**Online Test Application Source Code:**

index.html

<!doctype html>

<html lang="en">

<head>

  <meta charset="utf-8">

  <title>Online Quiz</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>

quiz.component.html

<nav class="navbar navbar-default navbar-static-top bg-info">

  <div class="container-fluid">

    <a class="navbar-brand" href="#">

      <img src="https://image.pngaaa.com/664/2956664-middle.png" alt="" width="50" height="40" class="d-inline-block align-text-top">

      <div class="col-6 text-right mt-3" style="color: white;">

       Online Test/Quiz Application

       </div>

    </a>

    <div class="col-6 text-right mt-3" style="color: white;">

      Select Quiz Type:

      <select class="form-select form-select-lg mb-3" [(ngModel)]="quizName" (change)="loadQuiz(quizName)">

        <option \*ngFor="let quiz of quizes" [value]="quiz.id">{{quiz.name}}</option>

      </select>

    </div>

  </div>

</nav>

<div class="container">

  <!--div class="row">

    <div class="col-6">

      <h2 class="fw-light">Online Test Application</h2>

    </div>

    <div class="col-6 text-right mt-3">

      Select Quiz Type:

      <select [(ngModel)]="quizName" (change)="loadQuiz(quizName)">

        <option \*ngFor="let quiz of quizes" [value]="quiz.id">{{quiz.name}}</option>

      </select>

    </div>

  </div-->

  <div id="quiz">

    <h2 class="text-center font-weight-normal">{{quiz.name}}</h2>

    <hr />

<br/>

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

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

        <div class="badge badge-info">Question {{pager.index + 1}} / {{pager.count}}.</div>

        <div \*ngIf="config.duration" class="badge badge-warning float-right">Time: {{ellapsedTime}} / {{duration}}</div>

        <h3 class="font-weight-normal">{{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="font-weight-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 />

      <div class="text-center">

        <button class="btn btn-default" \*ngIf="config.allowBack" (click)="goTo(0);">First</button>

        <button class="btn btn-default" \*ngIf="config.allowBack" (click)="goTo(pager.index - 1);">Prev</button>

        <button class="btn btn-info" (click)="goTo(pager.index + 1);">Next</button>

        <button class="btn btn-default" \*ngIf="config.allowBack" (click)="goTo(pager.count - 1);">Last</button>

        <!-- <pagination \*ngIf="config.showPager" direction-links="false" total-items="totalItems" items-per-page="itemsPerPage" ng-model="currentPage" ng-change="pageChanged()"></pagination> -->

      </div>

    </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</h2>

      <div class="alert-info" style="padding: 2rem;">

      <h4>Correct Answers: {{correctAns}}</h4>

      <h4>Your Score:{{percentScore}}%</h4>

      </div>

      <hr/>

      <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'}}">Your answer is {{isCorrect(question)}}.</div>

        </div>

      </div>

      <h4 class="alert alert-info text-center">You may close this window now.</h4>

    </div>

    <hr />

    <div class = "text-center row mb-5" \*ngIf="mode!='result'">

      <div class="col-2"></div>

      <div class="col-3"><button class="btn btn-info" style="width:90%" (click)="mode = 'quiz'">Quiz</button></div>

      <div class="col-3"><button class="btn btn-info" style="width:100%" (click)="onSubmit();">Submit Quiz</button></div>

      <div class="col-3"><button class="btn btn-info" style="width:90%" (click)="mode = 'review'">Review</button></div>

    </div>

  </div>

</div>

quiz.component.spec.ts

/\* tslint:disable:no-unused-variable \*/

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

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

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

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

describe('QuizComponent', () => {

  let component: QuizComponent;

  let fixture: ComponentFixture<QuizComponent>;

  beforeEach(async(() => {

    TestBed.configureTestingModule({

      declarations: [ QuizComponent ]

    })

    .compileComponents();

  }));

  beforeEach(() => {

    fixture = TestBed.createComponent(QuizComponent);

    component = fixture.componentInstance;

    fixture.detectChanges();

  });

  it('should create', () => {

    expect(component).toBeTruthy();

  });

});

quiz.component.ts

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

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

import { HelperService } from "../services/helper.service";

import { Option, Question, Quiz, QuizConfig } from "../models/index";

@Component({

  selector: "app-quiz",

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

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

  providers: [QuizService],

})

export class QuizComponent implements OnInit {

  quizes: any[];

  quiz: Quiz = new Quiz(null);

  mode = "quiz";

  quizName: string;

  config: QuizConfig = {

    allowBack: true,

    allowReview: true,

    autoMove: false, // if true, it will move to next question automatically when answered.

    duration: 600, // indicates the time (in secs) in which quiz needs to be completed. 0 means unlimited.

    pageSize: 1,

    requiredAll: false, // indicates if you must answer all the questions before submitting.

