Source Code

App.Ts

import { browser, logging } from 'protractor';

import { AppPage } from './app.po';

describe('workspace-project App', () => {

let page: AppPage;

beforeEach(() => {

page = new AppPage();

});

it('should display welcome message', async () => {

await page.navigateTo();

expect(await page.getTitleText()).toEqual('online-test-application app is running!');

});

afterEach(async () => {

// Assert that there are no errors emitted from the browser

const logs = await browser.manage().logs().get(logging.Type.BROWSER);

expect(logs).not.toContain(jasmine.objectContaining({

level: logging.Level.SEVERE,

} as logging.Entry));

});

});

po ts

import { browser, by, element } from 'protractor';

export class AppPage {

async navigateTo(): Promise<unknown> {

return browser.get(browser.baseUrl);

}

async getTitleText(): Promise<string> {

return element(by.css('app-root .content span')).getText();

}

}

Apps

list.html

<div class="container mt-5">

<div class="row">

<div \*ngFor="let tempQuiz of quizes" class="col-md-4">

<div class="card">

<img

src="{{ tempQuiz.imageUrl }}"

class="card-img-top"

alt="Quiz Image"

/>

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

<h5 class="card-title">

{{ tempQuiz.name }}

</h5>

<hr />

<p class="card-text">

{{ tempQuiz.description }}

</p>

<a

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

class="btn btn-outline-primary"

>Give {{ tempQuiz.name }} Quiz</a

>

</div>

</div>

</div>

</div>

</div>

List ts

import { ComponentFixture, TestBed } from '@angular/core/testing';

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

describe('QuizListComponent', () => {

let component: QuizListComponent;

let fixture: ComponentFixture<QuizListComponent>;

beforeEach(async () => {

await TestBed.configureTestingModule({

declarations: [ QuizListComponent ]

})

.compileComponents();

});

beforeEach(() => {

fixture = TestBed.createComponent(QuizListComponent);

component = fixture.componentInstance;

fixture.detectChanges();

});

it('should create', () => {

expect(component).toBeTruthy();

});

});

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();

}

}

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

<a class="navbar-brand ml-2 fw-bolder" routerLink="/quiz">Online 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-dark text-center text-lg-start mt-5">

<!-- Copyright -->

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

© 2021 Copyright:

<a class="text-light fw-bold" routerLink="/quiz">Online Quiz Application</a>

</div>

<!-- Copyright -->

</footer>

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 {}

jason File

// @ts-check

// Protractor configuration file, see link for more information

// <https://github.com/angular/protractor/blob/master/lib/config.ts>

const { SpecReporter, StacktraceOption } = require('jasmine-spec-reporter');

/\*\*

\* @type { import("protractor").Config }

\*/

exports.config = {

allScriptsTimeout: 11000,

specs: [

'./src/\*\*/\*.e2e-spec.ts'

],

capabilities: {

browserName: 'chrome'

},

directConnect: true,

SELENIUM\_PROMISE\_MANAGER: false,

baseUrl: '<http://localhost:4200/>',

framework: 'jasmine',

jasmineNodeOpts: {

showColors: true,

defaultTimeoutInterval: 30000,

print: function() {}

},

onPrepare() {

require('ts-node').register({

project: require('path').join(\_\_dirname, './tsconfig.json')

});

jasmine.getEnv().addReporter(new SpecReporter({

spec: {

displayStacktrace: StacktraceOption.PRETTY

}

}));

}

};

Angular File

{

"id": 1,

"name": "Asp.Net Quiz",

"description": "Asp.Net Quiz (contains webform, mvc, web API, etc.)",

"questions": [

{

"id": 1010,

"name": "ASP.NET webform separates the HTML output from program logic using a feature named as",

"questionTypeId": 1,

"options": [

{

"id": 1055,

"questionId": 1010,

"name": "Exception",

"isAnswer": false

},

{

"id": 1056,

"questionId": 1010,

"name": "Code-behind",

"isAnswer": true

},

{

"id": 1057,

"questionId": 1010,

"name": "Code-front",

"isAnswer": false

},

{

"id": 1058,

"questionId": 1010,

"name": "None of the above",

"isAnswer": false

}

],

"questionType": {

"id": 1,

"name": "Multiple Choice",

"isActive": true

}

},

{

"id": 1011,

"name": "The feature in ASP.NET 2.0 that is used to fire a normal postback to a different page in the application is called",

