

Online Test Applications

Version History:

Author	Shaik Ibrahim
Purpose	Online Test Applications
Date	December 15, 2021.
Version	13.1

Contents:

1. Modules of the project:	2
2. Git hub link:	2
3. Project Code:	3
4. Sprint Work.	19

1. Project Modules:

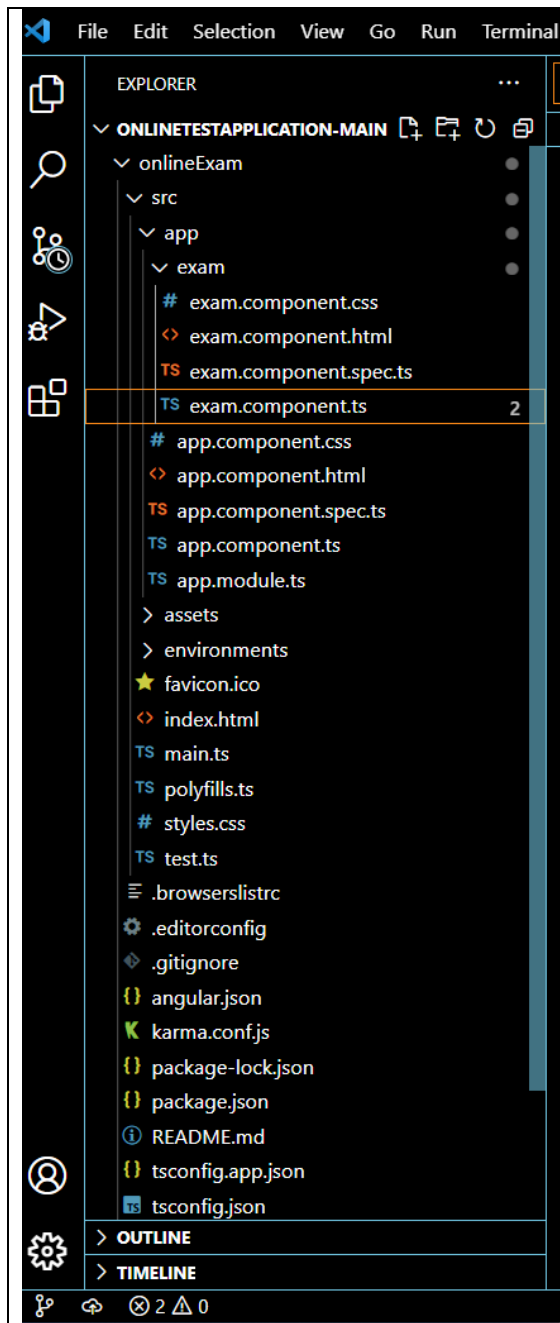
1. Node js.
2. Visual Studios.
3. Angular 13.3.1
4. Angular .json
5. HTML
6. Css
7. Java Scriptor
8. karma configuration model.
9. Sql Data Base
10. PowerShell

2. Git hub link:

Repository name:	FourthPhaseProject
GitHub link	https://github.com/skibrahimsk/FourthPhaseProject.git

3. Project Code:

Folder Structure



angular.json - OnlineTestApplication-main - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

onlineExam > angular.json > {} projects > {} onlineExam > {} architect > {} build > {} configurations > {} production > {} budgets > {} 0

```
68 "builder": "@angular-devkit/build-angular:dev-server",
69 "configurations": {
70   "production": {
71     "browserTarget": "onlineExam:build:production"
72   },
73   "development": {
74     "browserTarget": "onlineExam:build:development"
75   }
76 },
77 "defaultConfiguration": "development"
78 },
79 "extract-i18n": {
80   "builder": "@angular-devkit/build-angular:extract-i18n",
81   "options": {
82     "browserTarget": "onlineExam:build"
83   }
84 },
85 "test": {
86   "builder": "@angular-devkit/build-angular:karma",
87   "options": {
88     "main": "src/test.ts",
89     "polyfills": "src/polyfills.ts",
90     "tsConfig": "tsconfig.spec.json",
91     "karmaConfig": "karma.conf.js",
92     "assets": [
93       "src/favicon.ico",
94       "src/assets"
95     ],
96     "styles": [
97       "src/styles.css"
98     ],
99     "scripts": []
100   }
101 }
102 }
103 }
104 },
105 "defaultProject": "onlineExam"
```

