

Online Test Application

DESCRIPTION

The Online Test Application system creates an application that enables users to provide online tests, review them, and display the results.

System Details

This system contains three main modules: Quiz, Review, and Result. The quiz section of the online test application accepts the questions in JSON format. The JSON file can be easily shared from the server in the pre-defined format. The application renders the test at the client-side.

The “Review and display result” section allows users to declare the results immediately. You can simply call another JSON with the answers in it and evaluate and display the results immediately.

Developer Details: P Praveen Kumar

Ppraveen62.ppk@gmail.com

9502646206

Code

Questions.json and answers.json are files where u can give questions and answers in json format.

Index.html

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>QuizProj</title>
  <base href="/">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="favicon.ico">

  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min.css"
integrity="sha384-x0OlHFLEh07PJGoPkLv1IbcEPTNtaed2xpHsD9ESMhqIYd0nLMwNLD69Npy4HI+N"
crossorigin="anonymous">
  <script src="https://cdn.jsdelivr.net/npm/jquery@3.5.1/dist/jquery.slim.min.js"
integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
crossorigin="anonymous"></script>
```

```
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-Fy6S3B9q64WdZWQUiU+q4/2Lc9npb8tCaSX9FK7E8HnRr0Jz8D60P9d05Vg3Q9ct"
crossorigin="anonymous"></script>

</head>

<body>

  <div class="id">
    <app-root></app-root>
  </div>

</body>
</html>
```

Styles.css

```
html {
  height: 100%;
}

body {
  height: 100%;
  margin: 0;
  font-family: Arial, Helvetica, sans-serif;
  background-color:#0a0845;
}

app-root{
  margin: 10%;
  width: 80%;
  height: 80%;
  display: grid;
  align-items: center;
  padding: 50px;
  background-color: #cecdd0;
  border-radius: 20px;
}
```

questions.json

```
[
  {
    "questionNo":1,
```

```

    "questionDesc": "Which country hosted the 2018 FIFA World Cup?",
    "choice1": "Russia",
    "choice2": "Saudi Arabia",
    "choice3": "US",
    "choice4": "India"
  },
  {
    "questionNo":2,
    "questionDesc": "Who won the 2015 Formula 1 World Championship?",
    "choice1": "Nico Rosberg",
    "choice2": "Sebastian Vettel",
    "choice3": "Lewis Hamilton",
    "choice4": "Jenson Button"
  },
  {
    "questionNo":3,
    "questionDesc": "Who won the 2011 Cricket World Cup?",
    "choice1": "Pakistan",
    "choice2": "Sri Lanka",
    "choice3": "New Zealand",
    "choice4": "India"
  },
  {
    "questionNo":4,
    "questionDesc": "Who won the UEFA Champions League in 2019?",
    "choice1": "Real Madrid",
    "choice2": "Liverpool",
    "choice3": "Barcelona",
    "choice4": "Tottenham"
  },
  {
    "questionNo":5,
    "questionDesc": "Which player holds the NHL record of 2,857 points?",
    "choice1": "Sidney Crosby",
    "choice2": "Mario Lemieux",
    "choice3": "Gordie Howe",
    "choice4": "Wayne Gretzky"
  }
]

```

answers.json

```

[
  {
    "questionNo":1,
    "answer": "A"
  },

```

```
{
  "questionNo":2,
  "answer": "C"
},
{
  "questionNo":3,
  "answer": "D"
},
{
  "questionNo":4,
  "answer": "B"
},
{
  "questionNo":5,
  "answer": "D"
}
]
```

app.component.html

```
<div class="box">
  <router-outlet></router-outlet>
</div>
```

app.component.css

```
.box{
  position:center;
}
```

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 { StatQuizComponent } from './stat-quiz/stat-quiz.component';
import { HomeComponent } from './home/home.component';
import { FormsModule } from '@angular/forms';
import { FinishComponent } from './finish/finish.component';
```

```

@NgModule({
  declarations: [
    AppComponent,
    StatQuizComponent,
    HomeComponent,
    FinishComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    FormsModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }

```

app-routing-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 { StatQuizComponent } from './stat-quiz/stat-quiz.component';
import { HomeComponent } from './home/home.component';
import { FormsModule } from '@angular/forms';
import { FinishComponent } from './finish/finish.component';

@NgModule({
  declarations: [
    AppComponent,
    StatQuizComponent,
    HomeComponent,
    FinishComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    FormsModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }

```

stat-quiz.component.html

```
<div>
  <div id="quiz" class="justify-center flex-column">
    <p id="countdown">{{Minutes}}:{{Seconds}}</p>
    <div id="hud">
      <div id="hud-item">
        <p class="hud-prefix">Question</p>
        <h3 class="hud-text" id="questionNo">{{qsn[i].questionNo}}/{{qsn.length}}</h3>
      </div>
    </div>
    <p id="question">{{qsn[i].questionDesc}}</p>