"questionTypeId": 1,

"options": [

{

"id": 1055,

"questionId": 1010,

"name": "Theme",

"isAnswer": false

},

{

"id": 1057,

"questionId": 1010,

"name": "Code-front",

"isAnswer": false

},

{

"id": 1056,

"questionId": 1010,

"name": "Cross Page Posting",

"isAnswer": true

},

{

"id": 1058,

"questionId": 1010,

"name": "None of the above",

"isAnswer": false

}

],

"questionType": {

"id": 1,

"name": "Multiple Choice",

"isActive": true

}

},

{

"id": 1012,

"name": "What class does the ASP.NET Web Form class inherit from by default?",

"questionTypeId": 1,

"options": [

{

"id": 1055,

"questionId": 1010,

"name": "System.Web.UI.Page",

"isAnswer": true

},

{

"id": 1057,

"questionId": 1010,

"name": "System.Web.UI.Form",

"isAnswer": false

},

{

"id": 1056,

"questionId": 1010,

"name": "System.Web.GUI.Page",

"isAnswer": false

},

{

"id": 1058,

"questionId": 1010,

"name": "System.Web.Form",

"isAnswer": false

}

],

"questionType": {

"id": 1,

"name": "Multiple Choice",

"isActive": true

}

},

{

"id": 1013,

"name": "What does MVC stand for?",

"questionTypeId": 1,

"options": [

{

"id": 1055,

"questionId": 1010,

"name": "Model View Controller",

"isAnswer": true

},

{

"id": 1057,

"questionId": 1010,

"name": "Maximum Virtual Control",

"isAnswer": false

},

{

"id": 1056,

"questionId": 1010,

"name": "Microsoft Visual Core",

"isAnswer": false

},

{

"id": 1058,

"questionId": 1010,

"name": "None of the above",

"isAnswer": false

}

],

"questionType": {

"id": 1,

"name": "Multiple Choice",

"isActive": true

}

},

{

"id": 1014,

"name": "Which of the following does NOT require type casting?",

"questionTypeId": 1,

"options": [

{

"id": 1055,

"questionId": 1010,

"name": "Session",

"isAnswer": false

},

{

"id": 1057,

"questionId": 1010,

"name": "TempData",

"isAnswer": false

},

{

"id": 1056,

"questionId": 1010,

"name": "ViewData",

"isAnswer": false

},

{

"id": 1058,

"questionId": 1010,

"name": "ViewBag",

"isAnswer": true

}

],

"questionType": {

"id": 1,

"name": "Multiple Choice",

"isActive": true

}

},

{

"id": 1015,

"name": "Which is the correct order of Page life-cycle in asp.net webform?",

"questionTypeId": 1,

"options": [

{

"id": 1055,

"questionId": 1010,

"name": "Init, PreRender, Load",

"isAnswer": false

},

{

"id": 1057,

"questionId": 1010,

"name": "Load, PreRender, Init",

"isAnswer": false

},

{

"id": 1056,

"questionId": 1010,

"name": "Init, Load, PreRender",

"isAnswer": true

},

{

"id": 1058,

"questionId": 1010,

"name": "None of the above",

"isAnswer": false

}

],

"questionType": {

"id": 1,

"name": "Multiple Choice",

"isActive": true

}

},

{

"id": 1016,

"name": "Which of these data source controls do not implement caching?",

"questionTypeId": 1,

"options": [

{

"id": 1055,

"questionId": 1010,

"name": "ObjectDataSource",

"isAnswer": false

},

{

"id": 1056,

"questionId": 1010,

"name": "LinqDataSource",

"isAnswer": true

},

{

"id": 1057,

"questionId": 1010,

"name": "SqlDataSource",

"isAnswer": false

},

{

"id": 1058,

"questionId": 1010,

"name": "XmlDataSource",

"isAnswer": false

}

],

"questionType": {

"id": 1,

"name": "Multiple Choice",

"isActive": true

}

},

{

"id": 1017,

