

# Online Test Application

## Source Code

Create an Angular App

And adding the Bootstrap5 links in **index.html**

### **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-
1BmE4kWBq78iYhF1dvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
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-
ka7Sk0GlIn4gmtz2MlQnikT1wXgYs0g+OMhuP+IlRH9sENB00LRn5q+8nbTov4+1p"
    crossorigin="anonymous"></script>
    <app-root></app-root>
</body>

</html>

```

Adding the router-outlet and route components into the **app.component.html**

**app.component.html**

```
<app-header></app-header>
```

```
<router-outlet></router-outlet>

<app-footer></app-footer>
```

Now creating the components for welcome and question.

And adding the frontend UI into both components.

### 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 9 questions. Each question
hold 10 points </p>

  <h4><u>Rules:</u></h4>

  <ol>

    <li>Correct question gives 10 points</li>

    <li>Incorrect question gives -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>

  <div class="name col-md-4 my-4">

    <label for="">Enter Your Name:</label>

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

  </div>

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

```
</div>
```

### question.component.html

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

  <div class="card">

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

      <div class="image">

        

      </div>

      <div class="quiz-header">

        <h4 style="font-family: cursive;"><u>Angular and Typescript
Quiz</u></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-remaining">

          <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-2x" aria-
hidden="true"></i></button>

        <button class="btn" (click)="resetQuiz()"><i class="fa fa-
refresh text-primary fa-2x"
        aria-hidden="true"></i></button>

        <button class="btn" (click)="nextQuestion()"><i class="fa
text-primary fa-chevron-right fa-2x"
        aria-hidden="true"></i></button>

    </div>
</ng-container>

<ng-container *ngIf="isQuizCompleted">
    <div class="row d-flex justify-content-between">

        

        <div class="result text-center col-md-6 col-sm-12">
            <h3 style="font-family: cursive;">Congratulations!! </h3>
            <h4><br> You have completed the quiz. <br>Below is your
result:</h4>

            <p>Total question attempted: {{questionList.length}} </p>
            <p>Total Correct Answers: {{correctAnswer}}</p>
            <p>Total Wrong Answers: {{inCorrectAnswer}} </p>
            <p>Your Score: {{points}} Points </p>

        </div>
    </div>

```

```

</div>

<!-- Button trigger modal -->

<button type="button" class="btn btn-primary" data-bs-
toggle="modal" data-bs-target="#exampleModal">
    View the Correct Answers
</button>

<!-- Modal -->

<div class="modal fade" id="exampleModal" tabindex="-1" aria-
labelledby="exampleModalLabel"
    aria-hidden="true">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title"
id="exampleModalLabel">Correct Answers</h5>
                <button type="button" class="btn-close" data-bs-
dismiss="modal" aria-label="Close"></button>
            </div>
            <div class="modal-body">
                <div>
                    <p>1. Which of the following does TypeScript
use to specify types?</p><br>
                    <p>Answers: <b>:</b></p><br>

```

```
<p>Explanation: TS uses a colon (:) to
separate the property name from the property type</p>

</div>

<div>

<p>2. Which of the following is NOT a type
used in TypeScript?</p><br>

<p>Answers: <b>enum</b></p><br>

<p>Explanation: enum is not used as a type in
TypeScript</p>

</div>

<div>

<p>3. How can we specify properties and
methods for an object in TypeScript?</p><br>

<p>Answers: <b>Use interfaces</b></p><br>

<p>Explanation: interfaces are typically used
to list the properties and methods for an object</p>

</div>

<div>

<p>4. How else can Array number be written in
TypeScript?</p><br>

<p>Answers: <b>number[]</b></p><br>

<p>Explanation: number[] is another way of
writing Array number in TypeScript</p>

</div>

<div>

<p>5. In which of these does a class take
parameters?</p><br>

<p>Answers: <b>constructor</b></p><br>
```



```
<p>Explanation: a constructor is used by a
class to take in parameters</p>

</div>

<div>

    <p>6. Which is NOT an access modifier?</p><br>

    <p>Answers: <b>async</b></p><br>

    <p>Explanation: async is not used as an access
modifier type in TypeScript</p>

</div>

<div>

    <p>7. Which keyword allows us to share
information between files in TypeScript?</p><br>

    <p>Answers: <b>contructor</b></p><br>

    <p>Explanation: the export keyword allows for
the information to be transmitted between files</p>

</div>

<div>

    <p>8. Which is an array method to generate a
new array based on a condition?</p><br>

    <p>Answers: <b>filter</b></p><br>

    <p>Explanation: filter is a method used to
conditionally create a new array</p>

</div>

<div>

    <p>9. How is a property accessible within a
class?</p><br>

    <p>Answers: <b>Using
this.propertyName</b></p><br>
```

```

        <p>Explanation: this.propertyName is the way
to access a specific property within a class</p>
    </div>

</div>

<div class="modal-footer">
    <button type="button" class="btn btn-secondary"
data-bs-dismiss="modal">Close</button>
</div>
</div>
</div>
</div>
</ng-container>
</div>
</div>