    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 = "";

  correctAns: number = 0;

  percentScore: number = 0;

  constructor(private quizService: QuizService) {}

  ngOnInit() {

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

    this.quizName = this.quizes[0].id;

    this.loadQuiz(this.quizName);

  }

  loadQuiz(quizName: string) {

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

      this.quiz = new Quiz(res);

      this.pager.count = 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.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,

        isCorrect: x.options,

      })

    );

    answers.forEach(function (elem) {

      elem.isCorrect.forEach(function (opt) {

        if (opt.isAnswer === true) {

          console.log(opt.name);

        }

      });

    });

    // Post your data to the server here. answers contains the questionId and the users' answer.

    console.log(this.quiz.questions);

    this.calculateResult(this.quiz.questions);

    this.mode = "result";

  }

  calculateResult(questions: Question[]) {

    this.correctAns=0;

    for (var question of questions) {

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

        this.correctAns++;

      }

    }

    this.percentScore=(this.correctAns/questions.length)\*100;

  }

}

helper.service.spec.ts

/\* tslint:disable:no-unused-variable \*/

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

import { HelperService } from './helper.service';

describe('HelperService', () => {

  beforeEach(() => {

    TestBed.configureTestingModule({

      providers: [HelperService]

    });

  });

  it('should ...', inject([HelperService], (service: HelperService) => {

    expect(service).toBeTruthy();

  }));

});

helper.service.ts

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

@Injectable()

export class HelperService {

  static toBool(val) {

    if (val === undefined || val === null || val === '' || val === 'false' || val === 'False') {

      return false;

    } else {

      return true;

    }

  }

  static shuffle(array) {

    let currentIndex = array.length, temp, randomIndex;

    while (0 !== currentIndex) {

      randomIndex = Math.floor(Math.random() \* currentIndex);

      currentIndex -= 1;

      temp = array[currentIndex];

      array[currentIndex] = array[randomIndex];

      array[randomIndex] = temp;

    }

    return array;

  }

  static extend(out) {

    out = out || {};

    for (let i = 1; i < arguments.length; i++) {

      if (!arguments[i]) {

        continue;

      }

      for (const key in arguments[i]) {

        if (arguments[i].hasOwnProperty(key)) {

          out[key] = arguments[i][key];

        }

      }

    }

    return out;

  }

}

quiz.service.ts

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

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

@Injectable()

export class QuizService {

  constructor(private http: HttpClient) { }

  get(url: string) {

    return this.http.get(url);

  }

  getAll() {

    return [

      { id: 'data/angular.json', name: 'Angular Quizz' },

      { id: 'data/JSON.json', name: 'JSON Quizz' },

      { id: 'data/JavaScript.json', name: 'JavaScript Quizz' },

      { id: 'data/Java.json' , name: 'Java Quizz' },

    ];

  }

}

app.component.html

  <app-quiz></app-quiz>

  <!-- Footer -->

<footer class="footer fixed-bottom navbar-expand-sm navbar-dark bg-dark mt-5">

  <!-- Copyright -->

  <div class="footer footer-copyright text-center py-3 " style="color: white;">© 2022 Copyright:

    Developed by Bhavana V.

    <a href="bhavana.gowdlar@gmail.com@gmail.com?subject=Query%20regarding%20the Quizz%20application%20project">Contact Us</a>

  </div>

  <!-- Copyright -->

</footer>

<!-- Footer -->

app.component.spec.ts

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

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

describe('AppComponent', () => {

  beforeEach(async(() => {

    TestBed.configureTestingModule({

      declarations: [

        AppComponent

      ],

    }).compileComponents();

  }));

  it('should create the app', async(() => {

    const fixture = TestBed.createComponent(AppComponent);

    const app = fixture.debugElement.componentInstance;

    expect(app).toBeTruthy();

  }));

  it(`should have as title 'app'`, async(() => {

    const fixture = TestBed.createComponent(AppComponent);

    const app = fixture.debugElement.componentInstance;

    expect(app.title).toEqual('app');

  }));

  it('should render title in a h1 tag', async(() => {

    const fixture = TestBed.createComponent(AppComponent);

    fixture.detectChanges();

    const compiled = fixture.debugElement.nativeElement;

    expect(compiled.querySelector('h1').textContent).toContain('Welcome to app!');

