**EXAM PORTAL**

Developed by Shivam Rai 0157CS191167 (LNCT’S)

A online portal which provides the platform for testing the student`s knowledge though MCQ based Quizzes and generating the report card.

***Technologies Used:***

Frontend : Angular , Html , Css , angular-material

Backend: Springboot(java), API , Jwt-Authentication,MySQL

Tools Used : VSCode , Intellij, MySQL workbench , POSTMAN

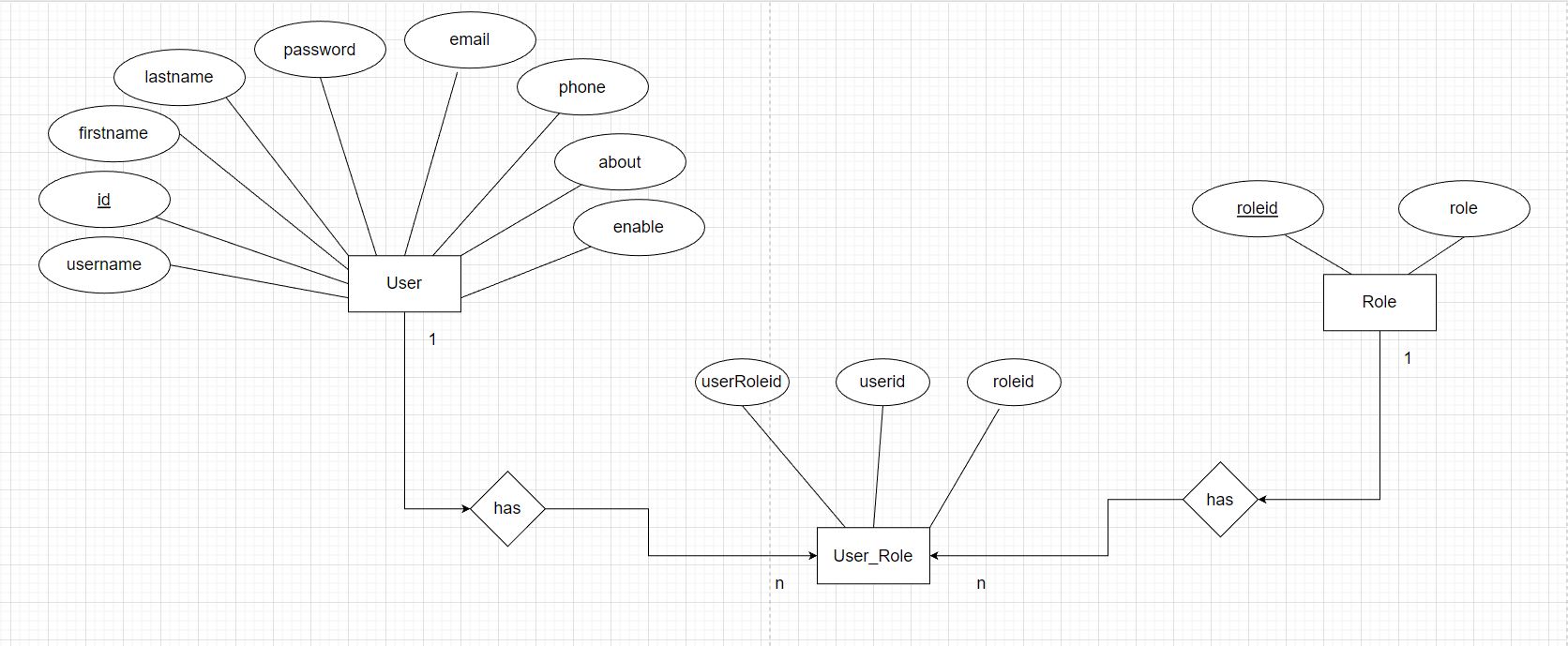
**Theme:**

The project basically consists of two main modules Admin and the User. The admin dashboard has the facility of creating the various categories and add the Quizzes in them, also add the Question to different Quizzes created by him.

The User Dashboard is containing the Interface where different categories quizzes are shown to him and he can attempt the Quizzes any no of times. Each Quizzes is bounded by the timer and auto-submitted as soon timer goes off.

Each of the Entries of the User / Admin is being save to the database developed using MySQL . The data of the Quizzes , categories is being saved to database in the tablular format.

1. Diagram of User, Role and User Role classes :



# DESCRIPTION STARTS :

# 

Refer Screen Shot / Video for introduction :

Frontend with angular

Backend with Spring Boot

Our approch is to desing the APIs and call them from the Frontend.

First create the Api for the normal user then go to the other sections.

# STEP 3 : CREATING the entity classes "1.User" & "2.Roles" , then using the "3.User\_Roles" class to maintain the many to many relationship.

//A single user can have multiple user roles so using set to store them.

Also make getter setter of set.

@OneToMany(cascade = CascadeType.ALL, fetch = FetchType.EAGER, mappedBy = "user")

**cascade-means** when we save/delete user changes are reflected in userRole table also.

**fetch-means,** when user is fetched its roles are also fetched.

**mappedby-user** means : table created 1 field named "user" to map two tables .

@JSONIgnore is written to avoid circular dependency , continous table linkage.

Similary create the relationship between the Role and UserRole tables.

JPA is a specification and the Hibernate is the implementation of it.

#STEP 4 : CREATE SERVICE TO SET USER TO THE DATABASE.

#########################

#Video 5 :

Now to save the the data in the DB creating the Repository interfaces.

After that creating the services package and make interfaces to save the data to database , and implementing their classes (they uses the methods of repositories )

Create new package impl\_classes , now implement the methods declared in interfaces to save the user and the user Role.

In main running class , we can use the CommandLineRunner Interface to check the program . and crate users etc

Fake data is saved into database and role table also updated.

#########################

#Video 6 :

Now we need to create the controllers so we can access the services through api's

//STEP 7 is creating the handlers method . in Controllers

Creating the end points in the controller.

THE BASIC DESING PATTERN IS FOLLOWED , IT

* Creating the Controllers ( has handlers methods )
* service interface variable is auto wired.
* Implementation class ob. is auto wired
* finally repository variable are auto wired to perform the task on the database.

CREATING THE APIs with the help of handlers methods.

We will integrating the concept of the Authentication later , after making small part of Fronted in Angular.

IMPORTANT\*

**HOW TO RESTORE THE ORIGINAL PROJECT in new Machine .Extract the saved .zar file of drive.**

In order to restore the project

Backend :

1. Install the my-sql 64 bit installer(full version refer durgesh video ) and gui(workbench) .

2. Create the database “exam”.

3. Reload the full project - In maven categories(Intelli-j)

Frontend:

Install the node (npm)

Install the angular cli

Create the new frontend project using the cmd (refer steps of Video durgesh) , then copy the files of old

In Vs code run the ng serve at frontend project.