    <div class="choicebox">
      <input type="radio" class="radioButton" id="a" name="choice"
[ngModel]="choiceRes" (click)="RadioHandler($event)" value="A">
      <label for="A" class="choice-text">{{qsn[i].choice1}}</label>
    </div>
    <div class="choicebox">
      <input type="radio" class="radioButton" id="b" name="choice"
[ngModel]="choiceRes" (click)="RadioHandler($event)" value="B">
      <label for="B" class="choice-text">{{qsn[i].choice2}}</label>
    </div>
    <div class="choicebox">
      <input type="radio" class="radioButton" id="c" name="choice"
[ngModel]="choiceRes" (click)="RadioHandler($event)" value="C">
      <label for="C" class="choice-text">{{qsn[i].choice3}}</label>
    </div>
    <div class="choicebox">
      <input type="radio" class="radioButton" id="d" name="choice"
[ngModel]="choiceRes" (click)="RadioHandler($event)" value="D">
      <label for="D" class="choice-text">{{qsn[i].choice4}}</label>
    </div>

  </div>

</div>
<div class="finishBox">
  <div class="f1box">
    <button type="button" disabled class="click" id="submitOption" (click)="submit()"
routerLink="/finish">Submit</button>
  </div>
  <div class="f2box">
    <button type="button" disabled class="click" id="previousOption"
(click)="previousQuestion()">Prev</button>
    <button type="button" class="click" id="nextOption"
(click)="nextQuestion()">Next</button>
  </div>
</div>
```

```
</div>  
</div>
```

stat-quiz.component.css

```
.container{  
  max-width: fit-content;  
  margin: 0 auto;  
}  
  
.choicebox{  
  display: flex;  
  align-items:baseline;  
  background-color:#030313;  
  border-radius: 15px;  
  color: whitesmoke;  
  margin-bottom: 1rem;  
  padding-left: 10px;  
  padding-top: 10px;  
  padding-right: 10px;  
  font-weight:normal;  
}  
  
.choicebox:hover {  
  color:dodgerblue;  
  box-shadow: 0 0.4rem 1rem 0 dodgerblue;  
  transform: translateY(-0.1rem);  
  transition: transform 150ms;  
}  
  
.radioButton{  
  font-family: system-ui, sans-serif;  
  font-size: 2rem;  
  font-weight: bold;  
  line-height: 1.1;  
  display: grid;  
  grid-template-columns: 1em auto;  
  cursor: pointer;  
}  
  
.choice-text{  
  padding-left: 10px;
```

```
background-color: #030313;
}

#hud {
  display: inline-block;
  justify-content: space-between;
  text-transform: uppercase;
}

.hud-prefix {
  text-align: center;
  font-size: 1.5rem;
}

.hud-text {
  color: firebrick;
  font-weight: bolder;
  text-align: center;
  text-justify:auto
}

#question{
  font-size: 1.5rem;
  font-weight: normal;
  text-align:left;
  width:100%;
}

.click{
  border-radius: 10px;
  background-color: rgb(3, 3, 46);
  border-color:rgb(3, 3, 46);
  color: white;
  margin-right: 10px;
}

.click:hover{
  background-color:seagreen;
  border-color:seagreen;
  box-shadow: 0 0.4rem 1rem 0 yellowgreen;
  transform: translateY(-0.1rem);
}
```



```

    transition: transform 150ms;
  }
  .click[disabled]:hover{
    background-color:firebrick;
    border-color:firebrick;
    box-shadow: 0 0.4rem 1rem 0 tomato;
    transform: translateY(-0.1rem);
    transition: transform 150ms;
    cursor: not-allowed;
  }

  .flbox{
    float: right;
  }

  #countdown{
    text-align: right;
    padding-left: 10px;
    padding-right: 10px;
    padding: 2px;
    color:black;
    font-weight: 600;
    font-size: xx-large;
  }

```

stat-quiz.component.ts

```

import { FinishComponent } from '../finish/finish.component';
import { Component, EventEmitter, OnInit, Output } from '@angular/core';
import questionData from '../assets/questions.json';
import answerData from '../assets/answers.json';
import { interval, Subscription, takeUntil, timeout, timer } from 'rxjs';
import { Router, RouterLink } from '@angular/router';

interface question {
  questionNo: Number,
  questionDesc: String,
  choice1: String,

```