  }));

});

app.component.ts

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

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

@Component({

  selector: 'app-root',

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

  styleUrls: ['./app.component.css']

})

export class AppComponent {

}

app.module.ts

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

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

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

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

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

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

@NgModule({

  declarations: [

    AppComponent,

    QuizComponent

  ],

  imports: [

    BrowserModule,

    FormsModule,

    HttpClientModule

  ],

  providers: [],

  bootstrap: [AppComponent]

})

export class AppModule { }

Json Files

Java.json

{

    "id": 1,

    "name": "Java Quiz",

    "description": "Java Quiz (based on basic concepts)",

    "questions": [

        {

            "id": 1010,

            "name": " Who invented Java Programming?",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 1010,

                    "name": "Guido van Rossum",

                    "isAnswer": false

                },

                {

                    "id": 1056,

                    "questionId": 1010,

                    "name": "James Gosling",

                    "isAnswer": true

                },

                {

                    "id": 1057,

                    "questionId": 1010,

                    "name": "Dennis Ritchie",

                    "isAnswer": false

                },

                {

                    "id": 1058,

                    "questionId": 1010,

                    "name": "Bjarne Stroustrup",

                    "isAnswer": false

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 1011,

            "name": "Which statement is true about Java?",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 1010,

                    "name": "Java is a sequence-dependent programming language",

                    "isAnswer": false

                },

                {

                    "id": 1057,

                    "questionId": 1010,

                    "name": " Java is a code dependent programming language",

                    "isAnswer": false

                },

                {

                    "id": 1056,

                    "questionId": 1010,

                    "name": "Java is a platform-dependent programming language",

                    "isAnswer": false

                },

                {

                    "id": 1058,

                    "questionId": 1010,

                    "name": " Java is a platform-independent programming language",

                    "isAnswer": true

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 1012,

            "name": "Which component is used to compile, debug and execute the java programs? ",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 1010,

                    "name": "JRE",

                    "isAnswer": false

                },

                {

                    "id": 1057,

                    "questionId": 1010,

                    "name": "JIT",

                    "isAnswer": false

                },

                {

                    "id": 1056,

                    "questionId": 1010,

                    "name": "JDK",

                    "isAnswer": true

                },

                {

                    "id": 1058,

                    "questionId": 1010,

                    "name": "JVM",

                    "isAnswer": false

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 1013,

            "name": " Which of these cannot be used for a variable name in Java?",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 1010,

                    "name": " identifier & keyword",

                    "isAnswer": false

                },

                {

                    "id": 1057,

                    "questionId": 1010,

                    "name": "keyword",

                    "isAnswer": true

                },

                {

                    "id": 1056,

                    "questionId": 1010,

                    "name": " identifier",

                    "isAnswer": false

                },

                {

                    "id": 1058,

                    "questionId": 1010,

                    "name": "none of the mentioned ",

                    "isAnswer": false

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 1014,

            "name": " What is the extension of java code files?",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 1010,

                    "name": "'js",

                    "isAnswer": false

                },

                {

                    "id": 1057,

                    "questionId": 1010,

                    "name": ".txt",

                    "isAnswer": false

                },

                {

                    "id": 1056,

                    "questionId": 1010,

                    "name": ".class",

                    "isAnswer": false

                },

                {

                    "id": 1058,

                    "questionId": 1010,

                    "name": ".java",

                    "isAnswer": true

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 1015,

            "name": "Which of the following is a superclass of every class in Java?",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 1010,

                    "name": "ArrayList",

                    "isAnswer": false

                },

                {

                    "id": 1057,

                    "questionId": 1010,

                    "name": "Abstract class",

                    "isAnswer": false

                },

                {

                    "id": 1056,

                    "questionId": 1010,

                    "name": "Object class",

                    "isAnswer": true

                },

                {

                    "id": 1058,

                    "questionId": 1010,

                    "name": " String",

                    "isAnswer": false

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 1016,

            "name": "Which of these keywords is used to define interfaces in Java?",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 1010,

                    "name": "intf",

                    "isAnswer": false

                },

                {

                    "id": 1056,

                    "questionId": 1010,

                    "name": "Intf",

                    "isAnswer": true

                },

                {

                    "id": 1057,

                    "questionId": 1010,

                    "name": "interface",

                    "isAnswer": false

                },

                {

                    "id": 1058,

                    "questionId": 1010,

                    "name": "Interface",

                    "isAnswer": false

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 1017,

            "name": "Which of these are selection statements in Java?",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 1010,