"name": "Which tag asp:Label control by default renders to?",

"questionTypeId": 1,

"options": [

{

"id": 1055,

"questionId": 1010,

"name": "div",

"isAnswer": false

},

{

"id": 1056,

"questionId": 1010,

"name": "span",

"isAnswer": true

},

{

"id": 1057,

"questionId": 1010,

"name": "body",

"isAnswer": false

},

{

"id": 1058,

"questionId": 1010,

"name": "label",

"isAnswer": false

}

],

"questionType": {

"id": 1,

"name": "Multiple Choice",

"isActive": true

}

},

{

"id": 1018,

"name": "Which method do you use to explicitly kill a user's session?",

"questionTypeId": 1,

"options": [

{

"id": 1055,

"questionId": 1010,

"name": "Session.Terminate()",

"isAnswer": false

},

{

"id": 1057,

"questionId": 1010,

"name": "Session.TimeOut()",

"isAnswer": false

},

{

"id": 1056,

"questionId": 1010,

"name": "Session.Abondon()",

"isAnswer": true

},

{

"id": 1058,

"questionId": 1010,

"name": "Session.Kill()",

"isAnswer": false

}

],

"questionType": {

"id": 1,

"name": "Multiple Choice",

"isActive": true

}

},

{

"id": 1019,

"name": "Which of the following object is ideal for keeping data alive for a single request?",

"questionTypeId": 1,

"options": [

{

"id": 1055,

"questionId": 1010,

"name": "HttpContext",

"isAnswer": true

},

{

"id": 1056,

"questionId": 1010,

"name": "Session",

"isAnswer": false

},

{

"id": 1057,

"questionId": 1010,

"name": "Cookies",

"isAnswer": false

},

{

"id": 1058,

"questionId": 1010,

"name": "SqlServer",

"isAnswer": false

}

],

"questionType": {

"id": 1,

"name": "Multiple Choice",

"isActive": true

}

}

]

}

{

"id": 2,

"name": "C# and .Net Framework",

"description": "C# and .Net Quiz (contains C#, .Net Framework, Linq, etc.)",

"config": {

"shuffleQuestions": true,

"showPager": false,

"allowBack": true,

"autoMove": true

},

"questions": [

{

"id": 1010,

"name": "Which of the following assemblies can be stored in Global Assembly Cache?",

"questionTypeId": 1,

"options": [

{

"id": 1055,

"questionId": 1010,

"name": "Private Assemblies",

"isAnswer": false

},

{

"id": 1056,

"questionId": 1010,

"name": "Friend Assemblies",

"isAnswer": false

},

{

"id": 1057,

"questionId": 1010,

"name": "Public Assemblies",

"isAnswer": false

},

{

"id": 1058,

"questionId": 1010,

"name": "Shared Assemblies",

"isAnswer": true

}

],

"questionType": {

"id": 1,

"name": "Multiple Choice",

"isActive": true

}

},

{

"id": 1011,

"name": "Which of the following .NET components can be used to remove unused references from the managed heap?",

"questionTypeId": 1,

"options": [

{

"id": 1055,

"questionId": 1010,

"name": "Language Infrastructure",

"isAnswer": false

},

{

"id": 1056,

"questionId": 1010,

"name": "CLR",

"isAnswer": false

},

{

"id": 1057,

"questionId": 1010,

"name": "Garbage Collector",

"isAnswer": true

},

{

"id": 1058,

"questionId": 1010,

"name": "Class Loader",

"isAnswer": false

},

{

"id": 1058,

"questionId": 1010,

"name": "CTS",

"isAnswer": false

}

],

"questionType": {

"id": 1,

"name": "Multiple Choice",

"isActive": true

}

},

{

"id": 1012,

"name": "Which of the following utilities can be used to compile managed assemblies into processor-specific native code?",

