**Angular code**

1. app.component.html

<router-outlet></router-outlet>

1. app.component.ts

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

@Component({

selector: 'app-root',

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

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

})

export class AppComponent {

title = 'onlineapp';

}

1. loginuser.component.html

<div class="container-fluid">

<form [formGroup]="loginForm">

<div class="login\_container">

<h4 class="text-center mb-2">Online Assessment</h4>

<div class="input-group mb-3">

<div class="input-group-text">

<i class="fa fa-user" aria-hidden="true"></i>

</div>

<input type="text" class="form-control" placeholder="Username" formControlName="username" aria-label="Text input with checkbox">

</div>

<div class="input-group mb-3">

<div class="input-group-text">

<i class="fa fa-key" aria-hidden="true"></i>

</div>

<input type="password" class="form-control" formControlName="password" placeholder="Password"

aria-label="Text input with radio button">

</div>

<div class="text-warning mb-2" \*ngIf="isLoginClicked && !service.loggedIn">

Incorrect username or password

</div>

<div class="form-group text-center">

<button type="button" class="btn btn-primary mx-2" (click)="loginUser($event)">Login</button>

</div>

</div>

</form>

</div>

1. loginuser.component.ts

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

import { FormControl, FormGroup, Validators } from '@angular/forms';

import { Router } from '@angular/router';

import { CommondataService } from 'src/app/services/commondata.service';

@Component({

selector: 'app-loginuser',

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

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

})

export class LoginuserComponent implements OnInit {

username:any;

password:any;

isLoginClicked:boolean = false;

loginForm = new FormGroup({

username: new FormControl('', Validators.required),

password: new FormControl('', Validators.required),

});

constructor(public service:CommondataService, private router:Router) { }

loginUser(event:any){

this.isLoginClicked = true;

this.username = this.loginForm.value.username;

this.password = this.loginForm.value.password;

if(this.username === 'test' && this.password === 'password'){

this.service.loggedIn = true;

}else{

this.service.loggedIn = false;

}

if(this.service.loggedIn)

{

this.router.navigate(['instructions'])

}

}

signUpUser(event:any){

console.log("Sign Up screen to be done as well.")

}

ngOnInit(): void {

}

}

1. Loginuser.component.css

.login\_container{

position: absolute;

top: 50%; right: 50%;

transform: translate(50%,-50%);

background: blue;

padding: 2em;

}

.container-fluid{

position: absolute;

height:100%;

width:100%;

background:linear-gradient( to left top, blue, red);

}

6.instructions.component.html

<div class="container-fluid">

<div class="instructions-container">

<h3>Instructions to be followed by candidate before taking the exams.</h3>

<ul>

<li>The exam is monitored by the proctor online using the webcam.</li>

<li>The participant should not navigate away from the website.</li>

<li>The participant should not move away from the webcam as well.</li>

<li>In case of genuine reasons due to which the site gets navigated, please contact the support team.</li>

<li>On pressing the proceed button, the questions will be displayed and the participant has 50 minutes to complete the assessment.</li>

<li>On pressing the quit button, the participant will be redirected to the login page and the assessment attempt is not lost.</li>

</ul>

<div class="form-group text-center">

<button type="button" class="btn btn-primary mx-2" (click)="proceed($event)">Proceed</button>

<button type="button" class="btn btn-danger mx-2" (click)="quit($event)">Quit</button>

</div>

</div>

</div>

7. Instructions.component.ts

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

import { Router } from '@angular/router';

@Component({

selector: 'app-instructions',

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

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

})

export class InstructionsComponent implements OnInit{

constructor(private router:Router){}

ngOnInit(): void {

}

proceed(event:any){

this.router.navigate(['assessment']);

}

quit(event:any){

this.router.navigate(['login']);

}

}

8. instructions.component.css

.instructions-container{

z-index:1;

width:60%;

height:30%;

padding: 1em;

position: fixed;

top: 10%;

left: 20%;

background: orange;

border: 1px solid blue;