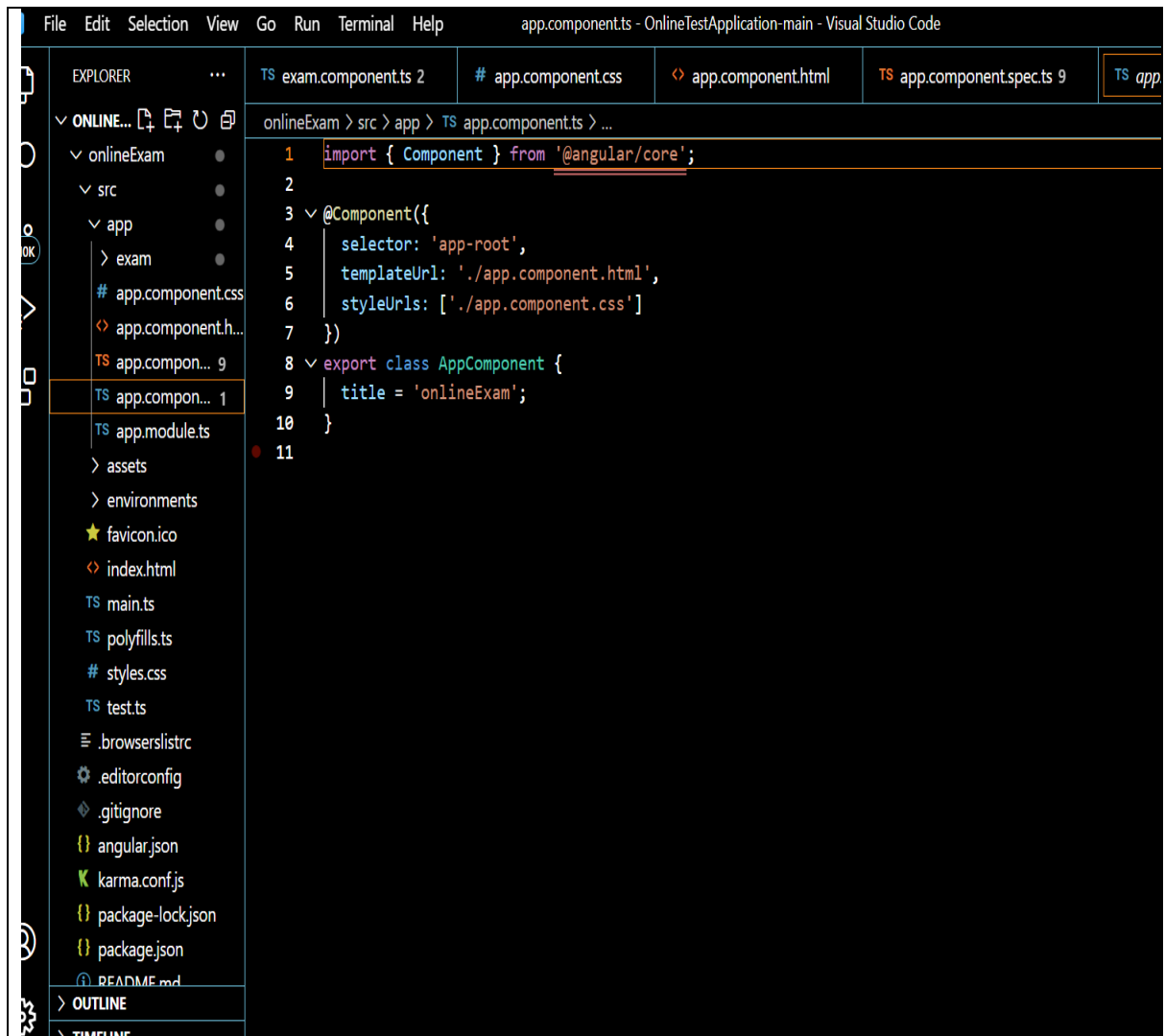
main* 11 1

Ln 39, Col 45 Spaces: 2 UTF-8

```
File Edit Selection View Go Run Terminal Help app.component.spec.ts - OnlineTestApplication-main - Visual Studio Code

EXPLORER
ONLINETESTAPPLICATION...
  onlineExam
    src
      app
        exam
          # app.component.css
          <> app.component.h...
          TS app.compon... 9
          TS app.component.ts
          TS app.module.ts
        > assets
        > environments
        ★ favicon.ico
        <> index.html
        TS main.ts
        TS polyfills.ts
        # styles.css
        TS test.ts
        .browserslistrc
        .editorconfig
        .gitignore
        {} angular.json
        {} karma.conf.js
        {} package-lock.json
        {} package.json
        README.md
      > OUTLINE
      > TIMELINE

TS exam.component.spec.ts 2
import { AppComponent } from './app.component';
3
4 describe('AppComponent', () => {
5   beforeEach(async () => {
6     await TestBed.configureTestingModule({
7       declarations: [
8         AppComponent
9       ],
10    }).compileComponents();
11  });
12
13  it('should create the app', () => {
14    const fixture = TestBed.createComponent(AppComponent);
15    const app = fixture.componentInstance;
16    expect(app).toBeTruthy();
17  });
18
19  it('should have as title 'onlineExam'', () => {
20    const fixture = TestBed.createComponent(AppComponent);
21    const app = fixture.componentInstance;
22    expect(app.title).toEqual('onlineExam');
23  });
24
25  it('should render title', () => {
26    const fixture = TestBed.createComponent(AppComponent);
27    fixture.detectChanges();
28    const compiled = fixture.nativeElement as HTMLElement;
29    expect(compiled.querySelector('.content span')?.textContent).toC
30  });
31
32
```