"questionTypeId": 1,

"options": [

{

"id": 1055,

"questionId": 1010,

"name": "gacutil",

"isAnswer": false

},

{

"id": 1057,

"questionId": 1010,

"name": "ngen",

"isAnswer": true

},

{

"id": 1056,

"questionId": 1010,

"name": "dumpbin",

"isAnswer": false

},

{

"id": 1058,

"questionId": 1010,

"name": "ildasm",

"isAnswer": false

}

],

"questionType": {

"id": 1,

"name": "Multiple Choice",

"isActive": true

}

},

{

"id": 1013,

"name": "Which of the following is NOT an Arithmetic operator in C#.NET?",

"questionTypeId": 1,

"options": [

{

"id": 1055,

"questionId": 1010,

"name": "\*\* (Double Star)",

"isAnswer": true

},

{

"id": 1057,

"questionId": 1010,

"name": "+ (Plus)",

"isAnswer": false

},

{

"id": 1056,

"questionId": 1010,

"name": "/ (Divide)",

"isAnswer": false

},

{

"id": 1058,

"questionId": 1010,

"name": "% (Modulo)",

"isAnswer": false

}

],

"questionType": {

"id": 1,

"name": "Multiple Choice",

"isActive": true

}

},

{

"id": 1014,

"name": "Which of the following statements is correct about an interface used in C#.NET?",

"questionTypeId": 1,

"options": [

{

"id": 1055,

"questionId": 1010,

"name": "If a class implements an interface partially, then it should be an abstract class.",

"isAnswer": true

},

{

"id": 1057,

"questionId": 1010,

"name": "A class cannot implement an interface partially.",

"isAnswer": false

},

{

"id": 1056,

"questionId": 1010,

"name": "An interface can contain static methods.",

"isAnswer": false

},

{

"id": 1058,

"questionId": 1010,

"name": "An interface can contain static data.",

"isAnswer": false

}

],

"questionType": {

"id": 1,

"name": "Multiple Choice",

"isActive": true

}

},

{

"id": 1015,

"name": "What does the term <strong>immutable</strong> means in term of string objects?",

"questionTypeId": 1,

"options": [

{

"id": 1055,

"questionId": 1010,

"name": "We can modify characters included in the string",

"isAnswer": false

},

{

"id": 1057,

"questionId": 1010,

"name": "We cannot modify characters contained in the string",

"isAnswer": true

},

{

"id": 1056,

"questionId": 1010,

"name": "We cannot perform various operation of comparison, inserting, appending etc",

"isAnswer": false

},

{

"id": 1058,

"questionId": 1010,

"name": "None of the above",

"isAnswer": false

}

],

"questionType": {

"id": 1,

"name": "Multiple Choice",

"isActive": true

}

},

{

"id": 1016,

"name": "Which of the following is NOT a .NET Exception class?",

"questionTypeId": 1,

"options": [

{

"id": 1055,

"questionId": 1010,

"name": "Exception",

"isAnswer": false

},

{

"id": 1056,

"questionId": 1010,

"name": "StackMemoryException",

"isAnswer": true

},

{

"id": 1057,

"questionId": 1010,

"name": "DivideByZeroException",

"isAnswer": false

},

{

"id": 1058,

"questionId": 1010,

"name": "InvalidOperationException",

"isAnswer": false

}

],

"questionType": {

"id": 1,

"name": "Multiple Choice",

"isActive": true

}

},

{

"id": 1017,

"name": "In C#.NET if we do not catch the exception thrown at runtime then which of the following will catch it?",

"questionTypeId": 1,

"options": [

{

"id": 1055,

"questionId": 1010,

"name": "Compiler",

"isAnswer": false

},

{

"id": 1056,

"questionId": 1010,

"name": "CLR",

"isAnswer": true

},

{

"id": 1057,

"questionId": 1010,

"name": "Linker",

"isAnswer": false

},

{

"id": 1058,

"questionId": 1010,

"name": "Operating system",

"isAnswer": false

}

],

"questionType": {

"id": 1,

"name": "Multiple Choice",

"isActive": true

}

},

{

"id": 1018,

"name": "Which of the following statements are correct about delegates?",

"questionTypeId": 1,

"options": [

{

"id": 1055,

"questionId": 1010,

"name": "Delegates cannot be used to call a static method of a class.",

"isAnswer": false

},

{

"id": 1057,

"questionId": 1010,

"name": "Delegates cannot be used to call procedures that receive variable number of arguments.",