To solve ERROR of VS Code not running the Ng serve Command

Video 25:

To solve the error of VS code not running the ng commands just open the Powershell and run the command and press Y

set-ExecutionPolicy RemoteSigned -Scope CurrentUser

\*Traditional / Genral way to creataing the new frontend project is to install the node npm in computer , then intall the angular cli in particular folder and then finally make the project and then run ng serve command

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_-

**Video 7: STARTING WITH FRONTEND**

We are using the Angular to make call to end points of API that we have made in springboot.

DONE ALREADY .

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Follow the basic details to install the angular cli :

**Install nodejs , set path , npm install -g @angular/cli**

Open cmd and check node -v , npm -v , etc. commands (to check user path var. are correctly set or not . )

Since we have already all thing installed . No worries.

DIRECTLY CHECKING VERSION OF ANGULAR CLI

**ng version**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**1.Now open the location of the project where you created the backend EXAMSERVER project.**

**2.Now open the cmd to create the angular project.**

***ng new projectname***( in our case “examfront “)

Choose yes to angular routing, CSS

Now folder is created .

**3.Now cd examfront and run the project**

**ng serve** -- command to run the project of angular.

Now angular project is live on port 4200.

**4.now open examfront folder in VS CODE**

Now we can edit content in app.component.html , delete previous content of it.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Now

We are adding the ANGULAR MATERIAL IN OUR PROJECT. so we can use navbar , buttons , and other components etc .

**ng add @angular/material**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

now we can close the cmd and open the terminal in VS code.

**#USAGE of Angular material.**

Now to use any component take help of the Angular material website and first import the module of it and them add it in imports[] array in app.module.ts file and then use code of buttons etc in the .html file.

**#attaching the bootstrap grid in angular (refer durgesh video also )**

Now we want to use the Bootstrap grid in angular so install it in project folder from terminal and then set the in angular.json file ctrl+P and add the path of grid-only css here.

**Google grid only css** from github link npm install.....s apply it terminal.

1.Apply command in terminal of project

**npm install bootstrap-grid-only-css --save**

No need to use any link , just paste the path in angular.json file.(Press Ctrl + P in VS Code and search angular.json)

Now open the angular.json file and in styles [ set the path here ]

//Set the path for the "./node\_modules/bootstrap-grid-only-css/dist/bootstrap-grid.min.css"

As downloaded bootstrap css are saved over here.

ie

*"styles"*: [

              "./node\_modules/@angular/material/prebuilt-themes/indigo-pink.css",

              "src/styles.css",

              "./node\_modules/bootstrap-grid-only-css/dist/css/bootstrap-grid.min.css"

            ],

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***VIDEO 8:***

Now we are designing the form to create the normal user.

Now we are going to create the component “signup” inside the folders so it can be reused when needed.

Headers, footers , etc are placed in **components** folder

Signup,login page,admin,user,home,profile, components are placed in **pages** folder.

to create the component use---ng g c navbar.

To show/use any component

* To add the signup component , we need to first create it.
* So go inside the src/app/pages/ using terminal.
* Create the component ng g c signup
* Now add the component and its path in “app.routing.ts” file.
* Now use this component anywhere in html. files

We need to add the object { path , component : , pathmatcher } in Routes [] in app-routing.module.ts file to use any component imported from the angular material website.

To fire the particular page/signup we use <router-outlet> tag in html file of base file app.component.html provided to show particular component.

now fire the /localhost:4200/signup in browser.

Now design the basic signup form using the html codes

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Video: 9**

Now we need to save data in database.

Angular data binding

* Binding the data to the empty local variable “user” declared in signup.component.ts file
* In each field add the [(ngModel)]=”user.firstname” and bind it with each of its field
* Now add the (ngSubmit)=formSubmit() event in form tag , and given definition of function in .ts file.
* Now we need to create the service in Angular app to call the backend API of springboot to save this user in database.
* Now open terminal in VS code and reach to /src/app/services/ and generate the new service
* Create a new helper.ts file . and give the base url as

*let* baseUrl="http://localhost:8080";

export default baseUrl;

* using command ng g s user
* Now open the .ts file of the created user service and design the method to add the user to database. Here basically using httpclient object we are calling the backend api handler method to add the new user.
* Now basically this userService object is used (injected) in signup.component.ts file of signup component and there it will use adduser(user) method to save the data in database.

//Content of user service

export *class* UserService {

*constructor*( private *http*:HttpClient)

{

}

public addUser(*user*:*any*)

{

return this.http.post(`${baseUrl}/user/`,*user*);

}

* Now come back to formSubmit() function and use the user( obj var of userSerive ) to call the addUser() function. Now the statement need to be subscribed and two results are used “data” and “error” . Data means task completed successfully and we can give the success message and error means something wrong in it.

  formSubmit()

  {

   console.log(this.user);

//STEP 11 b

// add user fuunction from the userSerivice component.

this.userService.addUser(this.user).subscribe(

(*data*:*any*)*=>*

{

// (data) means success , Since user is registereds we are emptying the fields only.

  this.user.username='';

  this.user.password='';

  this.user.firstname='';

  this.user.lastname='';

  this.user.email='';

  this.user.phone='';

  //success

  console.log(*data*);

  //alert('success');

  Swal.fire('Successfully Done ',' Registered User id : '+ *data*.id ,'success')

},

(*error*)*=>*

{

//error

console.log(*error*);

//alert('error');

this.snack.open('something went wrong','',{duration:3000} );

}

);

  }

* Also to give the message of success we can use Sweet Alert or MatSnackbar , both need to be imported from angular material and put them in app.module.ts file.

Now submit the form in formSubmit() function.

To make this submit button work , import the form module. in app.module file.

To bind the data , first create the vacant user.

The use of [(ngModel)] to bind the data to the user .for each of its fields eg username, pass

//STEPS :

#.Import the form module in app.module.ts

#.Apply the (ngSubmit) and a function in form tag , and give defination of fun in ts file.(done)

#.Include [(ngModel)] and name in the input field to bind the data.(done)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**#VIDEO 10 :** #9 series : Summarizing working with components

Send the binded data to the user.

1.Start the springboot / backend project from intellij to run always while working in VSCode.

2.Need to import httpClientModule here in app.module.ts .

Now we will make the seperate folder Services to call the APi related to the user.

Create a service component : ng g s user in the services

Create a new helper.ts file . and give the base url.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Now we will make the call to backend using Angular , for that add the @Crossorigin on controller to handle the different ports running .