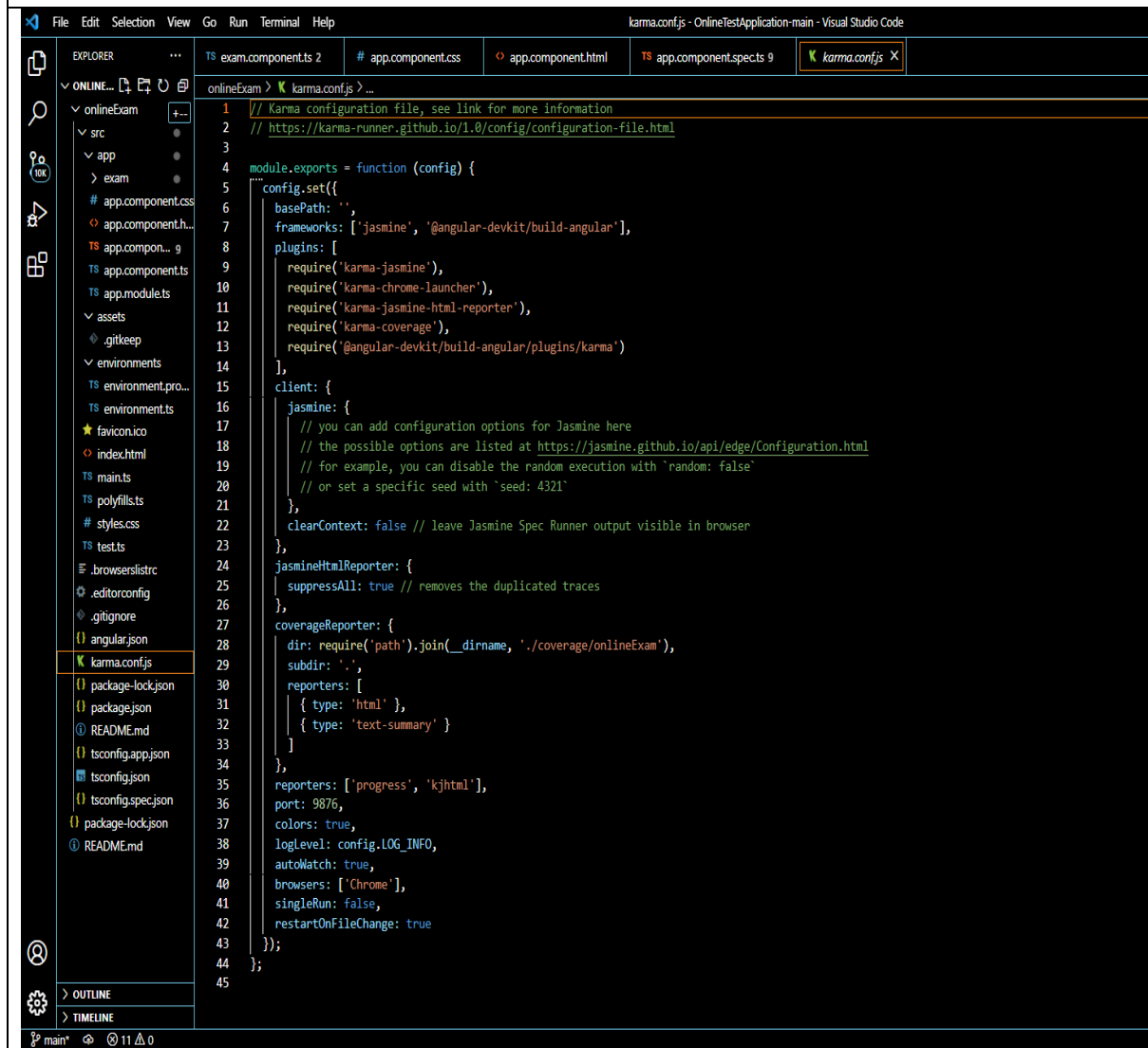


The screenshot shows the Visual Studio Code interface with the following components:

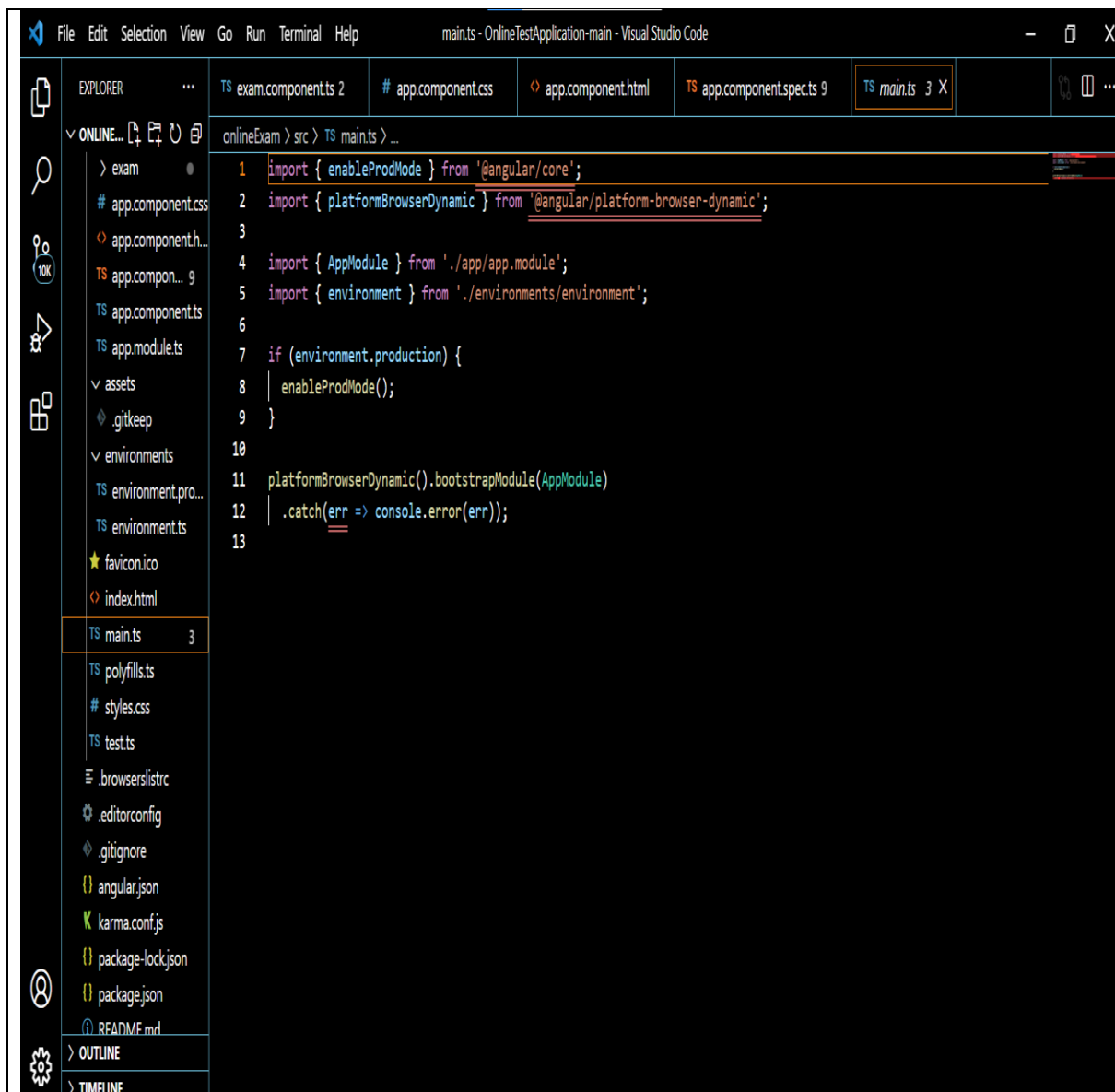
- Explorer Panel:** Displays the project structure. The 'src' directory is expanded, showing 'app' and 'exam' subdirectories. The 'app' directory contains 'app.component.css', 'app.component.html', and 'app.component.ts'. The 'exam' directory contains 'exam.component.ts'. Other files in the project include 'assets', 'environments', 'favicon.ico', 'index.html', 'main.ts', 'polyfills.ts', 'styles.css', 'test.ts', '.browserslistrc', '.editorconfig', '.gitignore', 'angular.json', 'karma.conf.js', 'package-lock.json', 'package.json', and 'README.md'.
- Editor Panel:** Shows the 'app.module.ts' file. The code is as follows:

```
1 import { NgModule } from '@angular/core';
2 import { FormsModule } from '@angular/forms';
3 import { BrowserModule } from '@angular/platform-browser';
4
5 import { AppComponent } from './app.component';
6 import { ExamComponent } from './exam/exam.component';
7
8 @NgModule({
9   de (alias) class ExamComponent
10     import ExamComponent
11     ExamComponent
12   },
13   imports: [
14     BrowserModule, FormsModule
15   ],
16   providers: [],
17   bootstrap: [AppComponent]
18 })
19 export class AppModule { }
20
```
- Terminal Panel:** Shows the command 'onlineExam > src > app > TS app.module.ts > ...'.
- Bottom Bar:** Shows the status bar with 'main*' and 'Ln 1, 0'.