"isAnswer": true

},

{

"id": 1056,

"questionId": 1010,

"name": "If signatures of two methods are same they can be called through the same delegate object.",

"isAnswer": false

},

{

"id": 1058,

"questionId": 1010,

"name": "Delegates cannot be used to call an instance function. Delegates cannot be used to call an instance subroutine.",

"isAnswer": false

}

],

"questionType": {

"id": 1,

"name": "Multiple Choice",

"isActive": true

}

},

{

"id": 1019,

"name": "Which of the following does NOT represent Integer?",

"questionTypeId": 1,

"options": [

{

"id": 1055,

"questionId": 1010,

"name": "Char",

"isAnswer": true

},

{

"id": 1056,

"questionId": 1010,

"name": "Byte",

"isAnswer": false

},

{

"id": 1057,

"questionId": 1010,

"name": "Short",

"isAnswer": false

},

{

"id": 1058,

"questionId": 1010,

"name": "Long",

"isAnswer": false

}

],

"questionType": {

"id": 1,

"name": "Multiple Choice",

"isActive": true

}

}

]

}

javaScript

{

"id": 1,

"name": "JavaScript Quiz",

"description": "JavaScript Quiz (Basic Multiple Choice Questions for JavaScript Developers)",

"questions": [

{

"id": 1010,

"name": "Which HTML tag do we use to put the JavaScript?",

"questionTypeId": 1,

"options": [

{

"id": 1055,

"questionId": 1010,

"name": "<javascript>",

"isAnswer": false

},

{

"id": 1056,

"questionId": 1010,

"name": "<script>",

"isAnswer": true

},

{

"id": 1057,

"questionId": 1010,

"name": "<js>",

"isAnswer": false

},

{

"id": 1058,

"questionId": 1010,

"name": "None of the above",

"isAnswer": false

}

],

"questionType": {

"id": 1,

"name": "Multiple Choice",

"isActive": true

}

},

{

"id": 1011,

"name": "Which built-in method calls a function for each element in the array?",

"questionTypeId": 1,

"options": [

{

"id": 1055,

"questionId": 1010,

"name": "while()",

"isAnswer": false

},

{

"id": 1057,

"questionId": 1010,

"name": "loop",

"isAnswer": false

},

{

"id": 1056,

"questionId": 1010,

"name": "forEach",

"isAnswer": true

},

{

"id": 1058,

"questionId": 1010,

"name": "takeUntil",

"isAnswer": false

}

],

"questionType": {

"id": 1,

"name": "Multiple Choice",

"isActive": true

}

},

{

"id": 1012,

"name": "What is the difference between let and var?",

"questionTypeId": 1,

"options": [

{

"id": 1055,

"questionId": 1010,

"name": "let has local scope",

"isAnswer": true

},

{

"id": 1057,

"questionId": 1010,

"name": "Both are same",

"isAnswer": false

},

{

"id": 1056,

"questionId": 1010,

"name": "var is new data type",

"isAnswer": false

},

{

"id": 1058,

"questionId": 1010,

"name": "let consumes more cpu and ram",

"isAnswer": false

}

],

"questionType": {

"id": 1,

"name": "Multiple Choice",

"isActive": true

}

},

{

"id": 1013,

"name": "What is TypeScript?",

"questionTypeId": 1,

"options": [

{

"id": 1055,

"questionId": 1010,

"name": "A Language based on Javascript",

"isAnswer": true

},

{

"id": 1057,

"questionId": 1010,

"name": "script that runs on browser",

"isAnswer": false

},

{

"id": 1056,

"questionId": 1010,

"name": "A DataType Collection of Javascript",

"isAnswer": false

},

{

"id": 1058,

"questionId": 1010,

"name": "None of the above",

"isAnswer": false

}

],

"questionType": {

"id": 1,

"name": "Multiple Choice",

"isActive": true

}

},

{

"id": 1014,

"name": "Which of the following is right syntex for arrow function?",

"questionTypeId": 1,

"options": [

{

"id": 1055,

"questionId": 1010,

"name": "a -> { return b; }",

"isAnswer": false

},

{

"id": 1057,

"questionId": 1010,

"name": "x <= x + y;",

"isAnswer": false

},

{

"id": 1056,

"questionId": 1010,

"name": "x <- x + 5;",

"isAnswer": false

},

{

"id": 1058,

"questionId": 1010,

"name": "x => x + 5;",

"isAnswer": true

}

],

"questionType": {

"id": 1,

"name": "Multiple Choice",

"isActive": true

}

},

{

"id": 1015,

"name": "Which new ES6 syntax helps with formatting output text - mixing variables with string literals, for example.",