                    "name": "break",

                    "isAnswer": false

                },

                {

                    "id": 1056,

                    "questionId": 1010,

                    "name": "continue",

                    "isAnswer": false

                },

                {

                    "id": 1057,

                    "questionId": 1010,

                    "name": "for()",

                    "isAnswer": false

                },

                {

                    "id": 1058,

                    "questionId": 1010,

                    "name": "if()",

                    "isAnswer": true

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 1018,

            "name": " Which of the following is not an OOPS concept in Java?",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 1010,

                    "name": "Polymorphism",

                    "isAnswer": false

                },

                {

                    "id": 1057,

                    "questionId": 1010,

                    "name": "Inheritance",

                    "isAnswer": false

                },

                {

                    "id": 1056,

                    "questionId": 1010,

                    "name": "Compilation",

                    "isAnswer": true

                },

                {

                    "id": 1058,

                    "questionId": 1010,

                    "name": "Encapsulation",

                    "isAnswer": false

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 1019,

            "name": "Which environment variable is used to set the java path?",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 1010,

                    "name": "MAVEN\_Path",

                    "isAnswer": false

                },

                {

                    "id": 1056,

                    "questionId": 1010,

                    "name": "JavaPATH",

                    "isAnswer": false

                },

                {

                    "id": 1057,

                    "questionId": 1010,

                    "name": "AVA",

                    "isAnswer": false

                },

                {

                    "id": 1058,

                    "questionId": 1010,

                    "name": "JAVA\_HOME",

                    "isAnswer": true

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        }

    ]

}

angular.json

{

    "id": 1,

    "name": "Angular Quiz",

    "description": "Angular Quiz (based on basic concepts)",

    "questions": [

        {

            "id": 1010,

            "name": "Which of the following is correct about tsconfig.json?",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 1010,

                    "name": "This file is used to give the options about TypeScript used for the Angular JS project.",

                    "isAnswer": true

                },

                {

                    "id": 1056,

                    "questionId": 1010,

                    "name": "This file contains information about Angular 2 project.",

                    "isAnswer": false

                },

                {

                    "id": 1057,

                    "questionId": 1010,

                    "name": "This file contains the system files required for Angular JS application.",

                    "isAnswer": false

                },

                {

                    "id": 1058,

                    "questionId": 1010,

                    "name": "All of the above.",

                    "isAnswer": false

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 1011,

            "name": "Which of the following is true about lowercase filter?",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 1010,

                    "name": "Lowercase filter converts a text to lower case text.",

                    "isAnswer": true

                },

                {

                    "id": 1057,

                    "questionId": 1010,

                    "name": "Lowercase filter is a function which takes text as input.",

                    "isAnswer": false

                },

                {

                    "id": 1056,

                    "questionId": 1010,

                    "name": "Both of the above.",

                    "isAnswer": false

                },

                {

                    "id": 1058,

                    "questionId": 1010,

                    "name": "None of the above.",

                    "isAnswer": false

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 1012,

            "name": " Who is sometimes called as father of Angularjs?",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 1010,

                    "name": "Brad Green",

                    "isAnswer": false

                },

                {

                    "id": 1057,

                    "questionId": 1010,

                    "name": "Misko Hevery",

                    "isAnswer": true

                },

                {

                    "id": 1056,

                    "questionId": 1010,

                    "name": "Brad Richardson",

                    "isAnswer": false

                },

                {

                    "id": 1058,

                    "questionId": 1010,

                    "name": "Chuck Jazdzewski",

                    "isAnswer": false

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 1013,

            "name": "Which of the following is correct about TypeScript?",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 1010,

                    "name": "Angular is based on TypeScript.",

                    "isAnswer": false

                },

                {

                    "id": 1057,

                    "questionId": 1010,

                    "name": "This is a superset of JavaScript.",

                    "isAnswer": false

                },

                {

                    "id": 1056,

                    "questionId": 1010,

                    "name": "TypeScript is maintained by Microsoft.",

                    "isAnswer": false

                },

                {

                    "id": 1058,

                    "questionId": 1010,

                    "name": "All of the above.",

                    "isAnswer": true

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 1014,

            "name": "What is the decorator used for configuring your module class?",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 1010,

                    "name": "@NgModule",

                    "isAnswer": true

                },

                {

                    "id": 1057,

                    "questionId": 1010,

                    "name": "@NgApp",

                    "isAnswer": false

                },

                {

                    "id": 1056,

                    "questionId": 1010,

                    "name": "Both",

                    "isAnswer": false

                },

                {

                    "id": 1058,

                    "questionId": 1010,

                    "name": "None of above",

                    "isAnswer": false

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 1015,

            "name": "Which angular decorator allows us to define the pipe name that is globally available for use in any template in the across application?",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 1010,