Now in userService.ts file we make function of addUser

ALso import the variables in the constructor ,and then in fucntion use this variable to make request.

Now start the Angular server : ng server in examfront folder and fill the form .

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Video : 10** : MAtSnatchBAr importing and SweetAlert install

import the modules and show the status with help of beautiful messages.

Go to angular material copy the content from the imports and add it to app.moudle.ts and add same element to the imports[]

Steps to use

1.Import it in app.module.ts file

2.declare the variable in constructor(...) in ts file

3.Use them with codes.

TO add the validation in the fields we can use “required” in the input field .

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Video 13 -- 12

NavBar

Now design the navbar , but first import it then ( \* add the component in “NavbarComponent” in declaration:[] in app.module.ts file. )

Include it in app.component.html file . <app-navbar>. This will ensure the navbar Component to be shown every where.

Also see the @Component( selector: ’app-navbar’ ) to declare it in app.component.ts file . . <app-navbar>.

<!-- Navbar is static content -->

<app-navbar></app-navbar>

Now to make the buttons as link make <a> tags and give the href link to move.

but here new pages are loading to make changes on same page use <routerlink> instead of href

**[routerLink]=”/signup”** 🡪 it is used to for faster directing to other url

Also design the login to show the Login and logout button.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Video : 14 - 13

Close the Angular Application for now.

**SpringBoot work in adding security.**

Adding the JWT Authentication in Login for ADMIN And Normal Users.

**1.Add springboot starter security dependecy in pom.xml**

**2.Add Json Web token support for JVM dependecy and remove version from it(jjwt maven)**

**3.jaxb api (javax.xml.bind dependency )**

Reload maven and build it

Now rerun it and enter the password got(in run window of intellj ) in brower when accessing localhost.

**Now proceeding to make the configurations for the Spring Security**

Now implement the UserDetails in user class iteself and override the methods and then set the authorities method over here.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

JWT Authentication:

Now to apply the spring security , we need userDetails class over here.

So easy way is to implements the UserDetails itself in User class.

SO JUST ADD THE IMPLEMENTS in user class and click to override its all method

//VIDEO 14  
 //Methods to Override for UserDetails.Basically we need to set the Authority function over here.

@Entity  
@Table(name="users")  
public class User implements UserDetails {

.

. //other content of the user class

.

.

.

@Override  
 public boolean isAccountNonExpired() {  
 return true;  
 }  
  
 @Override  
 public boolean isAccountNonLocked() {  
 return true;  
 }  
  
 @Override  
 public boolean isCredentialsNonExpired() {  
 return true;  
 }  
  
  
  
 //VERY important Method ::::::::::::::::::::::  
 @Override  
 public Collection<? extends GrantedAuthority> getAuthorities() {  
  
  
 //STEP 12 a : create set of Authorities  
  
 Set<Authority> set = new HashSet<>();  
 // 12 c: Now we will set the roles in the collection , so we use the Set UserRoles to get userRole obj which intenally conatiains the roles available in Authorities.  
 // And we also keep on adding differnt roles in set using the Authority object.  
  
 this.userRoles.forEach( userRole -> {  
 set.add( new Authority(userRole.getRole().getRoleName()));  
 }  
 );  
  
  
////////////////////////////////////////////////  
 return set;  
 }  
  
  
}

Basically we need to set the Authority function over here.

ie we need to set the roles for ADMIN NORMAL USER etc .

So after overrinding all other methods with true , last method Granted Authority need to be over ridded . So created a new model named “authority” and its get Authority() function will be used internally by spring security.

In User class we have “UserRoles” set , which basically contains the role .

So use this ,

create the set of Authorities. and create the Authority class in Model and implement the GrantedAuthority.

getAuthority()

Basically here we have assigned the roles of user to set using Authority class and finally returned the set.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

STEp 13 :

Need two more classes JwtRequest and JWTResponse to set the username and password and sent the token etc.

// STEP 13 b:

**Now the spring security internally use the UserDetailsService interface to load the username, so we need to make its implementation class, in impl folder.**

**13.c** Create the new class UserDetailsImpl in the services/impl\_classes folder and override the methods .

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**STEP 14 : Now we are making package the Config class for Spring Configuration .**

//set the configuration using classes JwtAuthenticationEntryPoint , JwtAuthenticationFilter , JwtUtil , MySecurityConfig.

**1**.

The JwtAuthenticationFiilter class helps to fetch the jwtToken sent under the Authorization header using the substring function.

Now if token exits we are using the Utiltiy class to extract the username using the token and jwt class object.

**2.**

Now next time if we have the username , so we are going to validate the token of the Username with the help of UserDetailsServiceImpl class object.

Now perform coding operations to set the context in Security holder and finally after verification call the chain method filterchain(request,response).

JwtUtli : class copied from Google and some changes done(refer video)

JwtAuthenticationFilterPoint : used in configure method of MySecurityConfig class .

MySecurityConfig : This is the main class, which uses all other configuration classes to take decision to allow authentication on which urls .

Also Need to override the 2 method of configure ( one with obj of AuthenticationManagerBuilder and other with HttpSecurity ) and BcryptPasswordEncoder here.

Also using the .andMathers( .. ) to allow the direct access to the specific url eg generate token , user etc

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**//STEP 16 :**

**Create the controller named “AuthenticateController”**

generte the handler in controllers to handle /generate token. To call the handler method with username and password and generate the token from the POSTMAN app..

//Now at last of Video we got internal server error as bCrypt Password Encloder is used ,( so for testing purpose convert it into plain encloder later will revert this change to normal.)

Basically when localhost:8080/generate-token Url is fired from the POSTMAN with Attaching the json Request Obj (username and password ).

Then method generate Token of AuthenticateController handlers this call and first it call the authenticate function with username and password. (This function internally uses the authenticationManager object for internal implementation by spring , for this we need to declare the @bean of AuthenticationManager in MySecurityConfig class. )

After the authentication is successful the userDetailsService is used to create the object of UserDetails 🡪 to be further used by JwtUtil class to generate the Token and return it.

//IN this way the Token is generated from successful verification of username and password.

In order to generate the token , “First time username is “user” the password will be provided on Intellij Console.

Minimize the backend springboot project now.

/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

**TRICK : To navigate using the routerLink : using the obj of Router.**

**-Declare the object of Router in constructor of .ts file (private \_router : Router )**

**-then use this var. to show the redirecting url e.g.**

this.\_router.navigate(['user-dashboard/0']);

**Video: 16**

Now using the Angular to generate the token and login module.

