correct answer will incur 10 point for each. </li>

    <li>Similarly  incorrect answer will incur -10 point for each. </li>

    <li>Receive 60 second for each question. </li>

    <li>Refreshing will reset the quiz(so refrain refresh while doing quiz).</li>

 </ol>

 <h3 style="text-align :center ">Best of luck to score better!!</h3>

 <div class="name col-md-4 my-4 ">

    <label for="name"> Enter your name</label>

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

 </div>

 <button class="btn btn-outline-light" routerLink="/questions" (click)="startQuiz()">Click to start the quiz</button>

</div>

Welcome component .ts

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

@Component({

  selector: 'app-welcome',

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

  styleUrls: ['./welcome.component.css']

})

export class WelcomeComponent implements OnInit {

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

  constructor() { }

  ngOnInit(): void {

  }

 startQuiz(){

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

 }

}

Question component .htm’

<div class="container shadow-lg mt-3 mb-3 py-3" style="background: rgb(14,113,107);

background: linear-gradient(90deg, rgba(14,113,107,0.8155637254901961) 74%, rgba(113,62,14,0.10688025210084029) 99%, rgba(2,0,36,1) 99%, rgba(0,212,255,1) 100%);">

<div class="card shadow p-3" style="background: rgb(2,0,36);

background: radial-gradient(circle, rgba(2,0,36,1) 0%, rgba(14,16,113,0.10688025210084029) 99%, rgba(0,212,255,1) 100%);">

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

<div class="image">

    <img src="assets/img/Qui2.jpg" width ="100" alt="logo">

</div>

<div class="quiz-header text-light" >

<h4 style="font-family: cursive;"> Quiz on HTML Related question</h4>

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

</div>

</div>

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

    <div class="d-flex justify-content-around py-1 text-light " >

        <div class="score"><h5>{{points}} Points</h5></div>

        <div class="questio-remaing"><span> Question {{currentQuestion+1}} of {{questionslist.length}}</span></div>

        <div class="time">

            <h6>{{counter}} sec <span><img src="assets/img/images.png" width="20" ></span></h6>

        </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">

        <h2>{{questionslist[currentQuestion]?.questionText}}</h2>

        </div>

      </div>

    <div class="options">

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

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

            <div appChnageBg [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-sharp  fa-solid fa-circle-chevron-left fa-2x text-dark"></i>

        </button>

        <button class="btn" (click)="resetQuiz()">

            <i class="fa-sharp  fa-solid fa-arrows-rotate fa-2x"></i>

        </button>

        <button class="btn " (click)="nextQuestion()">

            <i class="fa-sharp  fa-solid fa-circle-chevron-right fa-2x"></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="assets/img/keepmoving bes.jpg"[alt]="">

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

        <h3>You have completed your quiz <br> Below is quiz attempted result:</h3>

        <p>No. of Question attempted :    {{questionslist.length}}</p>

        <p>No. of Question attempted correct : {{correctAnswer}}</p>

        <p>No. of Question attempted incorrect : {{incorrectAnswer}}</p>

        <p>Total Score of Quiz : {{points}} Points</p>

    </div>

   </div>

</ng-container>

</div>

</div>

Question Component.css

.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 rgba(60, 11, 194, 0.753);

}

ol{

    padding: 0;

}

Question Component.ts

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

import { interval } from 'rxjs';

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

@Component({

  selector: 'app-questions',

  templateUrl: './questions.component.html',

  styleUrls: ['./questions.component.css']

})

export class QuestionsComponent implements OnInit {

public name: string="";

public questionslist :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.getAllQuestion();

    this.startCounter();

  }

getAllQuestion(){

this.questionService.getQuestionJson()

.subscribe(res=>{

  this.questionslist=res.questions;

})

}

nextQuestion(){

this.currentQuestion++;

}

previousQuestion()

{

this.currentQuestion--;

}

answer(currentQno:number,option:any){

  if(currentQno === this.questionslist.length){

    this.isQuizCompleted=true;

    this.stopCounter();

  }

  if(option.correct){

    this.points+=10;

    this.correctAnswer++;

   setTimeout(()=>{

    this.currentQuestion++;

    //this.points=this.points+10;if

     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=0;

 this.startCounter();

}

resetQuiz(){

  this.resetCounter();

  this.getAllQuestion();

  this.points=0;

  this.counter=60;

  this.currentQuestion=0;

  this.progress="0";

}

getProgressPercent(){

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

  return this.progress;

}

}

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/questionlist.json")

  }

}

App routing modult

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

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

import { QuestionsComponent } from './questions/questions.component';

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

const routes: Routes = [

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

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

  {path:"questions",component:QuestionsComponent}

];

@NgModule({

  imports: [RouterModule.forRoot(routes)],

  exports: [RouterModule]

})

export class AppRoutingModule { }

app component Html

<app-header></app-header>

<router-outlet></router-outlet>

App component .ts

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

@Component({

  selector: 'app-root',

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

  styleUrls: ['./app.component.css']

})

export class AppComponent {

  title = 'quiz';

  myimage:string="assets/img/backround.jpg";

  myimage2:string="assets/img/images sddss.jpg";

  myimage3:string="assets/img/Ste.jpg";

}

App module ts

@NgModule({

  declarations: [

    AppComponent,

    WelcomeComponent,

    QuestionsComponent,

    HeaderComponent,

    ChnageBgDirective

  ],

  imports: [

    BrowserModule,

    AppRoutingModule,

    HttpClientModule

  ],

  providers: [],

  bootstrap: [AppComponent]

