Source Code For Online Test Application

1. Header component code

Header.component.html

```
<nav class="navbar navbar-dark bg-primary">
    <div class="container-fluid">
        <a class="navbar-brand text-bold" href="#">
      <img src="https://img.icons8.com/external-smashingstocks-flat-smashing-
stocks/452/external-online-test-distance-education-smashingstocks-flat-
smashing-stocks.png" width="50" height="50" alt="logo">
      Online Test Application <span class="badge badge-secondary">Quiz</span>
        </a>
    </div>
</nav>
```

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 {
  }

}
```

2. Question Component

Question.component.html

```
<div class="container mt-5 mb-5">
    <div class="card">
        <div class="d-flex justify-content-between p-3">
            <div class="image">
                <img
src="https://www.easilearnmusic.co.uk/uploads/1/2/8/2/12829937/quizzes_orig.pn
g"
                    width="100" height="100" alt="logo">

            </div>
            <div class="quiz-header">
                <h4 style="font-family: 'Franklin Gothic Medium', 'Arial
Narrow', Arial, sans-serif;"><b>Core Java Quiz</b></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: normal">Question
{{currentQuestion+1}} of {{questionList.length}}</span>


                </div>

                <div class="timer">
                    <h5>{{counter}} sec <i class="fas fa-clock" style="font-
size:24px;color:red;"></i> </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">
                    <h5 style="font-style:
normal;"><b>{{questionList[currentQuestion]?.questionText}}</b></h5>
                </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="card">
            <div class="container-fluid">
                <div class="row mt-3 mx-2">
                    <div class="col-md-2">
                        <div class="row d-flex justify-content-between">
                            <img class="img-fluid col-sm-12 mx-auto"
                                src="https://th.bing.com/th/id/OIP.wlOtwDspadt
f5hecgvN3IAHaHa?pid=ImgDet&rs=1"
```

```
                                width="70" height="70" alt="">
                        </div>
                    </div>
                    <div class="col-md-10">
                        <div class="result text-justify col-md-6 col-sm-12">
                            <h4 style="color: rgb(0, 191,
255);"><b>Congratulations {{name}} You have completed the Quiz:
<br><br></b></h4>
                            <h5 style="color:darkslateblue"><b>Your ScoreCard
is shown below: </b></h5>
                            <p style="color: green;"><b>Total Question
Attempted: {{questionList.length}} </b></p>
                            <p style="color: green;"><b>Total Correct Answered
: {{correctAnswer}} </b></p>
                            <p style="color: green;"><b>Total InCorrect
Anwered: {{inCorrectAnswer}} </b></p>
                            <p style="color: green;"><b>Your Score: {{points}}
Points</b></p>
                        </div>
                    </div>

                </div>
            </div>
        </div>
        <div class="card mt-3">
            <div class="container-fluid">
                <div class="row mt-3 mx-2">
                    <div class="col-md-2">
                        <div class="row d-flex justify-content-between">
                            <img class="img-fluid col-sm-12 mx-auto"
                                src="https://media.istockphoto.com/vectors/cre
dit-score-indicator-set-vector-
id880382670?k=6&m=880382670&s=170667a&w=0&h=bfaAm2goCGCvzaEf2XNCK2yClQywkTIoUs
CoaHt9-ZY="
                                width="70" height="70" alt="">
                        </div>
                    </div>
                    <div class="col-md-10">
                        <div class="result text-justify col-md-6 col-sm-12">
                            <h4 style="color: rgb(64, 64, 233);"><b>Your
Review Chart</b></h4>
                            <p style="color: brown ;"><b>Your Score:
{{points}} Points</b></p>
                            <p style="color: blueviolet;"><b>Analyse your
score based on the below criteria: </b></p>
                            <p style="color: red;"><b>Points <
20:  Fail</b></p>
```

```
                            <p style="color: red;"><b>Points 30 To 40 :Work
More</b></p>
                            <p style="color: green;"><b>Points are 40 to 60 :
Good</b></p>
                            <p style="color: green;"><b>Points are 40 to 70 :
Very Good</b></p>
                            <p style="color: green;"><b>Points are 70 to 90 :
Excellent</b></p>
                        </div>
                    </div>
                </div>
            </div>
        </div>
</ng-container>


    </div>
</div>
```

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 => {
      //console.log(res.questions);
      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 = this.points+10;
    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;
  }

}
```

3. Service Component

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");

  }

}
```

4. Welcome Page Component

Welcome.component.html

```
<div class="container">
    <div class="row">
        <div class="col-md-12 ">
            <div class="card mt-3 mb-3">

                <div class="card-header bg-light">
                    <h1 class="display-5  fw-bold " style="color:rgb(0, 98,
211);"> Welcome to Online Test Application</h1>
                </div>