}

.container-fluid{

z-index:0;

position:fixed;

background: grey;

width: 100%;

height: 100%;

}

9. Assessment.component.html

<div class="container-fluid">

<div class="assessment-container">

<form [formGroup]="assessmentForm">

<div [formArrayName]="'formArrayName'">

<h4 class="font-weight-bold text-center">Assessment</h4>

<div class="question\_answer" \*ngFor="let item of currentQuestionAnswers, let i=index;">

<div [formGroupName]="i">

<p class="questions">{{item.q}}</p>

<div class="form-check">

<input class="form-check-input" type="radio" id="answer"

formControlName="answer" value="{{item.optiona}}">

<label class="form-check-label" for="answer">

{{item.optiona}}

</label>

</div>

<div class="form-check">

<input class="form-check-input" type="radio" id="answer" value="{{item.optionb}}"

formControlName="answer">

<label class="form-check-label" for="answer">

{{item.optionb}}

</label>

</div>

<div class="form-check">

<input class="form-check-input" type="radio" id="answer" value="{{item.optionc}}"

formControlName="answer">

<label class="form-check-label" for="answer">

{{item.optionc}}

</label>

</div>

<div class="form-check">

<input class="form-check-input" type="radio" id="answer" value="{{item.optiond}}"

formControlName="answer">

<label class="form-check-label" for="answer">

{{item.optiond}}

</label>

</div>

</div>

</div>

</div>

<div class="form-group button-align">

<button type="button" class="btn btn-danger mx-2"

(click)="submit($event)">Submit</button>

</div>

</form>

</div>

</div>

10. assessment.component.ts

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

import { FormArray, FormBuilder, FormControl, FormGroup, Validators } from '@angular/forms';

import { Router } from '@angular/router';

import { AnswerModel } from 'src/app/models/model';

import { ApiService } from 'src/app/services/api.service';

@Component({

selector: 'app-assessment',

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

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

})

export class AssessmentComponent implements OnInit {

assessmentForm: FormGroup;

constructor(private router: Router, private formBuilder: FormBuilder, private api:ApiService) {

this.assessmentForm = new FormGroup({

formArrayName: this.formBuilder.array([])

})

}

indexFlag: boolean = false;

index: number = 0;

currentSet: number = 0;

answerOptions: any = [];

currentQuestionAnswers: any = [];

lastFlag: boolean = false;

main: any = {}

questions: any[]=[];

answerObj:AnswerModel = new AnswerModel();

buildForm() {

const controlArray = this.assessmentForm.get('formArrayName') as FormArray;

Object.keys(this.currentQuestionAnswers).forEach((i) => {

controlArray.push(

this.formBuilder.group({

answer: new FormControl({ value: this.currentQuestionAnswers[i].answer })

})

)

})

}

setQuestionsAnswers() {

let a: any[] = [];

this.questions.forEach((item: any) => {

a.push(item)

})

this.currentQuestionAnswers = a;

this.buildForm();

}

submit(event: any) {

var a:any = [];

this.assessmentForm.value.formArrayName.forEach((item:any)=>{

a.push(item.answer);

})

this.answerObj.answer1 = a[0];

this.answerObj.answer2 = a[1];

this.answerObj.answer3 = a[2];

this.answerObj.answer4 = a[3];

this.answerObj.answer5 = a[4];

this.answerObj.answer6 = a[5];

this.answerObj.answer7 = a[6];

this.answerObj.answer8 = a[7];

this.answerObj.answer9 = a[8];

this.answerObj.answer10 = a[9];

console.log(this.answerObj);

this.api.submitForm(this.answerObj).subscribe((res:any)=>{

console.log(res);

})

this.router.navigate(['completed']);

}

getInitialData(){

this.api.getForm().subscribe((res:any)=>{

this.questions=res;

this.setQuestionsAnswers();

})

}

ngOnInit(): void {

this.getInitialData();

}

}

11. Assessment.component.css

.assessment-container{

width:100%;

padding: 1em;

background-color: lightgray;

overflow-y: visible;