})

export class AppModule { }

change bg directory.ts

import { Directive,ElementRef,HostListener,Input,Renderer2 } from '@angular/core';

@Directive({

  selector: '[appChnageBg]'

})

export class ChnageBgDirective {

  @Input() isCorrect:boolean=false;

  constructor(private el:ElementRef,private render:Renderer2) { }

    @HostListener('click') answer(){

      if(this.isCorrect){

        this.render.setStyle(this.el.nativeElement,'background','green');

        this.render.setStyle(this.el.nativeElement,'color','#fff');

        this.render.setStyle(this.el.nativeElement,'border','#2px solid grey ');

      }else{

        this.render.setStyle(this.el.nativeElement,'background','red');

        this.render.setStyle(this.el.nativeElement,'color','#fff');

        this.render.setStyle(this.el.nativeElement,'border','#2px solid grey ');

      }

    }

}

Inside asset json file was created questionlist.json

{

"questions":[

    {

        "questionText":"The float property of a CSS defines the \_\_\_\_\_\_\_\_.",

        "options":[

            {

                "text":"content"

            },

            {

                "text":"text"

            },

            {

                "text":"flow of content" ,

                "correct": true

            },

            {

                "text":"None of these "

            }

        ],

        "explanation":"The CSS float property specifies how an element should float. The CSS clear property specifies what elements can float beside the cleared element and on which side."

    },

    {

        "questionText":"Symbol used in href for linking to the section in the same document.",

        "options":[

            {

                "text":"#",

                "correct": true

            },

            {

                "text":"@"

            },

            {

                "text":"?"

            },

            {

                "text":"None of these "

            }

        ],

        "explanation":"This is the most used attribute for the <a> tag. This is what is used to create a hyperlink. The value associated with the href attribute should be either a complete or relative address to another web page, a mail link or an anchor name ."

    },

    {

        "questionText":"\_\_\_\_\_\_\_\_\_ uses the id attribute of an HTML element to select a specific element.",

        "options":[

            {

                "text":"class selector"

            },

            {

                "text":" id selector" ,

                "correct": true

            },

            {

                "text":"Both (a) and (b)"

            },

            {

                "text":"None of these "

            }

        ],

        "explanation":"Because an id is always unique within the page, it is used to select a single, distinct element"

    },

    {

        "questionText":" \_\_\_\_\_\_\_\_ is the default color of a hyperlink..",

        "options":[

            {

                "text":"Red"

            },

            {

                "text":"Blue",

                "correct": true

            },

            {

                "text":"Green"

            },

            {

                "text":"Black "

            }

        ],

        "explanation":"An unvisited link is underlined and blue in color normally"

    },

    {

        "questionText":"\_\_\_\_\_\_\_\_\_ property of CSS specifies the horizontal alignment of text in an element.",

        "options":[

            {

                "text":"float"

            },

            {

                "text":"width"

            },

            {

                "text":"text-align" ,

                "correct": true

            },

            {

                "text":"padding"

            }

        ],

        "explanation":"The text-align property in CSS is used to specify the horizontal alignment of text in an element ie., it is used to set the alignment of the content horizontally, inside a block element or table-cell box."

    },

    {

        "questionText":" Which of the following linking allows a link to another section on the same or different web pages? ",

        "options":[

            {

                "text":"Internal",

                "correct": true

            },

            {

                "text":"External"

            },

            {

                "text":"Inline"

            },

            {

                "text":"Outline "

            }

        ],

        "explanation":"Internal links are hyperlinks that point at (target) the same domain as the domain that the link exists on (source). In layman's terms, an internal link is one that points to another page on the same website."

    },

    {

        "questionText":" For internal linking, section names are provided by \_\_\_\_\_\_ attribute of tag.",

        "options":[

            {

                "text":"title"

            },

            {

                "text":"href"

            },

            {

                "text":" name" ,

                "correct": true

            },

            {

                "text":"None of these "

            }

        ],

        "explanation":"The name attribute is used to reference elements in a JavaScript, or to reference form data after a form is submitted."

    },

    {

        "questionText":"Which of the following is used to send E-mails through a website?",

        "options":[

            {

                "text":"tomail"

            },

            {

                "text":"mailto",

                "correct": true

            },

            {

                "text":"Both (a) and (b)"

            },

            {

                "text":"None of these "

            }

        ],

        "explanation":"HTML <a> tag provides you option to specify an email address to send an email. While using <a> tag as an email tag, you will use mailto: email address along with href attribute. Following is the syntax of using mailto instead of using http. This code will generate the following link which you can use to send email."

    },

    {

        "questionText":" Which property of CSS is used to set the font type of an HTML element?",

        "options":[

            {

                "text":"font-size"

            },

            {

                "text":"font-style"

            },

            {

                "text":"font-family" ,

                "correct": true

            },

            {

                "text":"float"

            }

        ],

        "explanation":"To change font type purely with HTML, use the CSS font-family property. Set it to the value you want and place it inside a style attribute."

    },

    {

        "questionText":" A CSS declaration always ends with a \_\_\_.",

        "options":[

            {

                "text":"dot"

            },

            {

                "text":"colon"

            },

            {

                "text":"semicolon" ,

                "correct": true

            },

            {

                "text":"hash "

            }

        ],

        "explanation":"The declaration block contains one or more declarations separated by semicolons. Each declaration includes a CSS property name and a value, separated by a colon. Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces"

    }

]

}