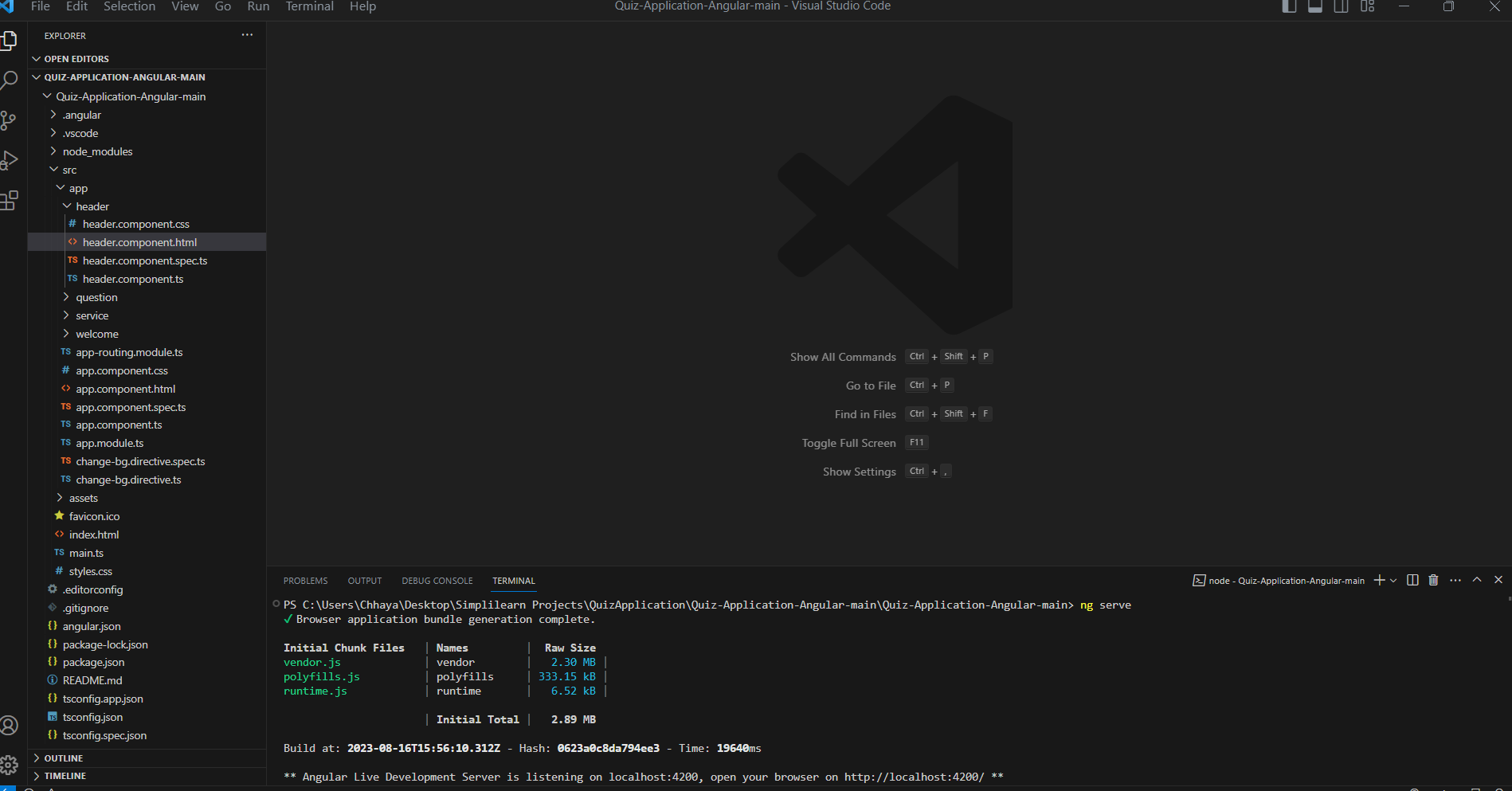
* **Home Main Page**



* app.component.module.routing:
* import { NgModule } from '@angular/core';
* import { RouterModule, Routes } from '@angular/router';
* import { WelcomeComponent } from './welcome/welcome.component';
* import { QuestionComponent } from './question/question.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 { }
* App.component.html:
* <app-header></app-header>
* <router-outlet></router-outlet>
* App.component.ts:

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

@Component({

  selector: 'app-root',

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

  styleUrls: ['./app.component.css']

})

export class AppComponent {

  title = 'MyQuizAppp';