}

.questions{

font-weight: bold;

}

.question\_answer{

padding: 1em;

}

.button-align{

position: fixed;

right: 5%;

bottom: 2%;

}

.container-fluid{

margin:0em;

padding:0em;

}

12. completedassessment.component.html

<div class="container-fluid">

<div class="completed-container">

<h3>Completed</h3>

<p>You have submitted the assessment for evaluation. The recruitment team will contact you if you have cleared the assessment.</p>

</div>

</div>

13.completedassessment.component.ts

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

import { Router } from '@angular/router';

import { interval } from 'rxjs';

@Component({

selector: 'app-completedassessment',

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

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

})

export class CompletedassessmentComponent implements OnInit {

constructor(private router:Router){}

ngOnInit(): void {

setTimeout((interval:any)=>{

this.router.navigate([''])

},5000);

}

}

14. completedassessment.component.css

.completed-container{

z-index:1;

width:50%;

height:10%;

padding: 1em;

position: fixed;

top: 10%;

left: 20%;

background: orange;

border: 1px solid blue;

}

.container-fluid{

z-index:0;

position:fixed;

background: grey;

width: 100%;

height: 100%;

}

15. model.ts

export class AnswerModel{

answer1:string='';

answer2:string='';

answer3:string='';

answer4:string='';

answer5:string='';

answer6:string='';

answer7:string='';

answer8:string='';

answer9:string='';

answer10:string='';

}

16. api.service.ts

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

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

@Injectable({

providedIn: 'root'

})

export class ApiService {

url:string='http://localhost:3000/form/';

constructor(private http:HttpClient) { }

getForm(){

return this.http.get(this.url);

}

submitForm(req:any){

return this.http.post(this.url,req, {responseType: 'text'})

}

}

17. commondata.service.ts

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

@Injectable({

providedIn: 'root'

})

export class CommondataService {

loggedIn:boolean = false;

constructor() { }

}

18. app-routing.module.ts

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

import { RouterModule, Routes } from '@angular/router';

import { AssessmentComponent } from './components/assessment/assessment.component';

import { CompletedassessmentComponent } from './components/completedassessment/completedassessment.component';

import { InstructionsComponent } from './components/instructions/instructions.component';

import { LoginuserComponent } from './components/loginuser/loginuser.component';

const routes: Routes = [

{path:'login', component:LoginuserComponent},

{path:'instructions', component:InstructionsComponent},

{path:'assessment', component:AssessmentComponent},

{path:'completed', component:CompletedassessmentComponent},

{path: '', redirectTo: '/login', pathMatch: 'full'}];

@NgModule({

imports: [RouterModule.forRoot(routes)],

exports: [RouterModule]

})

export class AppRoutingModule { }

19. app.module.ts

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

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

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

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

import { AppRoutingModule } from './app-routing.module';

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

import { LoginuserComponent } from './components/loginuser/loginuser.component';

import { InstructionsComponent } from './components/instructions/instructions.component';

import { AssessmentComponent } from './components/assessment/assessment.component';

import { CompletedassessmentComponent } from './components/completedassessment/completedassessment.component';

@NgModule({

declarations: [

AppComponent,

LoginuserComponent,

InstructionsComponent,

AssessmentComponent,

CompletedassessmentComponent

],

imports: [

BrowserModule,

AppRoutingModule,

FormsModule,

ReactiveFormsModule,

HttpClientModule

],

providers: [],

bootstrap: [AppComponent]

})

export class AppModule { }

20. proxy.config.json

{

"/form":{

"target":"http://localhost:3000/form",

"secure":"false",

"changeOrigin":true,

"logLevel":"debug"

}

}

**NODE JS**

1. Form.route.js

const express = require('express');

const FormController = require('../Controllers/Form.Controller');

const router = express.Router();

//Get a form by Id

router.get('/', FormController.getForm);

//save a new form

router.post('/', FormController.submitForm);

module.exports = router;