Start VS-code and “ng serve” command in terminal to run it .

Also declare a variable loginData in login component and then use [(ngModel)] to bind the data.

Making a login service to maintain the roles of User

Now using angular to make request to handlers of generate token api and hence saving the token in local storage . Also after authentication we also save the user in local storage to avoid repetitive calls to the backend.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

VIDEO 17 --- 16

Now adding an API in the springboot.

Now ANGULAR INTERCEPTOR we need to use which attaches the token in every request made to backend so that API can be accessed.

Make a file in services to implement auth.interceptor.ts

Design the basis structure of the class and then a array authInterceptorProviders[] and also configure this in the app.moudle.ts file under the separate tag of providers[].

in providers[]:

DONE till 30 min Video 17- 16 series

We have configured till the connecting with the Admin Dashboard or User Dashboard.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Now we are generating the guard to secure the ADMIN and NORMAL pages access.**

Go to Services folder

ng g guard admin

And create the constructor and check if user is actually the ADMIN user or not.

Also need to set the Admin gaurd in the app.routing.module.ts file .

//By this , we cannot directly access the Admin url , as guard is added. and use router link redirect back to the same login page.

Now doing the conditional redendring in navbar component , to show logout button when user is logged in .

--------------------------------------------

IMP ::::

EACH TIME START THE both springboot project and Angular Project to see effective changes in Project.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Video 18 : Correcting the error , two times clicking the Login button.

This error came because app uses the data before its set in local storage.

As we are printing the username in navbar before its set to the localstorage.

Do not directly use the LoginService variable instead decalre the other variables to check

eg isLoggedIn .

WE are going to use loginStatusSubject var which infroms all its subscribes that user is logged in by using its next(true) fucntion. .

So what we need to do is , just as soon we verifies the user as ADMIN or NORMAL we also pass the next() function to it , in login componenet .ts files. And then subscribe this status in navbar componenet .

So Navbar component can print the username now easily.

This solves the problem of lagging in Login button.

Switching the RefreshRate to 60Hz solved the problem of Screen Flickering.(It was Screeen issue service center man , replaced the motherboard and screen of my laptop)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Video 18 : BCryptPasswordEncoder ....

go to myconfig class and change the type of Password En.

Now we will encode the password at the time of sign up in userController

Delete the old entries from workbench , First remove roles then users .

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

VIDEO 19 : Correcting the UserNotFoundException then working on the ADMIN PROFILE

Create the helper folder and do the repective coding .

#WORKING ON CREATING THE ADMIN PROFILE.

import the list in app.module.ts to use in admin pages.

To include any component in other html page just also include it. <><>

use outer-routlet to display the page when link button is clicked.

Also load it in app.routing.ts file. Provide the CHildern of AdminDashboard to load the profile

Basically loading the profile component on Admin\_dashboard componenet we need to provide the childern [] in routing.module.ts file.

---------------------------------------------------------------

VIDEO 20 :

Desing the layout of the profile page.

Now fetech the details of the current login user from local session.

Now in login.ts file , fetch the details of the user and then display them.

We take user=null in .ts file , then we declare the variable of LoginService in contructor , then finally use it in ngOnInit method.

<!-- To check that null values are not present in phone field we use \*ngIf="user.phone" -->

-\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

VIDEO 21 : Adding more option in Admin dashboard.

Add the features in sidebar component.

**Now working on the categories of quizes , using the backend application of springboot.**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Video 22 : Date : 07/12/22

Make new folder exams under the models .

Made the seperate classes for the different entities like the Quiz, Questions , Category with there contructors and the getters and setters .

Now establish the relationship

@OnetoMany

1 Category can have many quizes ,

-> So take LinkedHashSet(mantain inlet order ) variable in the Category class.

at the same time take here

@ManytoOne in the quiz class by declaring the variable named Category .

Also using the jsonIgnore to avoid the continous cascading .

Similarly develop relationship for the Quiz and questions

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

VIDEO 23 :

API CREATION:

Now making the API /END Points for the following accessing categories , quizzes and Questions.

Then we can use this Api from the Angular.

Need POSTMAN (160 MB) TO TEST THE API

We needed Repositories to perform the operations and then their impl classes.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Now we are following the order

Repositories ->>> Services (decl of methods) --->> Impl classes

Now 1 declare the all methods of the categoryService and then finally also make its implementaion class and autowire the CategoryRepository variable then override all the declared methods.

In findAll categories method we use : typecasting into the hashset

return new HashSet<> (this .quizRepository.findAll());

In finding the single object we use the .get() fucntional

return this.quizRepository.findById(quizId).get();

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Similarly follow the pattern for the Questions class make its

Interface , then service interface (declare methods here ) and then the implmentation class (give defination here) .

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\*\*\*\*\*\*\*\*\*\*\* IMP

If we want to use some function which does not have the basic implementation .

So just declare the method as it in the REpository as in case of

FindByQuiz method in the QuestionRepository

The springboot itself gives the implementation of the cutomfindermethods

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

NOW THE WORK IS TO TEST THESE WITH THE HELP OF api BY DESINGNING A cONTROLLER

mAKE Controllers like Category Contoller ,

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Now ARCHITECTURE IS LIKe

Controllers(contains API) has the var of ( Services interfaces ) --> which have implemtation in Impl classes ---> which uses Repository var to execute the action.

Just create the fucntional Handler method and respected API to access them fluently.

//add category.

//update category.

// get category .

// get all categories.

// delete categopries.

To give the defination of the handler method we can use either the Category return type itself or we can use the ResponseEntity<?>

ALSO add the @CrossOrigin("\*") above the Category Controller , which container the various handlers method to access the API /category/categoryId etc

Also rememeber to Add @Service annotation to the Service Implementaion classes.

Now check all the APi using the POSTMAN

Be sure to create the token using Post Request. :

ie first access the localhost:8080/generate-token and pass the "password" and "username" in json raw format

Now copy the token and pass it in header of Authorization , in Bearer asdfasdfasdflkjlkjoijj5454dfsdf

And then we can make the request using the "Get" ie

localhost:8080/category/ --> fetch all categories

--> To add a category , just pass the data to raw underr Body label and in json format.

{

"title":"Programming Lang" ,

"description" : "This is very nice quiz on c++"

}

Now just make the POST request on localhost/category/

--> to update pass the old cid with new data and in Put mapping.

HEnce after Testing with the POSTMAN we came to conclusion that all the API are working fine , Later we will call them with Angular forntend.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

VIDEO 24 :

// Making the Quiz Controller and makeing various API same as the Category .