                    "name": "pipeName",

                    "isAnswer": false

                },

                {

                    "id": 1057,

                    "questionId": 1010,

                    "name": "pipeDeco",

                    "isAnswer": false

                },

                {

                    "id": 1056,

                    "questionId": 1010,

                    "name": "Pipe",

                    "isAnswer": true

                },

                {

                    "id": 1058,

                    "questionId": 1010,

                    "name": "None of the above",

                    "isAnswer": false

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 1016,

            "name": "Router is part of which of the following module?",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 1010,

                    "name": "@angular/core",

                    "isAnswer": false

                },

                {

                    "id": 1056,

                    "questionId": 1010,

                    "name": "@angular/router",

                    "isAnswer": true

                },

                {

                    "id": 1057,

                    "questionId": 1010,

                    "name": "Both",

                    "isAnswer": false

                },

                {

                    "id": 1058,

                    "questionId": 1010,

                    "name": "None of the above",

                    "isAnswer": false

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 1017,

            "name": "Which character is used for chaining multiple pipes?",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 1010,

                    "name": "|",

                    "isAnswer": true

                },

                {

                    "id": 1056,

                    "questionId": 1010,

                    "name": ":",

                    "isAnswer": false

                },

                {

                    "id": 1057,

                    "questionId": 1010,

                    "name": "Both",

                    "isAnswer": false

                },

                {

                    "id": 1058,

                    "questionId": 1010,

                    "name": "None of the above",

                    "isAnswer": false

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 1018,

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

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 1010,

                    "name": "ahead-of-time compilation",

                    "isAnswer": true

                },

                {

                    "id": 1057,

                    "questionId": 1010,

                    "name": "Angular Object Templates",

                    "isAnswer": false

                },

                {

                    "id": 1056,

                    "questionId": 1010,

                    "name": "Angular Object Templates",

                    "isAnswer": false

                },

                {

                    "id": 1058,

                    "questionId": 1010,

                    "name": "None of the above",

                    "isAnswer": false

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 1019,

            "name": "Which of the following is correct about package.json?",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 1010,

                    "name": "This file is used to give the options about TypeScript used for the Angular JS project.",

                    "isAnswer": false

                },

                {

                    "id": 1056,

                    "questionId": 1010,

                    "name": "This file contains information about Angular project.",

                    "isAnswer": true

                },

                {

                    "id": 1057,

                    "questionId": 1010,

                    "name": "This file contains the system files required for Angular JS application.",

                    "isAnswer": false

                },

                {

                    "id": 1058,

                    "questionId": 1010,

                    "name": "All of the above.",

                    "isAnswer": false

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        }

    ]

}

JavaScript.json

{

    "id": 1,

    "name": "JavaScript Quiz",

    "description": "JavaScript Quiz (based on basic concepts)",

    "questions": [

        {

            "id": 1010,

            "name": "Which is not valid data type in Javascript.",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 1010,

                    "name": "Undefinded",

                    "isAnswer": false

                },

                {

                    "id": 1056,

                    "questionId": 1010,

                    "name": "Boolean",

                    "isAnswer": false

                },

                {

                    "id": 1057,

                    "questionId": 1010,

                    "name": "float",

                    "isAnswer": true

                },

                {

                    "id": 1058,

                    "questionId": 1010,

                    "name": "Number",

                    "isAnswer": false

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 1011,

            "name": "By which javascript code is used ?",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 1010,

                    "name": "Class.",

                    "isAnswer": false

                },

                {

                    "id": 1057,

                    "questionId": 3030,

                    "name": "Methods/Functions.",

                    "isAnswer": false

                },

                {

                    "id": 1056,

                    "questionId": 3030,

                    "name": "Event.",

                    "isAnswer": true

                },

                {

                    "id": 1058,

                    "questionId": 3030,

                    "name": "None of the above.",

                    "isAnswer": false

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 3012,

            "name": "  Among the given statements, which statement defines closures in JavaScript??",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 3030,

                    "name": "It’s a function that is enclosed with references to its inner function scope",

                    "isAnswer": false

                },

                {

                    "id": 1057,

                    "questionId": 3030,

                    "name": "It’s a function that is enclosed with references to its lexical environment",

                    "isAnswer": true

                },

                {

                    "id": 1056,

                    "questionId": 3030,

                    "name": "It’s a function that is enclosed with the object to its inner function scope",