}

* 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 { }
* Change-bg.directive.ts:
* import { Directive,HostListener, ElementRef,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','white');
* 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','white');
* this.render.setStyle(this.el.nativeElement, 'border','2px solid grey');
* }
* }
* }
* Indext.html:
* <!doctype html>
* <html lang="en">
* <head>
* <meta charset="utf-8">
* <title>MyQuizAppp</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@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-9ndCyUaIbzAi2FUVXJi0CjmCapSmO7SnpJef0486qhLnuZ2cdeRhO02iuK6FUUVM" crossorigin="anonymous">
* <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
* </head>
* <body>
* <app-root></app-root>
* <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js" integrity="sha384-geWF76RCwLtnZ8qwWowPQNguL3RmwHVBC9FhGdlKrxdiJJigb/j/68SIy3Te4Bkz" crossorigin="anonymous"></script>
* </body>
* </html>
* **Header Component:**
* 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/color/96/000000/angularjs.png" width="50" alt="logo">
* <span class="text-uppercase">ngular Quiz</span>
* </a>
* </div>
* </nav>
* header.component.ts:

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

@Component({

  selector: 'app-header',

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

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

})

export class HeaderComponent {

}

* **Question Component:**
* question.component.css:
* .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;
* }
* Question.component.html:

<div class="container mt-5">

    <div class="card">

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

            <div class="image">

                <img src="https://img.icons8.com/color/96/000000/angularjs.png"  width="90">

            </div>

            <div class="quiz-header">

                <h4 style="font-family: cursive;">Angular and TypeScript Quiz</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: italic;;">Question {{currentQuestion+1}} of {{questionList.length}}</span>

            </div>

            <div class="timer">

                {{counter}} sec ⏱

            </div>

        </div>

        <div class="progress mb-3">

            <div class="progress-bar progress-bar-striped bg-success" [ngStyle]="{'width':progress+'%'}"></div>

        </div>

        <div class="questions">

            <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-3x" aria-hidden="true"></i></button>

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

            <button [disabled]="currentQuestion===questionList.length-1" 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.component.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:any;
* public questionList :any=[];
* public currentQuestion: number=0;
* public points:number=0;
* counter=60;
* correctAnswer:number=0;
* incorrectAnswer:number=0;
* interval$:any;
* progress:string="0";
* isQuizCompleted : boolean =false;
* constructor(private questionService : QuestionService)
* {
* }
* ngOnInit(): void {
* this.name =localStorage.getItem("name")!;
* this.getAllQuestions();
* this.startCounter();
* }
* getAllQuestions()
* {
* this.questionService.getQuestionJson()
* .subscribe(res=>{this.questionList=res.questions;
* })
* }
* nextQuestion()
* {
* this.currentQuestion++;
* }
* previousQuestion()
* {
* this.currentQuestion--;
* }
* answer(currentQno :number, option:any)
* {
* if(currentQno==this.questionList.length)
* {
* this.isQuizCompleted= true;
* this.stopCounter();
* }
* if(option.correct)
* {
* this.points+=10;
* this.correctAnswer++;
* setTimeout(()=>{
* this.currentQuestion++;
* this.resetCounter();
* this.getProgressPerchant();
* },1000);
* }
* else
* {
* setTimeout(()=>{
* this.currentQuestion++;
* this.incorrectAnswer++;
* this.resetCounter();
* this.getProgressPerchant();
* },1000);
* this.points-=10;
* }
* }
* startCounter()
* {
* this.interval$ =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";
* }
* getProgressPerchant()
* {
* this.progress =((this.currentQuestion/this.questionList.length)\*100).toString();
* return this.progress;
* }
* }
* **Welcome Component:**
* Welcome Component.html:
* <div class="container bg-light py-5">
* <h1 class="display-5 fw-bold">Welocme to Quiz App</h1>
* <p class="col-md-8 fs-4">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">
* </div>
* <button class="btn btn-primary btn-lg"  routerLink="/question" (click)="startQuiz()" >Start the Quiz!!</button>
* </div>
* Welcome Component.ts:
* import { Component,OnInit, ViewChild, ElementRef } from '@angular/core';
* @Component({
* selector: 'app-welcome',
* templateUrl: './welcome.component.html',
* styleUrls: ['./welcome.component.css']
* })
* export class WelcomeComponent implements OnInit{
* @ViewChild('name') nameKey!:ElementRef;
* constructor()
* {
* }
* ngOnInit(): void {
* }
* startQuiz()
* {
* localStorage.setItem("name",this.nameKey.nativeElement.value);
* }
* }
* **Service**:
* 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");
* }
* }
* **Assest:**
* Question.JSON:
* {
* "questions": [{
* "questionText": "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": "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": "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": "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": "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": "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": "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": "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": "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"
* }
* ]
* }