**Online Test Application.**

Course-end Project 1

DESCRIPTION

**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.

**You must use the following:**

* Visual Studio Code
* NodeJS
* Angular
* GitHub

**Following requirements should be met:**

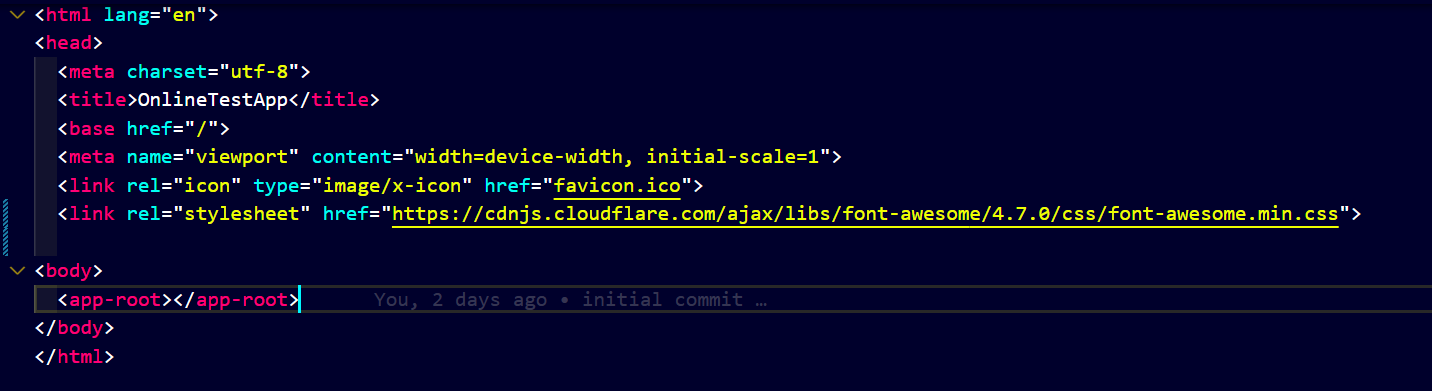
* A few of the source codes should be tracked on GitHub repositories. You need to document the tracked files that are ignored during the final push to the GitHub repository.

# Source Code

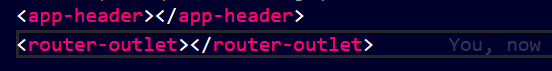
## 1. Create a Project named as

> ng new Online\_test\_app

## 2. Index.html



App.component.html



## 3. Style.css

## 4. Create Component:

### Header.html

Header.html

<nav class="navbar bg-danger ">

    <div class="container-fluid">

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

        <img src="assets/images/Ang1.jpg" alt="Logo" width="50" height="34">

      </a>

    </div>

  </nav>

### Welcome.html

*<!-- <p>welcome works!</p> -->*

*Welcome.html*

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

    <h1 class ="display-4 "> Welcome to <span class="mark">Quiz</span> App</h1>

    <p class="col-md-8 fs-5">This quiz will contains total 9 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>Refereshing the page will reset the Quiz</li>

    </ol>

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

    <div class="name col-md-4 my-3">

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

        <input #name type="text" class="form-control mt-2">

    </div>

         <button class="btn btn-primary btn-lg" routerLink="/question" (click)="startQuiz()" >Start the Quiz ! </button>

     </div>

Welcome.ts

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

@Component({

  selector: 'app-welcome',

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

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

})

export class WelcomeComponent {

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

  constructor(){

  }

  ngOnInit(): void {

  }

  startQuiz(){

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

  }

}

### Questions.html

*<!-- Question.html -->*

<div class="container">

<div class="card border-danger text-danger mt-5">

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

        <div class="image">

            <img src="assets/images/Ang2.jpg"width="200" alt="logo">

        </div>

        <div class="quiz-header">

            <h4 style="font-family: cursive;" class="d-flex justify-content-center">Quiz!</h4>

            <span style="font-style: italic; color:blue"> 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" role="progressbar" [ngStyle]="{'width':progress+'%'}" aria-valuenow="55" aria-valuemin="0" aria-valuemax="100"></div>

      </div>

      <div class="question">

        <div class="card">

            <h4>{{questionList[currentQuestion]?.questionText}}</h4>

        </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> -->*

            <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" type="button" class="btn " (click)="previousQuestion()"> <i class="fa text-primary fa-chevron-left fa-3x" aria-hidden="true"></i></button>