Karma.conf.js



```
1 // Karma configuration file, see link for more information
2 // https://karma-runner.github.io/1.0/config/configuration-file.html
3
4 module.exports = function (config) {
5   config.set({
6     basePath: '',
7     frameworks: ['jasmine', '@angular-devkit/build-angular'],
8     plugins: [
9       require('karma-jasmine'),
10      require('karma-chrome-launcher'),
11      require('karma-jasmine-html-reporter'),
12      require('karma-coverage'),
13      require('@angular-devkit/build-angular/plugins/karma')
14    ],
15    client: {
16      jasmine: {
17        // you can add configuration options for Jasmine here
18        // the possible options are listed at https://jasmine.github.io/api/edge/Configuration.html
19        // for example, you can disable the random execution with 'random: false'
20        // or set a specific seed with 'seed: 4321'
21      },
22      clearContext: false // leave Jasmine Spec Runner output visible in browser
23    },
24    jasmineHtmlReporter: {
25      suppressAll: true // removes the duplicated traces
26    },
27    coverageReporter: {
28      dir: require('path').join(__dirname, './coverage/onlineExam'),
29      subdir: '.',
30      reporters: [
31        { type: 'html' },
32        { type: 'text-summary' }
33      ]
34    },
35    reporters: ['progress', 'kjhtml'],
36    port: 9876,
37    colors: true,
38    logLevel: config.LOG_INFO,
39    autoWatch: true,
40    browsers: ['Chrome'],
41    singleRun: false,
42    restartOnFileChange: true
43  });
44 };
45
```



Package:

Package

The screenshot shows the Visual Studio Code editor with the `package.json` file open. The Explorer sidebar on the left shows the project structure, including `src`, `app`, `exam`, and various configuration files. The main editor displays the `package.json` content, which includes the application name, version, scripts for building and testing, and a list of dependencies and devDependencies.

```
1 {
2   "name": "online-exam",
3   "version": "0.0.0",
4   "scripts": {
5     "ng": "ng",
6     "start": "ng serve",
7     "build": "ng build",
8     "watch": "ng build --watch --configuration development",
9     "test": "ng test"
10  },
11  "private": true,
12  "dependencies": {
13    "@angular/animations": "~12.2.0",
14    "@angular/common": "~12.2.0",
15    "@angular/compiler": "~12.2.0",
16    "@angular/core": "~12.2.0",
17    "@angular/forms": "~12.2.0",
18    "@angular/platform-browser": "~12.2.0",
19    "@angular/platform-browser-dynamic": "~12.2.0",
20    "@angular/router": "~12.2.0",
21    "rxjs": "~6.6.0",
22    "tslib": "^2.3.0",
23    "zone.js": "~0.11.4"
24  },
25  "devDependencies": {
26    "@angular-devkit/build-angular": "~12.2.12",
27    "@angular/cli": "~12.2.12",
28    "@angular/compiler-cli": "~12.2.0",
29    "@types/jasmine": "~3.8.0",
30    "@types/node": "^12.11.1",
31    "jasmine-core": "~3.8.0",
32    "karma": "~6.3.0",
33    "karma-chrome-launcher": "~3.1.0",
34    "karma-coverage": "~2.0.3",
35    "karma-jasmine": "~4.0.0",
36    "karma-jasmine-html-reporter": "~1.7.0",
37    "typescript": "~4.3.5"
38  }
39 }
```

Test.ts

The screenshot shows the Visual Studio Code editor with the `test.ts` file open. The Explorer sidebar on the left shows the project structure, including `src`, `app`, `exam`, and various configuration files. The main editor displays the `test.ts` content, which includes imports for testing utilities, a `require` function declaration, and the initialization of the Angular testing environment.

```
1 // This file is required by karma.conf.js and loads recursively all the .spec and framework files
2
3 import 'zone.js/testing';
4 import { getTestBed } from '@angular/core/testing';
5 import {
6   BrowserDynamicTestingModule,
7   platformBrowserDynamicTesting
8 } from '@angular/platform-browser-dynamic/testing';
9
10 declare const require: {
11   context(path: string, deep?: boolean, filter?: RegExp): {
12     keys(): string[];
13     <T>(id: string): T;
14   };
15 };
16
17 // First, initialize the Angular testing environment.
18 getTestBed().initTestEnvironment(
19   BrowserDynamicTestingModule,
20   platformBrowserDynamicTesting(),
21   { teardown: { destroyAfterEach: true } },
22 );
23
24 // Then we find all the tests.
25 const context = require.context('./', true, /\.spec\.ts$/);
26 // And load the modules.
27 context.keys().map(context);
```

tsconfig.json

```
{
  "extends": "./tsconfig.json",
  "compilerOptions": {
    "outDir": "./out-tsc/app",
    "types": []
  },
  "files": [
    "src/main.ts",
    "src/polyfills.ts"
  ],
  "include": [
    "src/**/*.d.ts"
  ]
}
```

tsconfig.json

```
{
  "compileOnSave": false,
  "compilerOptions": {
    "baseUrl": "./",
    "outDir": "./dist/out-tsc",
    "forceConsistentCasingInFileNames": true,
    "strict": true,
    "noImplicitReturns": true,
    "noFallthroughCasesInSwitch": true,
    "sourceMap": true,
    "declaration": false,
    "downlevelIteration": true,
    "experimentalDecorators": true,
    "moduleResolution": "node",
    "importHelpers": true,
    "target": "es2017",
    "module": "es2020",
    "lib": [
      "es2018",
      "dom"
    ]
  },
  "angularCompilerOptions": {
    "enableI18nLegacyMessageIdFormat": false,
    "strictInjectionParameters": true,
    "strictInputAccessModifiers": true,
    "strictTemplates": true
  }
}
```

Tsconfig/spec

```
/* To learn more about this file see: https://angular.io/config/tsconfig. */
{
  "extends": "./tsconfig.json",
  "compilerOptions": {
    "outDir": "./out-tsc/spec",
    "types": [
      "jasmine"
    ]
  },
  "files": [
    "src/test.ts",
    "src/polyfills.ts"
  ],
  "include": [
    "src/**/*.spec.ts",
    "src/**/*.d.ts"
  ]
}
```

Index.html

```
<!doctype html>
<html lang="en">
<head>
```

```

<meta charset="utf-8">
<title>OnlineExam</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@4.6.1/dist/css/bootstrap.min.css" integrity="sha384-
zCbKRCUGaJDkqS1kPbPd7TveP5iyJE0EjAuZQTgFLD2ylzuqKfdKlfG/eSrtxUkn"
crossorigin="anonymous">
</head>
<body>
  <app-root></app-root>
</body>
</html>