```

    choice2: String,
    choice3: String,
    choice4: String,
}

interface answer {
    questionNo: Number,
    answer: String,
}

@Component({
    selector: 'app-stat-quiz',
    templateUrl: './stat-quiz.component.html',
    styleUrls: ['./stat-quiz.component.css']
})

export class StatQuizComponent implements OnInit {
    qsn: question[] = questionData;
    ans: answer[] = answerData;

    userAnswerlist: String[] = [];
    choiceRes: String = "";
    i: number = 0;

    public static QsnTotal: number = questionData.length;
    public static QsnAnswered: number = 0;
    public static RightAnswered: number = 0;

    //countdown

    startingMinutes: number = 2;
    time: number = this.startingMinutes * 60;
    countdownEl = <HTMLInputElement>document.getElementById("countdown");
    Minutes: any = 0;
    Seconds: any = 0;

    subs: Subscription = new Subscription;

    ///

    constructor(private router: Router) { }

    ngOnInit(): void {

        //

```

```

const obs = interval(1000);
this.subs = obs.subscribe((d) => {
  const minutes = Math.floor(this.time / 60);
  let seconds = this.time % 60;
  this.Minutes = minutes
  this.Seconds = seconds
  this.time--;

  if ((this.startingMinutes * 60) == d) {
    this.subs.unsubscribe();
    this.submit();
    this.router.navigate(["/finish"])
  }
})

}

reset() {
  let e1 = <HTMLInputElement>document.getElementById('a');
  e1.checked = false
  let e2 = <HTMLInputElement>document.getElementById('b');
  e2.checked = false
  let e3 = <HTMLInputElement>document.getElementById('c');
  e3.checked = false
  let e4 = <HTMLInputElement>document.getElementById('d');
  e4.checked = false
}

submit(): void {

  for (let i = 0; i < this.qsn.length; i++) {
    if (this.ans[i].answer == this.userAnswerlist[i]) {
      StatQuizComponent.RightAnswered++;
    }
    if (this.userAnswerlist[i] == "A" || this.userAnswerlist[i] == "B" ||
this.userAnswerlist[i] == "C" || this.userAnswerlist[i] == "D") {
      StatQuizComponent.QsnAnswered++;
    }
  }

}

}

previousQuestion(): void {

```

```

    if (this.i > 0 || this.i < this.qsn.length - 1) {

        this.i--;
    }
    if (this.i !== this.qsn.length - 1) {
        let element = <HTMLInputElement>document.getElementById('nextOption');
        element.disabled = false;
    }
    if (this.i === 0) {
        let element = <HTMLInputElement>document.getElementById('previousOption');
        element.disabled = true;
    }
    this.reset();
}

nextQuestion(): void {

    if (this.i === 0 || this.i < this.qsn.length - 1) {
        console.log(this.i);
        this.i++;
        let element = <HTMLInputElement>document.getElementById('previousOption');
        element.disabled = false;
    }
    if (this.i === this.qsn.length - 1) {
        let element = <HTMLInputElement>document.getElementById('nextOption');
        element.disabled = true;
    }
    if (this.i === 0) {
        let element = <HTMLInputElement>document.getElementById('previousOption');
        element.disabled = true;
    }
    this.reset();

    if (this.i === this.qsn.length - 1) {
        let element = <HTMLInputElement>document.getElementById('submitOption');
        element.disabled = false;
    }
}

RadioHandler(e: any) {
    let answer = e.target.value;
    this.userAnswerlist[this.i] = answer;
    console.log(this.userAnswerlist);
}

```

```
}
```

home.component.html

```
<div class="startbutton">
  <h1>Are you ready to Start...</h1><br>

</div>
<div class="box">
  <h2 style="text-align:center">Instructions</h2>
  <p class="para">
    1. Read carefully and answer the questions.<br>
    2. There will be a submit Option to complete the Quiz.<br>
    3. Submit option will be enabled on last Question of Quiz.<br>
    4. The total score for the quiz is based on your responses to all
questions.<br>
    5. It auto submit on completion of time.<br>
    6.On selecting options it auto saves the given answers.<br>
  </p>
</div>

<div class="startbutton">
  <button type="button" class="click" id="submitOption"
routerLink="/start">Start</button>
</div>
```

home.component.css

```
.startbutton{
  text-align: center;
  padding: 10px;
}

.click{
  border-radius: 10px;
  background-color: rgb(3, 3, 46);
  border-color:rgb(3, 3, 46);
  color: white;
  padding-left:20px;
  padding-right:20px;
  padding-top:5px;
  padding-bottom:5px;
  font-weight: bold;
```

```

}

.para{
  font-size:1rem;
  font-weight: bold;
  padding: 20px;
}

.box{
  display:grid;
  background-color:mintcream;
  border-radius: 20px;
}

h2{
  font-weight: bolder;
  color: crimson;
}

h1{
  color: midnightblue;
  font-weight: 900;
  font-style: italic;
}

```

home.component.ts