\*\*\*\*\*\*

To solve the problem of Not deleting from the Database in single api call , we have changed the type to LAZY in the Quiz model class while mapping them in @ManyToOne

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

If Error is comming like disble the seriazable inter.....

Just making mistake with mapping @ManyToOne(fetch=fetchType.LAZY)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

VIDEO 25 :

How to Start tesing on the POSTMAN :

1. Generate the token by passing the username and password in json

Make POST request to

localhost:8080/generate-token/ in body section pass {"username":"Shivamrai714@" , "password":"Shivamrai714@" } as object.

2. Copy the token and put it in Header section in key value pair of.

key - Authorization

value - Bearer copiedtokenikjlk;jsdlkfjal

3.Now make request to the api for testing with this token.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Object of Quiz : in JSON , used in POSTMAN while tesing the APIs

{

"qId": 45,

"title": "Intermediate with ML",

"description": "Grow yourself of ML",

"maxMarks": "200",

"numberOfQuestions": "10",

"active": false,

"category": {

"cid": 42,

"title": "Machine Learnings",

"description": "Robot programming "

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Format of the Quiz

{

"quesId": 47,

"content": "What is full forrm of ML",

"image": "java.png",

"option1": "Machine Learning",

"option2": "Mo Li",

"option3": "Many Lines",

"option4": "Must Lie",

"answer": "Machine Learning",

"quiz": {

"qId": 44,

"title": null,

"description": null,

"maxMarks": null,

"numberOfQuestions": null,

"active": false,

"category": null

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Video 24:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Error comming in the Deleting the Quiz in two times.

UPDATE: REMOVED FETCH TYPE AS EAGER in Category . AS ERROR WAS COMMING IN DELETEING QUIZ. (it was deleting in two times)

As both type Quiz and Category , fetch type was eager so hibernate save data, and does not delete it in one time

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Video 25:

1.

To solve the error of VS code not running the ng commands just open the Powershell and run the command and press Y

set-ExecutionPolicy RemoteSigned -Scope CurrentUser

2.

Some time error comes in loading the data from the backend due to CORS Policy although we have included the @CrossOrigin(\*) in our Controller method

Try these

.Start your backkend server

.Reset the browesr using clean up setting .

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Working on Frontend :

STEP 1 : create the Components

Just create the two components "view-categories" and "add-categories" ,

Also confiure them in app.routing.ts file , ie roiuterLink

Now just write the html to display the categories .

In view-category.ts file create the tempory category array[] , later we fetch the data from the backend.

Now we need to desing the appearence of the app using the mat card and display the data of category in the loops .

\_\_\_\_\_

STEP 2 : create the Services

Go to services folder and generate the services of add category.

Now open the ts file and declare the various variable in the constructor part of it.

We will design the function to load the data form back server

Using the httpClient var the data is loaded to categories using the handler method and api that we have created in springboot to access the database of Stored category

\_\_\_\_\_

STEP 3 : now to use this fetched data in the html page we need to inject categories in constructor of current page.

The variable will be of category service type.

Now functioning will take place in the ngOnit method where we need to subscribe the data which is comming ,

Need to give the suitable message on success and on error .

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Working on add-category component.

Just provide the routerLink to buttons to add new Category and in html page desing the basic structure to add it.

Now open the ts file of add-category and take a variable of category ={ title and description}

Now we need to bind it with html page using [ngModel] , so data entered will be fetched here

If any error in declaring the datatype in angular, just made it as :any type

To debug , if correct data is fecthed we can print it using

{{category|json}}

Now we will make a fuction to submit this form details to the Database using make a call the backend api of post method of add category

Also we will add this function in the formSubmit in ngSubmit() ...

Now after Successfull addition of the category , we need to reset the data in the field we can use

this.category.title=''

this.category.description=''

Or we can also redirect to the categories page

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Video 26 :

Now displaying the quizes over here ....

Make the componenet show quizes under the ADMIN "view-quizes"

Now add its link in app.routing.ts under the admin.

Also add the link to the sidebar Quizzes tab

Now we need to load the data from the database and show them

Same as diaplay category follow the same format.

Just Open the view-quizes .html

Now om .ts file create the dumy array of quizzes laterr vallues will com efrom the database

Any image we need to keep : just download it from web and put it in the assests folder then provide the link where ever you want to use it.

\*\*\*\*\*\*\* Error was comming in using the mat-card-action

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Now after setting the platfrom we can call the data from the backend and get the details publish over here

Just call the api of quiz , according to the quiz controller

AFTER setting up the frontend for showing the quiz data , just access the backend .

MAKE THE SERVICE TO ACCESS THE DATA:

1. Go to services folder and create the quiz service using

ng g s quiz .

Now work on the quiz.service.ts file

2. Declare the httpClient var in constructor , and then make a fun quizzes to call the backend api.

public quizzes(){

return this.\_http.get(`${baseUrl}/quiz/`);

}

3.Now to quiz.component.ts file

Declare the var of Service in constructor of it

Now in ngOnInit function we call the quizzes function and subscribe the results of it.

Then give the different results on the success and error

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

// ERROR IS COMMING IF WE EMPTY THE FULL DATA AND THEN title, description are not recognised

//ERROR : Due to Delete api of Quiz as data is not deleting in single call , Here data is also not displaying properly .So i have manully removed the null rows from the database.

To make it work

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Video 27 :

Adding a new Quiz to the Data base , by user.

Just fetch the details and bind it to ngModel then using the service save the data by calling the api of add-quiz.

Make the seperate component for the add-quiz.

Configure it in app.routing.ts file.

Make clickable buttons by making the routerLink="admin/add-quiz"

First open the sidebar html , then desing there clickable button.

Then under the View quiz , work on add quiz button using router Link

/admin/add-quiz/ will work here.

Now we need to desing the form to take the basic details of new quiz to be added such as titlt, no of Ques etc

Now to integrate the Angular material , eg using Slide Toggle to show the active status of Quiz.

We need to copy the api code of it and import it in the app.module.ts file

Also put the MatSlideToggle inside the imports array.

Now copy the html code of slide toggle and use it in project.

Now using the select module of anular material .

import the select module to show option in categories

and then finally put it app.module.ts file

Now we are using the select box of Select in a form , so creating the categories array , and then modifing it further.

Also keep the type of submit in button of Add

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Video 28:

Now we are going to bind the data using the ngModel , and later we will make form as ngSubmit on the function defined in .ts file

1 task is to load the categories form the database and show to the user in selecting the categories for a specific quiz

\_\_\_\_\_\_\_

General Method is to create the Service and make fucntion categories(){ and call to api of backend}

But Since we already have the category service and we can directyly use its object to access this method / function

So declare obj of CategorySerive in our add.quiz.ts file

This is how we have used this varible categories to loop over the variable categories fetched from the backend.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Bind the data with help of Model , so make a model in .ts file

and also check by printing it . {{quiz|json}}

now to Submit this data , we will make the ngSubmit=formSubmit() function in ts file.

