**Source Code:**

**Index.html**

<!doctype html>

<html lang="en">

<head>

  <meta charset="utf-8">

  <title>OnlineTestApplication</title>

  <base href="/">

  <meta name="viewport" content="width=device-width, initial-scale=1">

  <link rel="icon" type="image/x-icon" href="favicon.ico">

</head>

<body>

  <app-root></app-root>

</body>

</html>

**Style.css**

body{

  min-height: 100vh;

  display: flex;

  flex-direction: column;

}

.btn {

  border-radius: 0;

}

.option {

  background-color: lightgray;

  font-size: 20px;

  padding: 10px;

  margin: 3px;

}

#quiz .options input[type=checkbox] {

  height: 22px;

  width: 22px;

}

.cursor-pointer {

  cursor: pointer;

}

.result-question {

  background-color: #eee;

  margin: 4px;

  padding: 6px;

}

@media only screen and (max-width: 480px) {

  h1,

  .h1 {

    font-size: 22px;

  }

  h2,

  .h2 {

    font-size: 20px;

  }

  h3,

  .h3 {

    font-size: 18px;

  }

  .option {

    font-size: 16px;

    padding: 6px;

  }

}

.card-img-top {

  width: 100%;

  height: 15vw;

  object-fit: cover;

}

footer{

  margin-top: auto;

}

**AppComponent.html**

<nav class="navbar navbar-expand-lg navbar-dark bg-primary">

  <a class="navbar-brand ml-2 fw-bolder" routerLink="/quiz">Simplilearn Quiz</a>

  <button

    class="navbar-toggler"

    type="button"

    data-toggle="collapse"

    data-target="#navbarNav"

    aria-controls="navbarNav"

    aria-expanded="false"

    aria-label="Toggle navigation"

  >

    <span class="navbar-toggler-icon"></span>

  </button>

  <div class="collapse navbar-collapse" id="navbarNav">

    <ul class="navbar-nav ms-auto">

      <li class="nav-item">

        <a

          class="nav-link btn-outline-primary btn-sm text-light fw-bolder"

          routerLink="/quiz"

          >Home</a

        >

      </li>

    </ul>

  </div>

</nav>

<div class="container">

  <router-outlet></router-outlet>

</div>

<footer class="bg-primary text-center text-lg-start mt-5">

  <div class="text-center p-3 text-light fw-bold">

    Online Test Application

    <a class="text-light fw-bold" routerLink="/quiz">@simplilearn</a>

  </div>

  <div>

    <a class="text-light fw-bold" routerLink="/javascript">JavaScript</a>

  </div>

  <div>

    <a class="text-light fw-bold" routerLink="/angular">Angular</a>

  </div>

  <div>

  <a class="text-light fw-bold" routerLink="/corejava">Core Java </a>

  </div>

</footer>

**App.component.ts**

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

@Component({

  selector: 'app-root',

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

  styleUrls: ['./app.component.css']

})

export class AppComponent {

  title = 'online-test-application';

}

**App.module.ts**

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

import { FormsModule } from '@angular/forms';

import { BrowserModule } from '@angular/platform-browser';

import { AppComponent } from './app.component';

import { HttpClientModule } from '@angular/common/http';

import { QuizService } from './services/quiz.service';

import { QuizListComponent } from './components/quiz-list/quiz-list.component';

import { QuizComponent } from './components/quiz/quiz.component';

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

const routes: Routes = [

  { path: 'quiz/:quizName', component: QuizComponent },

  { path: 'quiz', component: QuizListComponent },

  { path: '', redirectTo: '/quiz', pathMatch: 'full' },

  { path: '\*\*', redirectTo: '/quiz', pathMatch: 'full' },

];

@NgModule({

  declarations: [AppComponent, QuizListComponent, QuizComponent],

  imports: [

    BrowserModule,

    FormsModule,

    HttpClientModule,

    RouterModule.forRoot(routes),

  ],

  providers: [QuizService],

  bootstrap: [AppComponent],

})

export class AppModule {}

**quiz.component.html**

<div id="quiz" class="container mt-5">

  <h2 class="text-center fw-bold text-uppercase">{{ quiz.name }}</h2>

  <hr />

  <div \*ngIf="mode == 'quiz' && quiz">

    <div \*ngFor="let question of filteredQuestions">

      <div class="mb-4">

        <span class="badge bg-success">

          Questions {{ pager.index + 1 }} of {{ pager.count }}

        </span>

        <span \*ngIf="config.duration" class="badge bg-success float-end">

          Time Left: {{ ellapsedTime }} / {{ duration }}

        </span>

      </div>

      <h3 class="fw-normal mb-4">

        {{ pager.index + 1 }}.