                    "isAnswer": false

                },

                {

                    "id": 1058,

                    "questionId": 3030,

                    "name": "None of the above",

                    "isAnswer": false

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 3013,

            "name": "Will the following JavaScript code work? var tensquared = (function(x) {return x\*x;}(10));",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 3030,

                    "name": "Exception will be thrown.",

                    "isAnswer": false

                },

                {

                    "id": 1057,

                    "questionId": 3030,

                    "name": "Memory leak.",

                    "isAnswer": false

                },

                {

                    "id": 1056,

                    "questionId": 3030,

                    "name": "Error.",

                    "isAnswer": false

                },

                {

                    "id": 1058,

                    "questionId": 3030,

                    "name": "Yes, perfectly.",

                    "isAnswer": true

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 3014,

            "name": "Which of the following is correct about callbacks?",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 3030,

                    "name": "A callback is a plain JavaScript function passed to some method as an argument or option",

                    "isAnswer": false

                },

                {

                    "id": 1057,

                    "questionId": 3030,

                    "name": "Some callbacks are just events, called to give the user a chance to react when a certain state is triggered.",

                    "isAnswer": false

                },

                {

                    "id": 1056,

                    "questionId": 3030,

                    "name": "Both",

                    "isAnswer": true

                },

                {

                    "id": 1058,

                    "questionId": 3030,

                    "name": "None of above",

                    "isAnswer": false

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 3015,

            "name": "Which of the following code creates an object?",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 3030,

                    "name": "var book = Object();",

                    "isAnswer": false

                },

                {

                    "id": 1057,

                    "questionId": 3030,

                    "name": "var book = new Object();",

                    "isAnswer": true

                },

                {

                    "id": 1056,

                    "questionId": 3030,

                    "name": "var book = new OBJECT();",

                    "isAnswer": false

                },

                {

                    "id": 1058,

                    "questionId": 3030,

                    "name": "var book = new Book();",

                    "isAnswer": false

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 3016,

            "name": "Which of the following function of Boolean object returns a string containing the source of the Boolean object?",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 3030,

                    "name": " toSource()",

                    "isAnswer": true

                },

                {

                    "id": 1056,

                    "questionId": 3030,

                    "name": "valueOf()",

                    "isAnswer": false

                },

                {

                    "id": 1057,

                    "questionId": 3030,

                    "name": "toString()",

                    "isAnswer": false

                },

                {

                    "id": 1058,

                    "questionId": 3030,

                    "name": "None of the above",

                    "isAnswer": false

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 3017,

            "name": "Which of the following function of Array object creates a new array with all of the elements of this array for which the provided filtering function returns true?",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 3030,

                    "name": "concat()",

                    "isAnswer": false

                },

                {

                    "id": 1056,

                    "questionId": 3030,

                    "name": "every()",

                    "isAnswer": false

                },

                {

                    "id": 1057,

                    "questionId": 3030,

                    "name": "filter()",

                    "isAnswer": true

                },

                {

                    "id": 1058,

                    "questionId": 3030,

                    "name": "some()",

                    "isAnswer": false

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 3018,

            "name": "Which one of the following also known as Conditional Expression:",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 3030,

                    "name": "Alternative to if-else",

                    "isAnswer": false

                },

                {

                    "id": 1057,

                    "questionId": 3030,

                    "name": "Switch statement",

                    "isAnswer": false

                },

                {

                    "id": 1056,

                    "questionId": 3030,

                    "name": "If-then-else statement",

                    "isAnswer": false

                },

                {

                    "id": 1058,

                    "questionId": 3030,

                    "name": "immediate if",

                    "isAnswer": true

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 3019,

            "name": "Which of the following type of a variable is volatile?",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 3030,

                    "name": "Mutable variable.",

                    "isAnswer": true

                },

                {

                    "id": 1056,

                    "questionId": 3030,

                    "name": "Dynamic variable.",

                    "isAnswer": false

                },

                {

                    "id": 1057,

                    "questionId": 3030,

                    "name": "Volatile variable.",

                    "isAnswer": false

                },

                {

                    "id": 1058,

                    "questionId": 3030,

                    "name": "Immutable variable.",

                    "isAnswer": false

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        }

    ]

}

JSON.json

{

    "id": 1,

    "name": "JSON Quiz",

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

    "questions": [

        {

            "id": 1010,

            "name": "JSON Stands for \_\_\_\_\_\_\_\_?",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 1010,

