

# Phase 4 Project

## Online Test Application - Codes

M.Ramakrishna

### quiz.component.html

```
<div class="container p-5" style="background-color: ghostwhite; font-size: x-
large;">
  <div class="row">
    <div class="col-sm-8 offset-2">
      <div *ngIf="currentQuiz<9" ><!--need to make changes in this line w.r.t
to '9'-->
        <h2 style="text-align: center;">The HTML Quiz</h2>
        <p style="background-color: azure;" class = " text-center "
style="text-align: center;">{{currentQuiz +1}} of {{quizzes.length+1}}</p>
        <hr/>
        <h1 >{{ quizzes[currentQuiz].question }}</h1>
        <br>
        <ul class="list-group" *ngFor="let quiz of
quizzes[currentQuiz].answer">
          <label>
            <li class="list-group-item">
              <input type="radio" name="options"
(change)="onAnswer(quiz.correct)" [disabled]="answerSelected">
                {{quiz.option}}
            </li>
          </label>
        </ul>
      </div>
      <div *ngIf="currentQuiz > 8"><!--to make changes here w.r.t. '8' -->
        <button class = "btn btn-success btn-block"
(click)="showResult()">Show Result</button>
        <div *ngIf="result">
          Correct Answers: {{correctAnswers}} || Incorrect Answers :
          {{incorrectAnswers}}
        </div>
      </div>
    </div>
  </div>
</div>
```

### quiz.service.ts

```
import { Injectable } from '@angular/core';
import { Quiz } from '../quiz.model';
```

```

@Injectables({
  providedIn: 'root'
})
export class QuizService {
  quizzes: Quiz[]=[
    {
      question : 'What does HTML stand for ',
      answer:[
        {option: 'Hyperlink Markup Language ' , correct : false},
        {option: 'Hyperlink and Text Markup Language ' , correct : false},
        {option: 'Hyper Text Markup Language ' , correct : true},
        {option: 'Home Tool Markup Language ' , correct :false }
      ]
    },
    {
      question : 'Choose the correct HTML tag for the largest heading ',
      answer:[
        {option: '<heading> ' , correct : false},
        {option: '<h6> ' , correct : false},
        {option: '<head> ' , correct : false},
        {option: '<h1> ' , correct :true }
      ]
    },
    {
      question : 'What is the correct HTML tag for inserting a line-break ',
      answer:[
        {option: '<hr> ' , correct : false},
        {option: '<lb> ' , correct : false},
        {option: '<br> ' , correct : true},
        {option: '<break> ' , correct :false }
      ]
    },
    {
      question : 'How do you make a numbered list ',
      answer:[
        {option: '<ol> ' , correct : true},
        {option: '<ul> ' , correct : false},
        {option: '<dl> ' , correct : false},
        {option: '<list> ' , correct :false }
      ]
    },
    {
      question : 'What is the corerct HTML for making a checkbox ',
      answer:[
        {option: '<input type = "checkbox"> ' , correct : true},
        {option: '<checkbox> ' , correct : false},

```

```

    {option: '<check> ' , correct : false},
    {option: '<input type = "check"> ' , correct :false }

  ],
},
{
  question : 'What does CSS stand for ',
  answer:[
    {option: 'Cascading Style Sheets ' , correct : true},
    {option: 'Creative Style Sheets ' , correct : false},
    {option: 'Computer Style Sheets ' , correct : false},
    {option: 'Colorful Style Sheets ' , correct :false }

  ],
},
{
  question : 'What is the correct CSS syntax ',
  answer:[
    {option: 'body {color: black;} ' , correct : true},
    {option: 'body:color=black; ' , correct : false},
    {option: '{body:color=black;} ' , correct : false},
    {option: 'body = black; ' , correct :false }

  ],
},
{
  question : 'What property is used to change the background color ',
  answer:[
    {option: 'color ' , correct : false},
    {option: 'bgcolor ' , correct : false},
    {option: 'background-color ' , correct : true},
    {option: 'bgc ' , correct :false }

  ],
},
{
  question : 'What property is used to change th left margin of an element ',
  answer:[
    {option: 'padding-left ' , correct : true},
    {option: 'margin-left ' , correct : false},
    {option: 'margin ' , correct : false},
    {option: 'indent ' , correct :false }

  ],
},
]

constructor() { }

```

```

    getQuizzes()
    {
        return this.quizzes;
    }
}

```

### quiz.model.ts

```

export class Quiz
{
    question :string | undefined;
    answer : { option: string; correct: boolean; }[] | undefined;
}

```

### quiz.component.ts

```

import { Component, OnInit } from '@angular/core';
import { Quiz } from '../quiz.model';
import { QuizService } from '../quiz.service';

@Component({
    selector: 'app-quiz',
    templateUrl: './quiz.component.html',
    styleUrls: ['./quiz.component.css']
})
export class QuizComponent implements OnInit {
    quizzes : Quiz[] =[];
    currentQuiz=0;//this will return the first question
    answerSelected=false;
    correctAnswers = 0 ;
    incorrectAnswers = 0;

    randomize:number | undefined;

    result=false;

    constructor(private quizService : QuizService) { }

    ngOnInit(): void {
        this.quizzes=this.quizService.getQuizzes();
        //this.randomize=Math.floor(Math.random()*this.quizzes.length)
    }

    onAnswer(option:boolean)
    {
        this.answerSelected = true;
        setTimeout(() => {
            this.currentQuiz++;

```

```
        this.answerSelected = false; // we do this so that we can re click the
radio button
    }, 300);

    if(option)
    {
        this.correctAnswers++;
    }
    else
    {
        this.incorrectAnswers++;
    }
}
showResult()
{
    this.result=true;
}
}
```