                <div class="card-body">
                    <p class="col-md-8 fs-4">This application enables
                            users to take online tests, review them, and
display the results at the same time.<br>
                    </p>
                    <h4 style="color:rgb(0, 98, 211);">Quiz Rules:</h4>
                    <ol>
                        <li> Each correct answer gives you 10 points and each
incorrect answer will result in -10 points.</li>
                        <li> You will have 60 seconds to answer each
question.</li>
                        <li> Refreshing the Page will reset the Quiz and you
have to start from the beginning.
                        </li>
                    </ol>
                </div>
                <div class="card-footer bg-light text-center">
                    <h1 style="font-family:Georgia, 'Times New Roman', Times,
serif;">All the Best...!!!</h1>
                </div>
            </div>
        </div>
    </div>

    <div class="name col-md-4 my-3">
        <label> <b>Enter your name </b></label>
        <input #name type="text" class="form-control" placeholder="Enter your
name here">
    </div>
    <div class="container">
```

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

</div>

<div class="card text-center">
    <!-- <div class="card-header badge-secondary">
        <p class="card-text"><b>Owned By © 2022 Copyright: By </b><a
style="color:whitesmoke;"
                href="#!">OnlineTestApplication.com</a>
    </div> -->
</div>
```

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);
  }

}
```

5. 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 { }
```

6. questions.json

```
{
    "questions": [
        {
            "questionText": "Which of the following is not a keyword in
java?",
            "options": [
                {
                    "text": "Exception",
                    "correct": true
                },
                {
```

```
                    "text": "static"
                },
                {
                    "text": "void"
                },
                {
                    "text": "final"
                }
            ],
            "explanation": "Exception is a class in Java not a keyword"
        },
        {
            "questionText": " What is class variable?",
            "options": [
                {
                    "text": " class variables are variables defined inside
methods, constructors or blocks."
                },
                {
                    "text": "class variables are variables within a class but
outside any method."
                },
                {
                    "text": "class variables are always private"
                },
                {
                    "text": "class variables are static variables within a
class but outside any method.",
                    "correct": true
                }
            ],
            "explanation": "static variables in Java are called Class
variables as they have class scope and are common to all objects of class"
        },
        {
            "questionText": "What is true about a final class?",
            "options": [
                {
                    "text": " class declard final is a final class."
                },
                {
                    "text": "Final classes are created so the methods
implemented by that class cannot be overridden."
                },
                {
                    "text": "It can't be inherited."
                },
                {
```

```
                    "text": "All of the above.",
                    "correct": true
                }
            ],
            "explanation": "All the options are true and explain a fact about
final class"
        },
        {
            "questionText": "Which of these keywords are used for the block to
be examined for exceptions?",
            "options": [
                {
                    "text": "check"
                },
                {
                    "text": "try",
                    "correct": true
                },
                {
                    "text": "catch"
                },
                {
                    "text": "true"
                }
            ],
            "explanation": " try is used for the block that needs to checked
for exceptions"
        },
        {
            "questionText": "Which one of the following is not an access
modifier?",
            "options": [
                {
                    "text": "void",
                    "correct": true
                },
                {
                    "text": "protected"
                },
                {
                    "text": "public"
                },
                {
                    "text": "private"
                }
            ],
            "explanation": "public, private, protected and default are the
only access modifiers in Java, void is a keyword"
```

```
        },
        {
            "questionText": "What is true about constructor?",
            "options": [
                {
                    "text": " It can contain return type"
                },
                {
                    "text": "It can have any non access modifiers"
                },
                {
                    "text": "Constructor cannot throw an exception"
                },
                {
                    "text": " It can take any number of parameters",
                    "correct": true
                }
            ],
            "explanation": "Constructor can have any number of parameters"
        },
        {
            "questionText": "What is not the use of “this” keyword in Java?",
            "options": [
                {
                    "text": "It represents a return type"
                },
                {
                    "text": "It is a reference variable that refers to the
current object",
                    "correct": true
                },
                {
                    "text": "It is an access modifier"
                },
                {
                    "text": "calls a parameterized constructor"
                }
            ],
            "explanation": "In Java this keyword refers to current class
object"
        },
        {
            "questionText": "Which of the following is a valid declaration of
an object of class Box?",
            "options": [
                {
                    "text": " Box obj = new Box();",
                    "correct": true
```

```
                },
                {
                    "text": "Box obj = new Box;"
                },
                {
                    "text": "obj = new Box();"
                },
                {
                    "text": "new Box obj;"
                }
            ],
            "explanation": "Box obj = new Box(); is the right way"
        },
        {
            "questionText": "Which of these is used to allocate memory for an
object?",
            "options": [
                {
                    "text": "new",
                    "correct": true
                },
                {
                    "text": "class"
                },
                {
                    "text": "obj"
                },
                {
                    "text": "malloc"
                }
            ],
            "explanation": "new keyword helps to allocate memory for an object
in heap"
        },

        {
            "questionText": "Finally block is attached to?",
            "options": [
                {
                    "text": "Try-catch block",
                    "correct": true
                },
                {
                    "text": "Class block"
                },
                {
                    "text": "Method block"
                },
```

```
                {
                    "text": "All of these"
                }
            ],
            "explanation": "Finally, block of code runs at the end of the try-
catch block"
        }
    ]
}
```