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

        <button type="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.ts

Question.ts

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

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

import { interval } from 'rxjs';

@Component({

  selector: 'app-question',

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

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

})

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.getQuestionsJson()

      .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) *//1sec*

      .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;

  }

}

### Displaying Timer:

File : Question.ts

import { interval } from 'rxjs';

 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();

  }

Call Start Counter

ngOnInit(): void {

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

*this*.getAllQuestions();

*this*.startCounter();

  }

Reset Quiz on clicking reset btn

File : question.ts

 resetQuiz() {

*this*.resetCounter();

*this*.getAllQuestions();

*this*.points = 0;

*this*.counter = 60;

*this*.currentQuestion = 0;

*this*.progress = "0";

  }

For Progress Bar

File : Question.html

   <div class="progress mb-3">

File : question.ts

getProgressPercent() {

*this*.progress = ((*this*.currentQuestion / *this*.questionList.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) { }

  getQuestionsJson(){

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

}

### App-routing.module.ts



To make this router work, we need to set the router outlet in app.component.html

<router-outlet> </router-outlet>

### Changing the background option of correct ans:

At terminal :

ng g d change\_bg

delete spec file

File : change\_bg.directive

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

@Directive({

  selector: '[appChangeBg]'

})

export class ChangeBgDirective {

  @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');

    }

  }

}

### Result display

File: question.html

<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>

}

### Questions.json

{

    "questions": [{

            "questionText": "AngularJS is a \_\_\_\_",

            "options": [{

                    "text": "Java Framework "

                },

                {

                    "text": "JavaScript Framework",

                    "correct": true

                },

                {

                    "text": "HTML Framework"

                },

                {

                    "text": "SQL framework"

                }

            ],

            "explanation": ""

        },

        {

            "questionText": "AngularJS is perfect for? ",

            "options": [{

                    "text": "MPAs"

                 },

                {

                    "text": " DPAs"

                },

                {

                    "text": "CPAs"

                },

                {

                    "text": "SPAs",

                    "correct": true

                }

            ],

            "explanation": ""

        },

        {

            "questionText": "AngularJS provides reusable components",

            "options": [{

                    "text": "true",

                    "correct": true

                },

                {

                    "text": "false"

                }

            ],

            "explanation": ""

        },

        {

            "questionText": "Choose the correct syntax for writing AngularJS expression.",

            "options": [{

                    "text": "[expression]"

                },

                {

                    "text": "{{expressions}}",

                    "correct": true

                },

                {

                    "text": "{{{expressions}}}"

                },

                {

                    "text": "{expressions}"

                }

            ],

            "explanation": ""

        },

        {

            "questionText": "Choose the reactor which is used to bind the application data to the HTML view in AngularJS.",

            "options": [{

                    "text": "ng-bind directive",

                    "correct": true

                },

                {

                    "text": "ng-model directive"

                },

                {

                    "text": "ng-init directive"

                },

                {

                    "text": "ng-app directive"

                }

            ],

            "explanation": ""

        },

        {

            "questionText": "Who is the father of AngularJS?",

            "options": [{

                    "text": "Misko Hevery",

                    "correct": true

                },

                {

                    "text": "Brad Green"

                },

                {

                    "text": "Mike Adams"

                },

                {

                    "text": "  Adam a bronze"

                }

            ],

            "explanation": ""

        },

        {

            "questionText": "Angular JS is completely based on HTML and JavaScript.",

            "options": [{

                    "text": "True",

                    "correct": true

                },

                {

                    "text": "false"

                }

            ],

            "explanation": ""

        },

        {

            "questionText": "Among the following choose the one which is responsible for the startup of the angular 2 projects.",

            "options": [{

                    "text": "main.ts",

                    "correct": true

                },

                {

                    "text": "app.ts"

                },

                {

                    "text": "index.ts"

                },

                {

                    "text": "angular.cli.json"

                }

            ],

            "explanation": ""

        },

        {

            "questionText": "How is interpolation in angular 2 done?",

            "options": [{

                    "text": "{{ }}",

                    "correct": true

                },

                {

                    "text": "{{{ }}}"

                },

                {

                    "text": "{{|var}}"

                },

                {

                    "text": "!!!!"

                }

            ],

            "explanation": ""

        }

    ]

}