1. App.js

const express = require('express');

const createError = require('http-errors');

const cors = require('cors');

// const dotenv = require('dotenv').config();

const app = express();

app.use(cors());

// app.use(express.json());

// app.use(express.urlencoded({ extended: true }));

// Initialize DB

// require('./initDB')();

const FormRoute = require('./Routes/Form.route');

app.use('/form', FormRoute);

//404 handler and pass to error handler

app.use((req, res, next) => {

/\*

const err = new Error('Not found');

err.status = 404;

next(err);

\*/

// You can use the above code if your not using the http-errors module

next(createError(404, 'Not found'));

});

//Error handler

app.use((err, req, res, next) => {

res.status(err.status || 500);

res.send({

error: {

status: err.status || 500,

message: err.message

}

});

});

const PORT = process.env.PORT || 3000;

app.listen(PORT, () => {

console.log('Server started on port ' + PORT + '...');

});

1. form.model.js

const mongoose = require('mongoose');

const Schema = mongoose.Schema;

const FormSchema = new Schema({

name: {

type: String,

required: true

},

price: {

type: Number,

required: true

}

});

const Product = mongoose.model('form', FormSchema);

module.exports = Product;

1. Form.controller.js

const createError = require('http-errors');

const mongoose = require('mongoose');

const Form = require('../Models/Form.model');

module.exports = {

getForm: async (req, res, next) => {

try {

const result = [

{ id: 0, q: '1. Who is the first Nobel Prize winner?', optiona: 'Madame Querie', optionb: 'Issac Newton', optionc: 'Albert Einstein', optiond: 'Thomas Edison', set: 0, lastSet: false },

{ id: 1, q: '2. Who is the first Football World Cup Winner?', optiona: 'Brazil', optionb: 'Argentina', optionc: 'Uruguay', optiond: 'France', set: 0, lastSet: false },

{ id: 2, q: '3. Who is the owner of Tesla?', optiona: 'Elon Musk', optionb: 'Mark Zuckerburg', optionc: 'Sunil Mittal', optiond: 'Mukhesh Ambani', set: 0, lastSet: false },

{ id: 3, q: '4. Who invented pencillin?', optiona: 'Alexander the Great', optionb: 'Alexander Graham Bell', optionc: 'Alexander Fleming', optiond: 'Alfred Nobel', set: 0, lastSet: false },

{ id: 4, q: '5. Who is the first Nobel Prize winner?', optiona: 'Frédéric Passy', optionb: 'Issac Newton', optionc: 'Madame Querie', optiond: 'John Clauser', set: 0, lastSet: false },

{ id: 5, q: '6. Who is the president of United States?', optiona: 'Joe Biden', optionb: 'Donald Trump', optionc: 'Barack Obama', optiond: 'Bill Gates', set: 1, lastSet: true },

{ id: 6, q: '7. Who is the current prime minister of India?', optiona: 'Rahul Gandhi', optionb: 'Dr. Manmohan Singh', optionc: 'Narendra Modi', optiond: 'Dr. APJ Abdul Kalam', set: 1, lastSet: true },

{ id: 7, q: '8. Where is the 2022 FIFA world cup held?', optiona: 'Brazil', optionb: 'Argentina', optionc: 'Qatar', optiond: 'France', set: 1, lastSet: true },

{ id: 8, q: '9. Who won the 2022 T20 World Cup', optiona: 'India', optionb: 'Pakistan', optionc: 'England', optiond: 'New Zealand',set:1, lastSet: true },

{ id: 9, q: '10. Who scored the highest runs in 2022 T20 World Cup', optiona: 'Jos Buttler', optionb: 'Surya Kumar Yadav', optionc: 'Virat Kohli', optiond: 'Mohammad Rizwan', set: 1, lastSet: true }

];

res.send(result);

} catch (error) {

console.log(error.message);

}

},

submitForm: async (req, res, next) => {

try {

// const form = new Product(req.body);

res.send('Success');

} catch (error) {

console.log(error.message);

next(error);

}

}

};