//declare the model

categoryData={

title:'',

descrption:'',

maxMarks:'',

numberOfQuestions:'',

active:true,

category:null,

}

// bind in html page using ngModel

[(ngModel)]="categoryData.title"

//To bind the category field , we use its cid ,

[(ngModel)]="quizData.category.cid"

then loop through the categories array.

<mat-option \*ngFor="let c of categories" [value]="c.cid">

{{c.title}}

</mat-option>

If we want to additionally show any msg of error , the inlcude the component var in constructor.

eg MatSnackBar

Now we need to define the fucntion addQuiz in already created service file quiz.service.ts

also use http to make the post request to add the data to the base url and to /quiz/

We will also provide the obj of quiz in post request.

Now next , we need to use this service in our add.quiz .ts file and use this method of addQuiz() to subscrible the call made to backend , and handle two cases of success and failure.

Error in subsrcribe method at 27:45 Video28

Solved : I have not added the return statement of addQuiz() fun in quiz service file.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

DELETING THE QUIZ USING DELETE BUTTON , and correcting error of one time deletion

VIDEO 29 :

Desing the delete button in view quiz,

\* on clicking the button , call the function , use (click) event and bind it. Pass the q.qid of quiz in function deleteQuiz()

\* now define this function in .ts file of view Quiz , now using the quiz service object , call the function of deleting, which futher contains the call to delete api of the backend .

\*Now make a deleteQuiz fun in the Quiz.service . to call the backend API.

\*//delete Quiz

public deleteQuiz(qId:any)

{

return this.\_http.delete(`${baseUrl}/quiz`,qId );

}

Pass the base url using string interpolation .

This function is going to return the observable , which we are going to subscribe , when we call this delete fucntion in .ts file.

\* as we have already injected the QuizService , so direclty defining the deleteQuiz function.

UPDATE: REMOVED FETCH TYPE AS EAGER AS ERROR WAS COMMING IN DELETEING QUIZ. (it was deleting in two times)

Now data will be removed from the Database , but to work with the deletion in the frontend , we need to use the Filtering technique to show only current valid quizes

That we use filter func here , It is will filter/reserve the quiz whose id are not equal to id which is to be deleted.

ie id matching quiz will not be filtered out.

TO apply the Swal alert , we are going to use the then() function.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Video 30 : Updating Quiz Module:

Basically when we click the update button we are going to redirected to another page, also we are going to send the id of that quiz with url of it.

Make a new component in admin folder , of update - quiz.

That is we need to make the front page , to show the old details from the backend , then

We need to make the put mapping request , with quiz object sending with the old id.

\_\_\_\_\_\_\_\_\_\_\_

1.

Configure this component in app.routing.ts file

Under the admin array define this as subcategory

{

path:'quiz/:qid',

component:UpdateQuizComponent,

},

Ie to open the Updatecompnent we need to fire the path url

Now bind it with the Update button using the [routerLink]

[routerLink]="'/admin/quiz'+qId"

//In html code :::::

<button

[routerLink]="'/admin/quiz/'+ q.qId"

mat-raised-button color="primary"

class="ml20"

>

Update

</button>

\*\*\* Make sure to add the correct path with all /

\_\_\_\_-

NOW THE MAIN WORK To Fetch the data to show in update page

We are going to use ActivatedRoute , so using the snapshot.params to collect the Id which is coming in URL

Now we need to make a function in quiz service , to fetch the data of single quiz , to be displayed during showing details for update quiz.

Using call to backend api.

Basically we have fetched this id using the ActivatedRoute , and then using the id to make call to getQuiz function of service to get the data of Single quiz.

And then subscribing the values returned from the function.

\*\* Error come in only putting params.qid , so click the yellow bulb.

this.qId=this.\_route.snapshot.params['qid'];

\*\* If error come in datatype just choose any type.

\*\*\*\*\*\*\*\*\*\*\*\*\*

Not able to work with [ngValue]

So just used teh [value] only and binded with the cid only, that is on selecting the cid will be only changed but in database cid is capable of making the other changes.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Now we just need to save the changes in the database once the button is clicked , so binding click event, and using the quiz service fucntion to send the data using put request.

1. Step create the update method in quiz.service

2. Now create the function in .ts file to Handle the submission of this form. So just use the method and subscribe it.

Now after successful updatation we can navigate to other pages also.

Add

private \_router:Router in the router

NOW USING THE THEN FUnction to redirect to other page after clicking ok.

Using the then function() after the Swal.fire like this.

\*\*\*\*\*\*\*\* If Internet is switched off then , All icons will not work.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

If any error comes I have added a slide toggle with disabled property to show what is staus of publication.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Video 30 :

Displaying all the Questions of Quiz. (made a new method in backend for this.)

--> Now making new component to show quiz questions.

ng g c view-quiz-questions

--> Now setting up the url by adding the path in app.routing.ts file.

{

path:'view-questions/:id/:title',

component:ViewQuizQuestionsComponent,

},

--> Now make the property binding with the button , of Questions using [routerLink]

<button

[routerLink]="'/admin/view-quiz-questions/'+q.qId+'/'+q.title"

mat-flat-button color="warn">

Questions</button>

--> Now check by clicking on Question button , if data is traporting in url or not.

Now we are going to fetch these two values using the Activated Route

this.qId= this.\_route.snapshot.params['qid']; // putting data in local variable from the var of URL using Activated Route

this.qTitle= this.\_route.snapshot.params['title'];

/////////////////

Now we are going to show the Questions.

So create the question[] and we will also use service to load the data fromt the backend..

So making the service in services folder.

Error :

During taking the empty array of questions[ ] error was comming in using its elements .

So i have defined a dummy obj inside the array.

Reason unknown , it will not work if we empty the array

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Making a new function in backend to fetch all the question of the quiz, as current fun is fetcing only noOfQuestions .

//CREATED NEW FUNCTION VIDEO 30 : to get all question of Quiz , ie removing the filter

Basically we have created this detection from future perseptive as older method is used for user to show limited random ques , but admin has access to all the questions of Quiz

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

I have tried to sort the quizes but doesnt work

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

VIDEO 31

Createing ADD button :

Adding Questions to the Quiz

Basically we need to record the quiz id , and along with it we will share the data of Question with the database by creating another component.

After creating button , we are making the url to access it , also we will fetch the quid id from there.

-Now we will add the routerLink to provide this id in the url acesss.