```

Also, adding little styling in the **question.component.scss**

**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;

}

```

Generating and adding the header and footer parts also

### **header.component.html**

```

<nav>

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

        <div class="container-fluid">

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

```

```

        <span class="text-uppercase">angularQuiz</span>
    </a>
</div>
</nav>
</nav>

```

### **footer.component.html**

```

<footer class=" bg-dark fixed-bottom container-fluid">
    <!-- Copyright -->
    <div class="text-center p-3" style="background-color: rgba(0, 0, 0,
0.2);">
        <p style="font-size:large" class="text-white">All Rights Reserved.
Copyright © 2022 || Angular Quiz by Bikki Shaw</p>
    </div>
    <!-- Copyright -->
</footer>

```

Adding the components into the **app-routing.module.ts**

### **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 { }

```

The modules will automatically add to the **app.module.ts**

### **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';
import { FooterComponent } from './footer/footer.component';
```

```
@NgModule({
  declarations: [
    AppComponent,
    WelcomeComponent,
    QuestionComponent,
    HeaderComponent,
    ChangeBgDirective,
    FooterComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    HttpClientModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

Creating a json file from where the questions will be fetched and display at front-end.

Create the **questions.json** inside the assets folder of the Angular project.

### **questions.json**

```
{
  "questions": [
    {
      "questionText": "1. Which of the following does TypeScript use to specify types?",
      "options": [
        {
          "text": ":",
          "correct": true
        },
        {
          "text": ";"
        },
        {
          "text": "!"
        },
        {
          "text": "&"
        }
      ]
    }
  ],
}
```

```
        "explanation": "TS uses a colon (:) to separate the property name
from the property type"
    },
    {
        "questionText": "2. Which of the following is NOT a type used in
TypeScript?",
        "options": [
            {
                "text": "number"
            },
            {
                "text": "string"
            },
            {
                "text": "boolean"
            },
            {
                "text": "enum",
                "correct": true
            }
        ],
        "explanation": "enum is not used as a type in TypeScript"
    },
    {
        "questionText": "3. How can we specify properties and methods for
an object in TypeScript?",
        "options": [
            {
```



```
        "text": "Use classes."
    },
    {
        "text": "Use interfaces.",
        "correct": true
    },
    {
        "text": "Use enums."
    },
    {
        "text": "Use async/await."
    }
],
    "explanation": "interfaces are typically used to list the
properties and methods for an object"
},
{
    "questionText": "4. How else can Array<number> be written in
TypeScript?",
    "options": [
        {
            "text": "@number"
        },
        {
            "text": "number[]",
            "correct": true
        },
        {
```

```
        "text": "number!"
    },
    {
        "text": "number?"
    }
],
    "explanation": "number[] is another way of writing Array<number>
in TypeScript"
},
{
    "questionText": "5. In which of these does a class take
parameters?",
    "options": [
        {
            "text": "constructor",
            "correct": true
        },
        {
            "text": "destructor"
        },
        {
            "text": "import"
        },
        {
            "text": "subscribe"
        }
    ],
}
```

```
        "explanation": "a constructor is used by a class to take in
parameters"
    },
    {
        "questionText": "6. Which is NOT an access modifier?",
        "options": [
            {
                "text": "private"
            },
            {
                "text": "protected"
            },
            {
                "text": "public"
            },
            {
                "text": "async",
                "correct": true
            }
        ],
        "explanation": "async is not used as an access modifier type in
TypeScript"
    },
    {
        "questionText": "7. Which keyword allows us to share information
between files in TypeScript?",
        "options": [
            {
```

```
        "text": "import"
      },
      {
        "text": "export",
        "correct": true
      },
      {
        "text": "async"
      },
      {
        "text": "constructor"
      }
    ],
    "explanation": "the export keyword allows for the information to
be transmitted between files"
  },
  {
    "questionText": "8. Which is an array method to generate a new
array based on a condition?",
    "options": [
      {
        "text": "filter",
        "correct": true
      },
      {
        "text": "map"
```

```
        "text": "async"
      },
      {
        "text": "enum"
      }
    ],
    "explanation": "filter is a method used to conditionally create a
new array"
  },
  {
    "questionText": "9. How is a property accessible within a class?",
    "options": [
      {
        "text": "Using this.propertyName",
        "correct": true
      },
      {
        "text": "Accessors"
      },
      {
        "text": "Destructuring"
      },
      {
        "text": "Arrow function"
      }
    ],
    "explanation": "this.propertyName is the way to access a specific
property within a class"
```

```
    }  
  ]  
}
```

Adding the functionality in welcome component, by editing in **welcome.component.ts**

**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);  
  }  
}
```

```
}
```

Adding the functionality into the **question.component.html**, adding the necessary changes.

**question.component.html**

```
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;
  }
}

```

Generating the question as a service, and adding the necessary changes into the file name.

### **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");
}
}

```

Finally, creating a directive (**change-bg**) for changing colour in options and add the necessary changes into the mentioned file.

#### **change-bg.directive.ts**

```

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');  
}  
}  
  
}
```