        <span [innerHTML]="question.name"></span>

      </h3>

      <div class="row text-left options">

        <div class="col-6" \*ngFor="let option of question.options">

          <div class="option">

            <label class="fw-normal" [attr.for]="option.id">

              <input

                id="{{ option.id }}"

                type="checkbox"

                [(ngModel)]="option.selected"

                (change)="onSelect(question, option)"

              />

              {{ option.name }}

            </label>

          </div>

        </div>

      </div>

    </div>

    <hr />

    <br />

    <div class="text-center">

      <button

        class="btn btn-outline-info mx-2"

        \*ngIf="config.allowBack"

        (click)="goTo(0)"

      >

        First

      </button>

      <button

        class="btn btn-outline-info mx-2"

        \*ngIf="config.allowBack"

        (click)="goTo(pager.index - 1)"

      >

        Prev

      </button>

      <button

        class="btn btn-outline-info mx-2"

        (click)="goTo(pager.index + 1)"

      >

        Next

      </button>

      <button

        class="btn btn-outline-info mx-2"

        \*ngIf="config.allowBack"

        (click)="goTo(pager.count - 1)"

      >

        End

      </button>

    </div>

    <br />

  </div>

  <div class="row text-center" \*ngIf="mode == 'review'">

    <div

      class="col-4 cursor-pointer"

      \*ngFor="let question of quiz.questions; let index = index"

    >

      <div

        (click)="goTo(index)"

        class="p-3 mb-2 {{

          isAnswered(question) == 'Answered' ? 'bg-info' : 'bg-warning'

        }}"

      >

        {{ index + 1 }}. {{ isAnswered(question) }}

      </div>

    </div>

  </div>

  <div class="result" \*ngIf="mode == 'result'">

    <h2>

      Quiz Result:

      <span class="badge bg-primary"

        >Score: {{ quizScore }} / {{ quizTotalScore }}</span

      >

    </h2>

    <div \*ngFor="let question of quiz.questions; let index = index">

      <div class="result-question">

        <h5>{{ index + 1 }}. {{ question.name }}</h5>

        <div class="row">

          <div class="col-6" \*ngFor="let Option of question.options">

            <input

              id="{{ Option.id }}"

              type="checkbox"

              disabled="disabled"

              [(ngModel)]="Option.selected"

            />

            {{ Option.name }}

          </div>

        </div>

        <div

          class="p-1 m-2 alert {{

            isCorrect(question) == 'correct' ? 'alert-success' : 'alert-danger'

          }}"

        >

          Answer is {{ isCorrect(question) }}.

        </div>

      </div>

    </div>

    <h4 class="alert alert-info text-center">Quiz Completed</h4>

  </div>

  <hr />

  <div \*ngIf="mode != 'result'">

    <button class="btn btn-info mx-2" (click)="mode = 'quiz'">Home</button>

    <button class="btn btn-info mx-2" (click)="mode = 'review'">Review</button>

    <button class="btn btn-success mx-2" (click)="onSubmit()">

      Submit

    </button>

  </div>

</div>

**Quiz.component.ts**

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

import { ActivatedRoute } from '@angular/router';

import { Option } from 'src/app/models/option';

import { Question } from 'src/app/models/question';

import { Quiz } from 'src/app/models/quiz';

import { QuizConfig } from 'src/app/models/quiz-config';

import { QuizService } from 'src/app/services/quiz.service';

@Component({

  selector: 'app-quiz',

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

  styleUrls: ['./quiz.component.css'],

})

export class QuizComponent implements OnInit {

  quizes: any[];

  quiz: Quiz = new Quiz(null);

  mode = 'quiz';

  quizName: string;

  quizScore: number;

  quizTotalScore: number;

  config: QuizConfig = {

    allowBack: true,

    allowReview: true,

    autoMove: false,

    duration: 420, // indicates the time (in secs) in which quiz needs to be completed. 0 means unlimited.