```

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

@Component({
  selector: 'app-home',
  templateUrl: './home.component.html',
  styleUrls: ['./home.component.css']
})
export class HomeComponent implements OnInit {

  constructor() { }

  ngOnInit(): void {
  }

}

```

Finish.component.html

```

<h2>Result</h2><br>
<p class="para">

```

```

    Total no of questions : <b>{{questionsTotal}}</b><br>
    Total no of questions attempted: <b>{{questionsAnswered}}</b><br>
    Total no of questions answered right : <b>{{rightAnswered}}</b><br>
    Total no of questions answered wrong : <b>{{wrongAnswered}}</b>
</p>
<h3>Score : <b class="score">{{rightAnswered*1}}</b></h3>
<br>
<div class="finish">
    <button type="button" class="click" id="submitOption"
(click)="submit()">Done</button>
</div>

```

finish.component.css

```

.finish{text-align: center;}

.para{
    font-size:1.5rem;
    font-weight: normal;
}
h2{
    text-align: center;
    font-weight:bolder;
    color: midnightblue;
    background-color:cornflowerblue;
    border-radius: 12px;
}

b{
    color: red;
}

.click{
    border-radius: 10px;
    background-color: rgb(3, 3, 46);
    border-color:rgb(3, 3, 46);
    color: white;
    padding-left:20px;
    padding-right:20px;
    padding-top:5px;
    padding-bottom:5px;
    font-weight: bold;
}

.score{
    color: midnightblue;
    background-color: springgreen;
}

```

```
padding: 2px;
padding-left: 20px;
padding-right: 20px;
}
```

finish.component.ts

```
import { Component, OnInit, Input } from '@angular/core';
import { StatQuizComponent } from '../stat-quiz/stat-quiz.component';
@Component({
  selector: 'app-finish',
  templateUrl: './finish.component.html',
  styleUrls: ['./finish.component.css']
})
export class FinishComponent implements OnInit {