                    "name": "Java Standard Output Network",

                    "isAnswer": false

                },

                {

                    "id": 1056,

                    "questionId": 1010,

                    "name": "JavaScript Object Notation",

                    "isAnswer": true

                },

                {

                    "id": 1057,

                    "questionId": 1010,

                    "name": "JavaScript Output Name",

                    "isAnswer": false

                },

                {

                    "id": 1058,

                    "questionId": 1010,

                    "name": "Java Source Open Network",

                    "isAnswer": false

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 1011,

            "name": "Which of the following is not a type in JSON ?",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 1010,

                    "name": "date",

                    "isAnswer": true

                },

                {

                    "id": 1057,

                    "questionId": 1010,

                    "name": "object",

                    "isAnswer": false

                },

                {

                    "id": 1056,

                    "questionId": 1010,

                    "name": "Array",

                    "isAnswer": false

                },

                {

                    "id": 1058,

                    "questionId": 1010,

                    "name": "string",

                    "isAnswer": false

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 1012,

            "name": "Who is the Father of JSON?",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 1010,

                    "name": "Douglas Crockford",

                    "isAnswer": true

                },

                {

                    "id": 1057,

                    "questionId": 1010,

                    "name": "Rasmus Lerdorf",

                    "isAnswer": false

                },

                {

                    "id": 1056,

                    "questionId": 1010,

                    "name": "Douglas Michel",

                    "isAnswer": false

                },

                {

                    "id": 1058,

                    "questionId": 1010,

                    "name": "Dennis Ritchie",

                    "isAnswer": false

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 1013,

            "name": "What extension is used to save a JSON file?",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 1010,

                    "name": ".json",

                    "isAnswer": true

                },

                {

                    "id": 1057,

                    "questionId": 1010,

                    "name": ".js",

                    "isAnswer": false

                },

                {

                    "id": 1056,

                    "questionId": 1010,

                    "name": ".javaN",

                    "isAnswer": false

                },

                {

                    "id": 1058,

                    "questionId": 1010,

                    "name": ".on",

                    "isAnswer": false

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 1014,

            "name": "JSON string have used in \_\_\_\_\_\_\_\_\_\_\_\_",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 1010,

                    "name": "Duoble Quote",

                    "isAnswer": true

                },

                {

                    "id": 1057,

                    "questionId": 1010,

                    "name": "Single Quote",

                    "isAnswer": false

                },

                {

                    "id": 1056,

                    "questionId": 1010,

                    "name": "Both 1 & 2",

                    "isAnswer": false

                },

                {

                    "id": 1058,

                    "questionId": 1010,

                    "name": "None of the above",

                    "isAnswer": false

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 1015,

            "name": "JSON elements are separated by the \_\_\_\_\_\_",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 1010,

                    "name": "line break",

                    "isAnswer": false

                },

                {

                    "id": 1057,

                    "questionId": 1010,

                    "name": "Semi-Colon",

                    "isAnswer": false

                },

                {

                    "id": 1056,

                    "questionId": 1010,

                    "name": "comma",

                    "isAnswer": true

                },

                {

                    "id": 1058,

                    "questionId": 1010,

                    "name": "white-space",

                    "isAnswer": false

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 1016,

            "name": "The MIME type of JSON is \_\_\_\_\_",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 1010,

                    "name": "application/json",

                    "isAnswer": true

                },

                {

                    "id": 1056,

                    "questionId": 1010,

                    "name": "application/x-json",

                    "isAnswer": false

                },

                {

                    "id": 1057,

                    "questionId": 1010,

                    "name": "application/javascript",

                    "isAnswer": false

                },

                {

                    "id": 1058,

                    "questionId": 1010,

                    "name": "text/json",

                    "isAnswer": false

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 1017,

            "name": "What is the common usage of JSON on modern websites?",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 1010,

                    "name": "To store information remotely",

                    "isAnswer": false

                },

                {

                    "id": 1056,

                    "questionId": 1010,

                    "name": "To send and receive bits of data",

                    "isAnswer": true

                },

                {

                    "id": 1057,

                    "questionId": 1010,

                    "name": "To store information locally",

                    "isAnswer": false

                },

                {

                    "id": 1058,

                    "questionId": 1010,

                    "name": "All of the Above",

                    "isAnswer": false

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 1018,

            "name": "Which of the following code will throw an error?",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 1010,

                    "name": "JSON.parse(‘{}’);",

                    "isAnswer": false

                },

                {

                    "id": 1057,

                    "questionId": 1010,

                    "name": "JSON.parse(null);",

                    "isAnswer": false

                },

                {

                    "id": 1056,

                    "questionId": 1010,

                    "name": "JSON.parse(undefined);",

                    "isAnswer": true

                },

                {

                    "id": 1058,

                    "questionId": 1010,

                    "name": "JSON.parse(‘[]’);",

                    "isAnswer": false

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 1019,

            "name": "\_\_\_\_\_\_\_ is not a valid way to parse JSON string",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 1010,