    pageSize: 1,

    requiredAll: false,

    richText: false,

    shuffleQuestions: false,

    shuffleOptions: false,

    showClock: false,

    showPager: true,

    theme: 'none',

  };

  pager = {

    index: 0,

    size: 1,

    count: 1,

  };

  timer: any = null;

  startTime: Date;

  endTime: Date;

  ellapsedTime = '00:00';

  duration = '';

  constructor(

    private quizService: QuizService,

    private route: ActivatedRoute

  ) {}

  ngOnInit() {

    this.quizes = this.quizService.getAll();

    this.quizScore = 0;

    this.route.paramMap.subscribe(() => {

      this.handleQuiz();

    });

  }

  handleQuiz() {

    const quizName: string = this.route.snapshot.paramMap.get('quizName');

    const quiz: any = this.quizes.find((o) => o.name === quizName);

    const quizUrl: string = quiz.id;

    this.loadQuiz(quizUrl);

  }

  loadQuiz(quizName: string) {

    this.quizService.get(quizName).subscribe((res) => {

      this.quiz = new Quiz(res);

      this.pager.count = this.quiz.questions.length;

      this.quizTotalScore = this.quiz.questions.length;

      this.startTime = new Date();

      this.ellapsedTime = '00:00';

      this.timer = setInterval(() => {

        this.tick();

      }, 1000);

      this.duration = this.parseTime(this.config.duration);

    });

    this.mode = 'quiz';

  }

  tick() {

    const now = new Date();

    const diff = (now.getTime() - this.startTime.getTime()) / 1000;

    if (diff >= this.config.duration) {

      this.onSubmit();

    }

    this.ellapsedTime = this.parseTime(diff);

  }

  parseTime(totalSeconds: number) {

    let mins: string | number = Math.floor(totalSeconds / 60);

    let secs: string | number = Math.round(totalSeconds % 60);

    mins = (mins < 10 ? '0' : '') + mins;

    secs = (secs < 10 ? '0' : '') + secs;

    return `${mins}:${secs}`;

  }

  get filteredQuestions() {

    return this.quiz.questions

      ? this.quiz.questions.slice(

          this.pager.index,

          this.pager.index + this.pager.size

        )

      : [];

  }

  onSelect(question: Question, option: Option) {

    if (question.questionTypeId === 1) {

      question.options.forEach((x) => {

        if (x.id !== option.id) x.selected = false;

      });

      if (this.isCorrect(question) == 'correct') {

        this.quizScore = this.quizScore + 1;

      }

    }

    if (this.config.autoMove) {

      this.goTo(this.pager.index + 1);

    }

  }

  goTo(index: number) {

    if (index >= 0 && index < this.pager.count) {

      this.pager.index = index;

      this.mode = 'quiz';

    }

  }

  isAnswered(question: Question) {

    return question.options.find((x) => x.selected)

      ? 'Answered'

      : 'Not Answered';

  }

  isCorrect(question: Question) {

    return question.options.every((x) => x.selected === x.isAnswer)

      ? 'correct'

      : 'wrong';

  }

  onSubmit() {

    let answers = [];

    this.quiz.questions.forEach((x) =>

      answers.push({

        quizId: this.quiz.id,

        questionId: x.id,

        answered: x.answered,

      })

    );

    this.mode = 'result';

  }

}

**ListQuiz.html**

<div class="container">

  <div class="row mt-5">

    <div \*ngFor="let tempQuiz of quizes" class="col mt-5">

      <div class="card">

        <div class="card-body text-left">

          <h5 class="card-title text-center">

            {{ tempQuiz.name }}

          </h5>

          <hr>

          <a

            routerLink="/quiz/{{ tempQuiz.name }}"

            class="btn btn-success mt-5"

            > {{ tempQuiz.name }} - Start Quiz</a

          >

        </div>

      </div>

    </div>

  </div>

</div>

**ListQuiz.ts**

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

import { Quiz } from 'src/app/models/quiz';

import { QuizService } from 'src/app/services/quiz.service';

@Component({

  selector: 'app-quiz-list',

  templateUrl: './quiz-list.component.html',

  styleUrls: ['./quiz-list.component.css'],

})

export class QuizListComponent implements OnInit {

  quizes: any[] = [];

  constructor(private quizService: QuizService) {}

  ngOnInit(): void {

    this.quizes = this.quizService.getAll();

  }

}