"questionTypeId": 1,

"options": [

{

"id": 1055,

"questionId": 1010,

"name": "Generator Functions",

"isAnswer": false

},

{

"id": 1057,

"questionId": 1010,

"name": "Arrow Functions",

"isAnswer": false

},

{

"id": 1056,

"questionId": 1010,

"name": "Template Strings",

"isAnswer": true

},

{

"id": 1058,

"questionId": 1010,

"name": "Set Data Structure",

"isAnswer": false

}

],

"questionType": {

"id": 1,

"name": "Multiple Choice",

"isActive": true

}

},

{

"id": 1016,

"name": "Which ES6 feature helps in merging of a number of changed properties into an existing object?",

"questionTypeId": 1,

"options": [

{

"id": 1055,

"questionId": 1010,

"name": "Class syntex",

"isAnswer": false

},

{

"id": 1056,

"questionId": 1010,

"name": "Object.assign()",

"isAnswer": true

},

{

"id": 1057,

"questionId": 1010,

"name": "map data structure",

"isAnswer": false

},

{

"id": 1058,

"questionId": 1010,

"name": "Array.includes(obj);",

"isAnswer": false

}

],

"questionType": {

"id": 1,

"name": "Multiple Choice",

"isActive": true

}

},

{

"id": 1017,

"name": "What is the difference between == and === ?",

"questionTypeId": 1,

"options": [

{

"id": 1055,

"questionId": 1010,

"name": "=== throws syntex error",

"isAnswer": false

},

{

"id": 1056,

"questionId": 1010,

"name": "== checks values only, === checks types as well",

"isAnswer": true

},

{

"id": 1057,

"questionId": 1010,

"name": "=== is reference type check only",

"isAnswer": false

},

{

"id": 1058,

"questionId": 1010,

"name": "Both are same",

"isAnswer": false

}

],

"questionType": {

"id": 1,

"name": "Multiple Choice",

"isActive": true

}

},

{

"id": 1018,

"name": "Which of the following is NOT the method of an Array?",

"questionTypeId": 1,

"options": [

{

"id": 1055,

"questionId": 1010,

"name": ".map()",

"isAnswer": false

},

{

"id": 1057,

"questionId": 1010,

"name": ".includes()",

"isAnswer": false

},

{

"id": 1056,

"questionId": 1010,

"name": ".subscribe()",

"isAnswer": true

},

{

"id": 1058,

"questionId": 1010,

"name": ".flatMap()",

"isAnswer": false

}

],

"questionType": {

"id": 1,

"name": "Multiple Choice",

"isActive": true

}

},

{

"id": 1019,

"name": "What will be the output of the following code: ['a', 'b', 'c'].fill(7, 1, 2);?",

"questionTypeId": 1,

"options": [

{

"id": 1055,

"questionId": 1010,

"name": "['a', 7, 'c']",

"isAnswer": true

},

{

"id": 1056,

"questionId": 1010,

"name": "['a', 7, 7, 'b', 'c']",

"isAnswer": false

},

{

"id": 1057,

"questionId": 1010,

"name": "['a', 'b', 'c']",

"isAnswer": false

},

{

"id": 1058,

"questionId": 1010,

"name": "['7', 7, 'c']",

"isAnswer": false

}

],

"questionType": {

"id": 1,

"name": "Multiple Choice",

"isActive": true

}

}

]

}

Style

body{

min-height: 100vh;

display: flex;

flex-direction: column;

}

.btn {

border-radius: 0;

}

.option {

background-color: #cadaee;

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;

}