\_\_\_\_\_

Defining the router link to access the url and id of quiz from where add quiz button is clicked.

\_\_\_\_\_

Defining the url call , under the admin component ,

{

path:'add-question/:qid',

component:AddQuestionComponent,

},

// Created the func to access the data from the url

\_\_\_\_\_\_

qId: any;

question={

quiz:{

qId:''

},

content:'',

option1:'',

option2:'',

option3:'',

option4:'',

answer:'',

};

constructor(private \_route:ActivatedRoute) { }

ngOnInit(): void {

this.qId = this.\_route.snapshot.params['qid'];

//The variable of the params.qid is should be same as varible used in url path in app.routing.ts file

console.log(this.qId);

this.question.quiz['qId']=this.qId; //basically binding the temporary url id with the local id variable to display in page

}

\_\_\_\_\_\_

Creating the form , and binding the values to the field using ngModel.

REMEMBER TO USE "name" and MatInput in filed to bind with ngModel

\_\_

TO show the title in currently adding quiz, we need to pass it in url from where the button is clicked to add question so go to app.routing.ts file and also add path:'add-question/:qid/:title' in it and later use it from Activated route varible .

Also we need to send the qTitle from the routerLink along with qId where button is clicked.

Overall need to send the variable of Quiz Service in routerLink , so we can fetch them to show on add Question page

Also we need to save data in qTitle using ActivatedRoute params.title in the ts file of Add question.

SO ALL THESE STEP ARE FOLLOWE TO TAKE THE ID AND TITLE IN URL AND FETCH THEM AND SHOW THEM.

\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Video 32 :

saving the data of Question to the database.

Refer to Code how to apply the model in mat-select .

<mat-form-field class="w100">

<mat-label>Select Answer</mat-label>

<mat-select [(ngModel)]="question.answer">

<mat-option \*ngIf="question.option1" [value]="question.option1" >{{question.option1}}</mat-option>

<mat-option \*ngIf="question.option2" [value]="question.option2" >{{question.option2}}</mat-option>

<mat-option \*ngIf="question.option3" [value]="question.option3" >{{question.option3}}</mat-option>

<mat-option \*ngIf="question.option4" [value]="question.option4" >{{question.option4}}</mat-option>

</mat-select>

</mat-form-field>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Now next thing is to refer to Question controller and then make the call to save the data of Question using call to service with post request.

Define the method to add Question in Question serice , using \_http client.

And then use this service to save data from .ts file. and subscribe the result and handle two cases of data and error

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

VIDEO 33 :

Deleting the Question :

So put the Delete and Update button in the card of view Questions and later bind the button with click event, which call the fucntion to delete which implicitly uses the service to call the delete api.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ERROR OF mat-card Action solved using the , importing the schemas statement in app.module.ts file , after the imports [] , and clicked on yellow bulb.

import { CUSTOM\_ELEMENTS\_SCHEMA, NgModule, NO\_ERRORS\_SCHEMA } from '@angular/core';

schemas: [ CUSTOM\_ELEMENTS\_SCHEMA ]

If Subscribe func does not come with cross line means , we have not added the retun statement in the function at the service.ts file.

ss

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Now desingning the mtehod to delete Question ,

Go to question service and add a deleteQuestion() to call the backend API of delete.

public deleteQuestion(questionId:any)

{

this.\_http.delete(`${baseUrl}/question/${questionId}`);

}

Now go to the ts file of question\_of\_Quiz.ts , and make a fucntion to call and subscribe the output of the service made.

Also bind this ts function with the click event.

<button

(click)="deleteQuestion(q.quesId)" // q.quesId - backend

mat-raised-button color="warn">

Delete

</button>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Now we need to define the deleteQuestion in the .ts file of it.

deleteQuestion(qId:any)

{

// alert(qId);

//Showing the alert of deleteing.

Swal.fire(

{

icon:'info',

showCancelButton:true,

confirmButtonText:'Delete',

title:'Are you sure to Delete Question ',

} ).then( (result)=>

{

if(result.isConfirmed)

{

//copied the code of delete funmction.

this.\_question.deleteQuestion(qId).subscribe(

(data)=>{

//success question deleted

//Using filter to delete the Quiestion from the fontend

this.questions=this.questions.filter((q)=> q.quesId != qId);

Swal.fire('Success',"Question Deleted",'success');

},

(error)=>{

Swal.fire('Error',"Error in Deleting Question",'error');

},

);

}//end of if

}

) //end of then

}

WE HAVE NOT IMPLEMENTED THE UPDATE QUESTION MODULE , LATER DO THAT JUST LIKE UPDATE QUIZ videos. lecture.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

VIDEO 34/35

INtegrating the rich text editor Here (Text field) .

To give the facility of Bold text, picture etc.

CK EDITOR

Google the ck editor angular intergration , now run the command to intall it using npm command in terminal.

Install in examfront folder- both .

npm install --save @ckeditor/ckeditor5-angular

npm install --save @ckeditor/ckeditor5-build-classic

now imports the CKEditor in app.module.ts file

import { CKEditorModule } from '@ckeditor/ckeditor5-angular';

also add in imports[ .. ]

Now use this line to use this editor

import \* as ClassicEditor from '@ckeditor/ckeditor5-build-classic';

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Not able to integrate it in our project , as import statement in ts file is not working.

Just close the admin Folder from the vs code

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Video 35 :

Starting with the User Module.

Just crate the new Normal user from the Register page.

THen Login , with Normal user.

Desing the basic page and divdie the 12 bootstrap grids into 2 and 10 .

Now give the basic html outlet.

To show the component on the user page , we need to use its selector and copy it to use its as tag in the main user.html page

As we have same componenet in admin also , so its not imported in app.module.ts file.

TO solve this problem we are going to import it, and give the alias name to it and then adding that alias name in the imports.

This is how we load other component on the user-dashboard component.

Now we need to dunamically ({{c.title}}) load the catergories on left sidebar, so use the Category Service. as done in admin part.

Now we need to show the fetch data in the categories ussing the for loop. \*ngFor

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Video 36 :

Make component to load all the quizes for the user.

Now we need to configure the Url for loading this component.

So go to app.routing.ts file.

Configuration will be same as we have done for the childerns of admin,

Same do for user.

\*Just remove the pathmatch full

So logic is to load the data by passing the category id in the path.

0 -means load all categories =quizes

14- specific id quizes

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

So load the quiz components using Acitvated Route object injection.

Modify the Login Component also, as we need to pass the 0 in url path.

this.router.navigate(['user-dashboard/0'])

Now we have seen the backend in springboot to load the all quizes we need to make get request to user/quiz/

using the quiz service already made.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

We can also implement pagination to show the quizes \* Taught in Smart Contact Manager

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Now showing the quizes in the .hmtl of load quiz component.