  constructor() { }

  ngOnInit(): void {

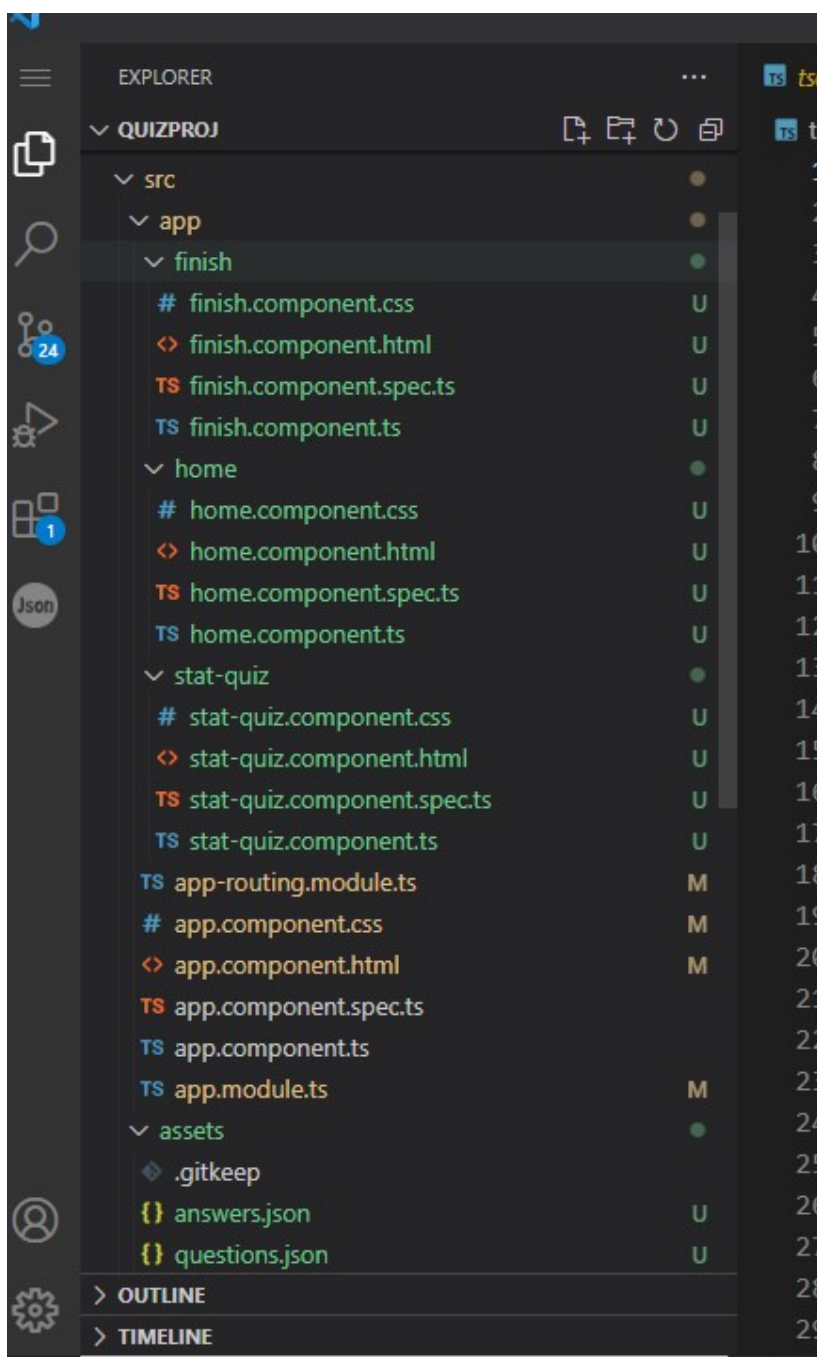
  }

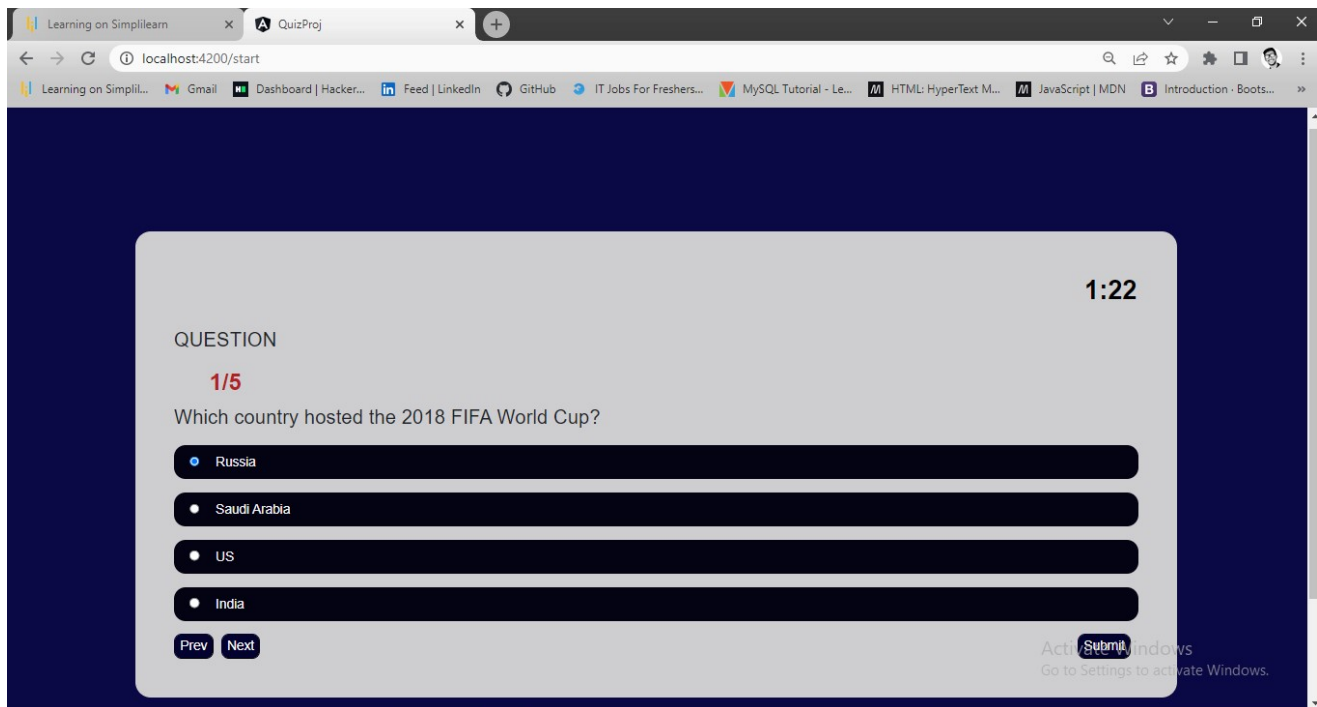
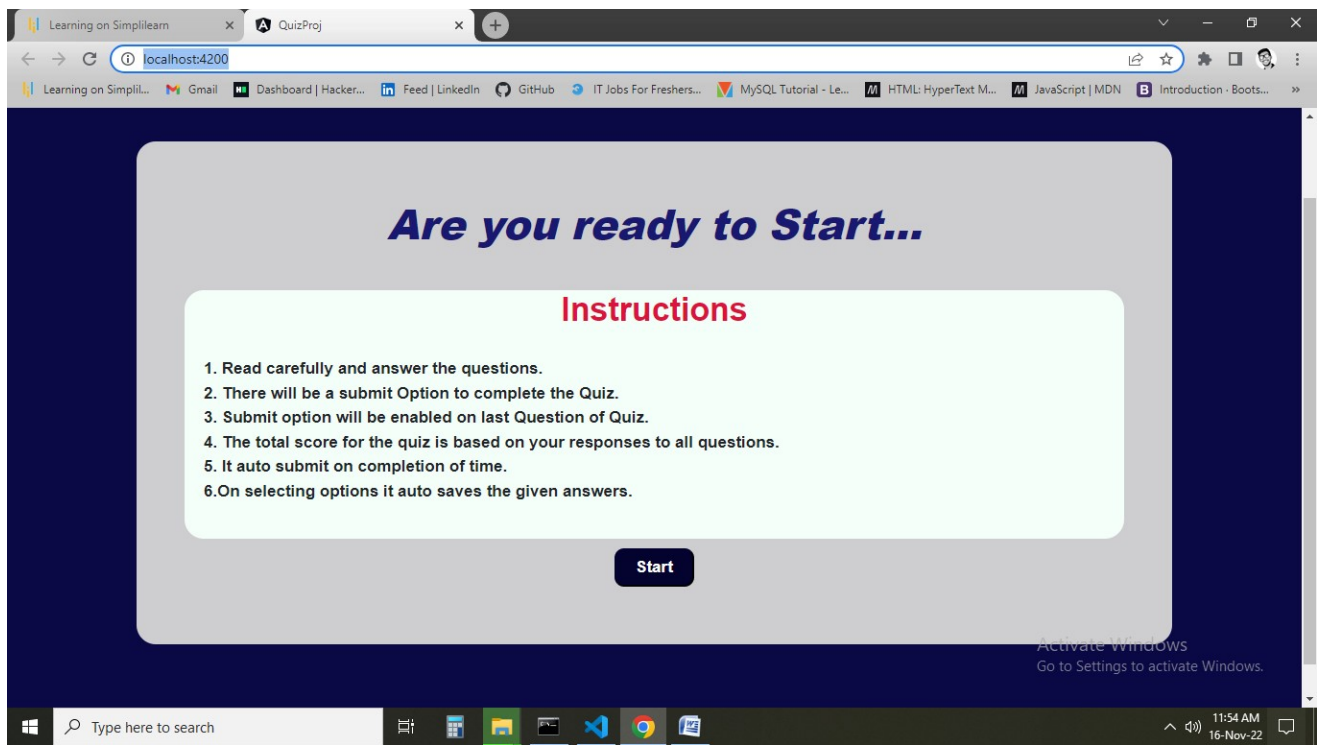
  questionsTotal:number= StatQuizComponent.QsnTotal;
  questionsAnswered:number=StatQuizComponent.QsnAnswered;
  rightAnswered: number =StatQuizComponent.RightAnswered;
  wrongAnswered: number=StatQuizComponent.QsnAnswered-StatQuizComponent.RightAnswered;