```

Exam.component

```

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

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

  correctAns:number=0;
  incorrectAns:number=0;
  submitted:boolean=false;
  flag:boolean=false;
  Attempt:string[][]=[];

  constructor() { }

  ngOnInit(): void {
  }
  questions=[
    {"question":"1000+1=
    _____", "op":["10001","1001","1000","100001"],"ans":"1001","attempt":null
  },

```

```

    {"question":"1000-500 =
    _____", "op":["500","501","502","503"],"ans":"500","attempt":null},
    {"question":"5+5 =
    _____", "op":["10","55","25","50"],"ans":"10","attempt":null},
    {"question":"10 + 90 =
    _____", "op":["1090","19","90","100"],"ans":"100","attempt":null},
    {"question":"30+ 70 =
    _____", "op":["100","101","10","10"],"ans":"100","attempt":null},
    {"question":"1000-1=
    _____", "op":["999","1001","1000","100001"],"ans":"999","attempt":null},
    {"question":"1000/1=
    _____", "op":["100","1001","1000","101"],"ans":"1000","attempt":null},
    {"question":"1000*1=
    _____", "op":["100","1001","1000","10001"],"ans":"1000","attempt":null},
    {"question":"100+1=
    _____", "op":["101","1001","1000","100001"],"ans":"101","attempt":null},
    {"question":"10+1=
    _____", "op":["11","1001","1000","101"],"ans":"11","attempt":null}
  ]

  checkAnswer(selected:NgForm){

    console.log(selected.value);
    for(let x=0;x<10;x++){

      let ans1=this.questions[x].ans;
      this.questions[x].attempt=selected.value[x+1];
      if(selected.value[x+1]==ans1){

        this.correctAns++;
      }
      else{
        this.incorrectAns++;
      }
    }
    this.flag=true;

  }

  Review(){
    this.submitted=true;
  }
}

```


Exam.component

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

import { ExamComponent } from './exam.component';

describe('ExamComponent', () => {
  let component: ExamComponent;
  let fixture: ComponentFixture<ExamComponent>;

  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [ ExamComponent ]
    })
    .compileComponents();
  });

  beforeEach(() => {
    fixture = TestBed.createComponent(ExamComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });
});
```

Exam Component.html

```
<div class="container ">
  <h1 class="text-center p-3 mb-2 bg-success text-white">Basic MCQ
  Test</h1>
  <form #selected="ngForm" (ngSubmit)="checkAnswer(selected)">
    <div *ngFor="let q of questions;let i=index">
      <p>{{i+1}} ) {{q.question}}</p>
      <div *ngFor="let o of q.op">
```

```

        <input type="radio" name="{{i+1}}" value="{{o}}" ngModel>{{o}}
    </div><br>
</div>
<input type="submit" class="btn btn-success" value="submit">
</form>
<br>
<div class="card text-center" *ngIf="flag">
    <div class="card-header">
        EXAM RESULT
    </div>
    <div class="card-body">
        <h5 class="card-title">Correct ANSWER={{correctAns}}</h5>
        <h5 class="card-title">Incorrect ANSWER={{incorrectAns}}</h5>
        <input type="button" value="Review" (click)="Review()" class="btn
btn-primary" ><br><br>
    </div>

</div>
<br><br>

<div *ngIf="submitted">
    <h6>Review the correct Answer of all the question and your attempts!</h6>
    <table border="2" class="table table-striped text-center">
        <thead class="thead-dark">
            <tr>
                <!-- <th scope="col">#</th> -->
                <th scope="col">Question</th>
                <th scope="col">CorrectAns</th>
                <th scope="col">Attempted</th>
            </tr>
        </thead>
        <tbody>
            <!-- <div *ngFor="let n of questions;">
                <tr>
                    <td></td>
                    <td>{{n.ans}}</td>
                    <td>{{n.attempt}}</td>
                </tr>

            </div> -->
            <tr *ngFor="let n of questions;">

                <td>{{n.question}}</td>
                <td>{{n.ans}}</td>
                <td>{{n.attempt}}</td>
            </tr>
        </tbody>
    </table>
</div>

```

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

4. Sprint Work.

1. Sprint:1:

- Created Project Folder .
- Added to Visual Studio.
- Created Entity.

5. Sprint:2:

- Created index.html.
- Created main file.
- Created HTML files.

6. Sprint:3:

- Created property files.
- Created OnlineTestApplication angular.json.
- Pushed the code to the repository.

ThankYou.