                    "name": "JSON.parse()",

                    "isAnswer": false

                },

                {

                    "id": 1056,

                    "questionId": 1010,

                    "name": "JSON.eval()",

                    "isAnswer": true

                },

                {

                    "id": 1057,

                    "questionId": 1010,

                    "name": "jQuery.parseJSON()",

                    "isAnswer": false

                },

                {

                    "id": 1058,

                    "questionId": 1010,

                    "name": "None of the above",

                    "isAnswer": false

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        }

    ]

}

test.ts

// This file is required by karma.conf.js and loads recursively all the .spec and framework files

import 'zone.js/dist/zone-testing';

import { getTestBed } from '@angular/core/testing';

import {

  BrowserDynamicTestingModule,

  platformBrowserDynamicTesting

} from '@angular/platform-browser-dynamic/testing';

declare const require: any;

// First, initialize the Angular testing environment.

getTestBed().initTestEnvironment(

  BrowserDynamicTestingModule,

  platformBrowserDynamicTesting()

);

// Then we find all the tests.

const context = require.context('./', true, /\.spec\.ts$/);

// And load the modules.

context.keys().map(context);

main.ts

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

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

import { AppModule } from './app/app.module';

import { environment } from './environments/environment';

if (environment.production) {

  enableProdMode();

}

platformBrowserDynamic().bootstrapModule(AppModule)

  .catch(err => console.log(err));

styles.css

.btn{

    border-radius: 4;

}

.h1{

    text-decoration-color: aqua;

}

.option{

    background-color:#faf8c07d;

    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;

        text-align: center;

    }

    h2,.h2{

        font-size:20px;

    }

    h3,.h3{

        font-size:18px;

    }

    .option {

        font-size: 14px;

        padding: 6px;

    }

    footer{

        opacity: 0.5;

        height:3rem;

    }

    .navbar{

            opacity: 0.8;

    }

}

index.ts

export \* from './option';

export \* from './question';

export \* from './quiz';

export \* from './quiz-config';

option.ts

export class Option {

    id: number;

    questionId: number;

    name: string;

    isAnswer: boolean;

    selected: boolean;

    constructor(data: any) {

        data = data || {};

        this.id = data.id;

        this.questionId = data.questionId;

        this.name = data.name;

        this.isAnswer = data.isAnswer;

    }

}

question.ts

import { Option } from './option';

export class Question {

    id: number;

    name: string;

    questionTypeId: number;

    options: Option[];

    answered: boolean;

    constructor(data: any) {

        data = data || {};

        this.id = data.id;

        this.name = data.name;

        this.questionTypeId = data.questionTypeId;

        this.options = [];

        data.options.forEach(o => {

            this.options.push(new Option(o));

        });

    }

}

quiz-config.ts

export class QuizConfig {

    allowBack: boolean;

    allowReview: boolean;

    autoMove: boolean;  // if boolean; it will move to next question automatically when answered.

    duration: number;  // indicates the time in which quiz needs to be completed. 0 means unlimited.

    pageSize: number;

    requiredAll: boolean;  // indicates if you must answer all the questions before submitting.

    richText: boolean;

    shuffleQuestions: boolean;

    shuffleOptions: boolean;

    showClock: boolean;

    showPager: boolean;

    theme: string;

    constructor(data: any) {

        data = data || {};

        this.allowBack = data.allowBack;

        this.allowReview = data.allowReview;

        this.autoMove = data.autoMove;

        this.duration = data.duration;

        this.pageSize = data.pageSize;

        this.requiredAll = data.requiredAll;

        this.richText = data.richText;

        this.shuffleQuestions = data.shuffleQuestions;

        this.shuffleOptions = data.shuffleOptions;

        this.showClock = data.showClock;

        this.showPager = data.showPager;

    }

}

quiz.ts

import { QuizConfig } from './quiz-config';

import { Question } from './question';

export class Quiz {

    id: number;

    name: string;

    description: string;

    config: QuizConfig;

    questions: Question[];

    constructor(data: any) {

        if (data) {

            this.id = data.id;

            this.name = data.name;

            this.description = data.description;

            this.config = new QuizConfig(data.config);

            this.questions = [];

            data.questions.forEach(q => {

                this.questions.push(new Question(q));

            });

        }

    }

}