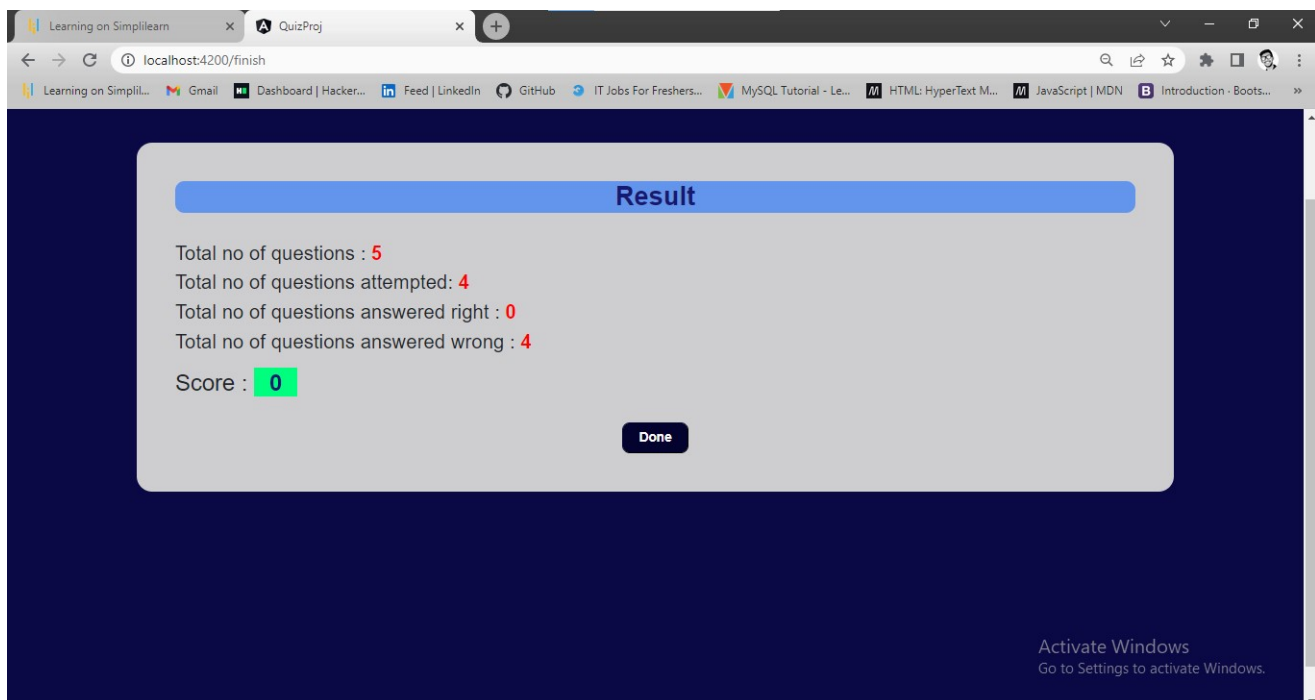
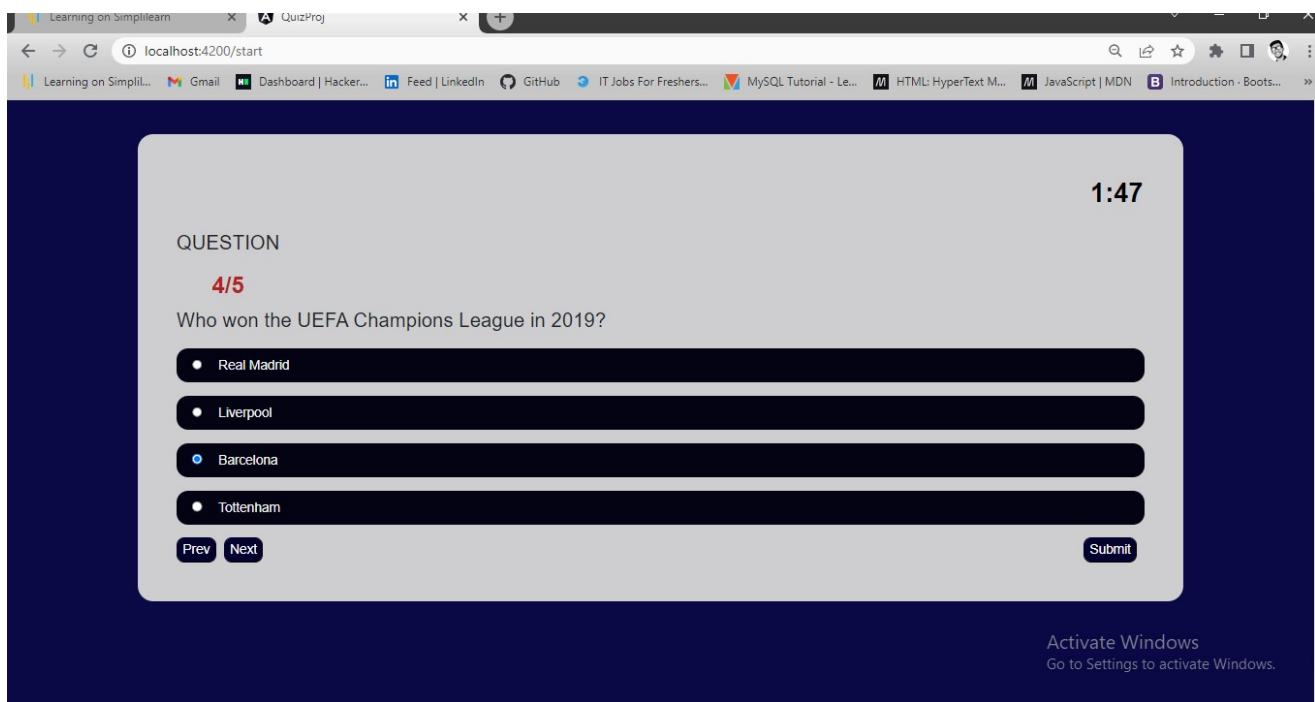
  submit(){
    window.location.replace("");
  }

}
```

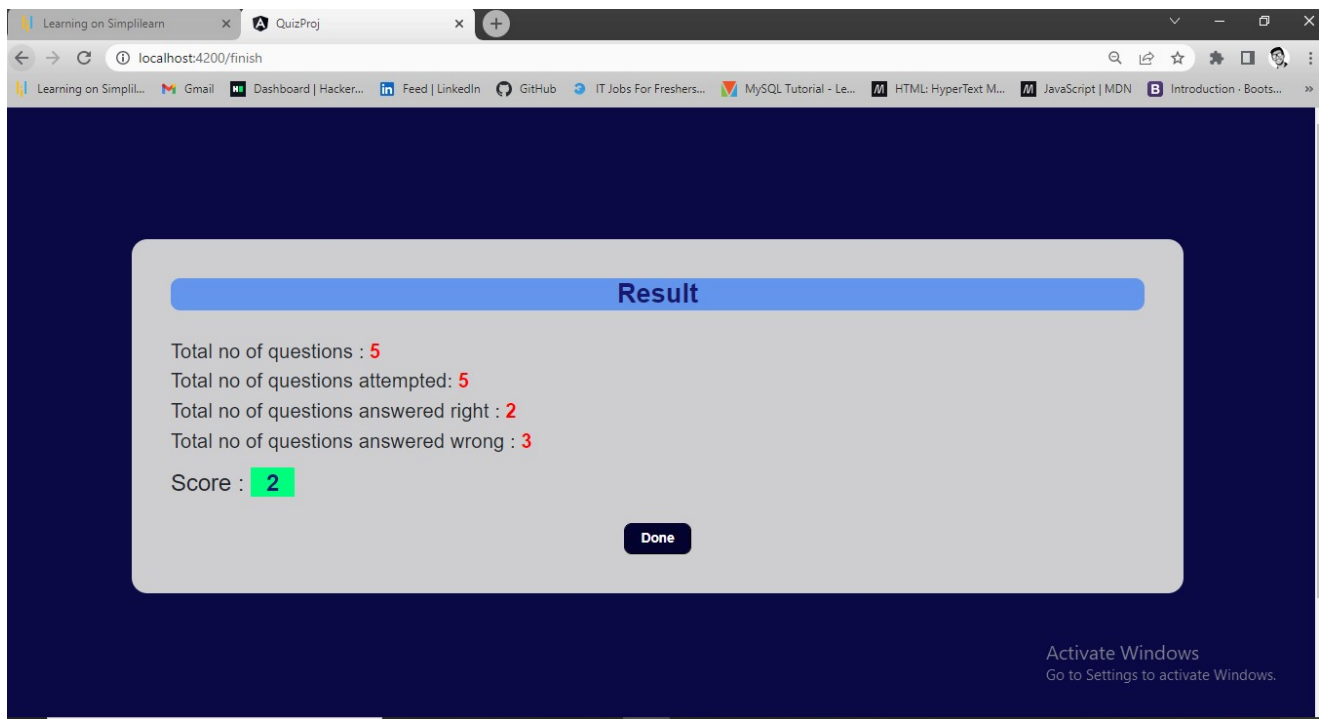
Path structures







Another attempt



GitHub Link

<https://github.com/ppraveen62/QuizProj-final-angular.git>

```
MINGW64/e/simplilearn/github/phase 4/Quiz-final-proj
hint: See "git help submodule" for more information.

PPK@PPK MINGW64 /e/simplilearn/github/phase 4/Quiz-final-proj (master)
$ git commit -m "files are added"
[master (root-commit) 4061c9b] files are added
10 files changed, 8 insertions(+)
create mode 100644 1.png
create mode 100644 2.png
create mode 100644 3.png
create mode 100644 4.png
create mode 100644 5.png
create mode 100644 6.png
create mode 100644 Description.txt
create mode 100644 Document.docx
create mode 160000 QuizProj
create mode 100644 ~$cument.docx

PPK@PPK MINGW64 /e/simplilearn/github/phase 4/Quiz-final-proj (master)
$ git push
fatal: No configured push destination.
Either specify the URL from the command-line or configure a remote repository using

    git remote add <name> <url>

and then push using the remote name

    git push <name>

PPK@PPK MINGW64 /e/simplilearn/github/phase 4/Quiz-final-proj (master)
$ git branch -M main
PPK@PPK MINGW64 /e/simplilearn/github/phase 4/Quiz-final-proj (main)
$ git remote add origin https://github.com/ppraveen62/QuizProj-final-angular.git
PPK@PPK MINGW64 /e/simplilearn/github/phase 4/Quiz-final-proj (main)
$ git push -u origin main
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 4 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (11/11), 419.86 KiB | 6.17 MiB/s, done.
Total 11 (delta 6), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (6/6), done.
To https://github.com/ppraveen62/QuizProj-final-angular.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

PPK@PPK MINGW64 /e/simplilearn/github/phase 4/Quiz-final-proj (main)
$
```