To fold the line .. need to use class tuncate and add this class in css. from google.

.truncate

{

display: block;

white-space: nowrap;

width: 100%;

overflow: hidden;

text-overflow: ellipsis;

}

Download the flaticon from google and add in the css file backgraound.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Video 37:

Now we are going to Apply the Filter accordiung to the Categories.

Now to do this we need to pass the id of the category thought the button clicked from the side bar that means we need to use the routerLink for passing the id with the url

[routerLink]="'/user-dashboard/'+c.cid"

To see thee correct varaible just go to console and see the category id varible from there.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

"Traditional approch"

Now we need to collect the id from the ACtivated Router , and then filter out the data .

As we have alredy done this so just

Now we need to make the little modification in the ts file as we have already used the route so making the subscibe direcly on the params variable,

This change will enable to load the components even when the id of url is changed

Basically reloading the content when the particular category button is clicked.

this enable us to get directed to the particular component.

Now this new modified function will even work when there is change in id of url.

ngOnInit(): void {

//Using Activated Route to fetch the catId passed using the path and then loading the quiz of Specific category

this.catId=this.\_route.snapshot.params['catId'];

//console.log(this.catId);

this.\_route.params.subscribe(

(params)=> {

this.catId=params['catId'];

if(this.catId==0)

{

console.log('Load all the quiz');

//Inject Quiz service var to call the backend

this.\_quiz.quizzes().subscribe(

(data:any)=>{

this.quizzes=data;

console.log(this.quizzes);

},

(error)=>{

console.log(error);

alert("Error in loading all quizes"); //Also can use \_snack, Swal.fire

}

);

}

else{

console.log('Specific cat loaded');

// this.quizzes=[];

}

}

);

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Also desingning the new handler method in the backend to load thee quizes according to the category id given in the url.

// VIDEO 37 : Defined this function explicity in contoller , service , impl , repository .;

Using the function in the frontend angular quiz.service.ts file and then make get request to the backend api.

public getQuizzesOfCategory(cid :any)

{

return this.\_http.get(`${baseUrl}/quiz/category/${cid}`);

}

On backend side we have made the same fucntion in Quiz Controller first.

//1.Code Quiz controller

@GetMapping("/category/{cid}")

public List<Quiz> getQuizzesOfCategory(@PathVariable("cid") Long cid)

{

Category category = new Category();

category.setCid(cid);

return this.quizService.getQuizzesOfCategory(category);

}

// 2. Declare this in interface QuizService

public List<Quiz> getQuizzesOfCategory(Category category);

// 3. Then need @Override to give the implementation of function class for the same method defined over here

@Override

public List<Quiz> getQuizzesOfCategory(Category category) {

return this.quizRepository.findBycategory(category);

}

// 4. Also this is the human made fucntion so need to define this in quiz.Repository .

//Code :

public interface QuizRepository extends JpaRepository <Quiz, Long>

{

public List<Quiz> findBycategory(Category category);

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

VIDEO 38 :Displaying only active quizes to user

Bascially the idea is to Desing the two new api in the backend, one for fetching the active quizes and other one to show the active quizes which are category specific.

We are not disturebing the old api as they are used by admin.

Go to Springboot backend.

//STEP 1 : Declare in Quiz Repository.

public List<Quiz> findByActive(Boolean b);

public List<Quiz> findByCategoryAndActive(Category c , Boolean b);

//STEP : 2 : Give declaration in service interface and then implementaion in the impl class.

// declaration in the interface ..

public List<Quiz> getActiveQuizzes();

public List<Quiz> getActiveQuizzesOfCategory(Category category)

// defination in the implementaion.class

@Override

public List<Quiz> getActiveQuizzes() {

this.quizRepository.findByActive(true);

}

@Override

public List<Quiz> getActiveQuizzesOfCategory(Category category) {

return this.quizRepository.findByCategoryAndActive(category,true);

}

// Now indicating the method in the QuizController so that they can be called using the angular frontend.

//get active Quizzes

@GetMapping("/active")

public List<Quiz> getActiveQuizzes()

{ return this.quizService.getActiveQuizzes();

}

// get Active quizzes of category.

@GetMapping("/category/active/{cid}")

public List<Quiz> getActiveQuizzes(@PathVariable("cid") Long cid)

{

Category category = new Category();

category.setCid(cid);

return this.quizService.getActiveQuizzesOfCategory(category);

}

Basically we have used the custome finder methods , to fetch the data based on active status

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Now comming back to the Frontend , to call the controllers method front the frontend quiz.wservice.ts file

// NOW USE CALL THESE METHODS IN QUIZ.Service.

//To get all active quiz

public getActiveQuizzes()

{

return this.\_http.get('${baseUrl}/quiz/active');

}

//To get only the active quiz of particular case.

public getActiveQuizzesOfCategory(cid:any)

{

return this.\_http.get(`${baseUrl}/quiz/category/active/${cid}`);

}

// Now we can use them load component.

Now replace this new two function

To load all the quiz , use getActivceQuizzes

To load specific category , use getActiveQuizzesOfCategory

NOW ONLY ACTIVE QUIZES ARE SHOWN TO THE USER .

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Video : 39 :

Setting the Instructions to show when the Start Quiz button is clicked.

Generate the new componenet in the users named instructions

1. Configure the component in the user [] as the childern , path will include the qid to get the id of quiz , of which button is clicked.

add in app.routing.ts

2. Now connect this to the start button of Quiz . using the [routerLink]=" '' + "

NOW IN .TS file of the particular component. We need to fetch the data from the url using activated Route. and then use the quiz service to get the specific quiz loaded.

qid:any;

quiz:any;

constructor(

private \_route:ActivatedRoute,

private \_quiz:QuizService,

) { }

ngOnInit(): void {

this.qid=this.\_route.snapshot.params['qid'];

console.log(this.qid);

this.\_quiz.getQuiz(this.qid).subscribe(

(data:any)=>{

console.log(data);

this.quiz=data;

},

(error)=>{

console.log(error)

Swal.fire('Error','Error in loading the Instruction page and Quiz','error');

},

);

// IMPORTANT :

['qid'] this variable should be same as the varible defined in the app.routing.ts file in path:"quiz/:cid"

Desing the basic Instruction page using the html.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

VIDEO 40 :

Make new component to Start the quiz , also bring the quiz id here.

Configure it in app.routing.ts file.

Now link it wiht the Start button.

[routerLink]="'/user-dashboard/start/'+ qid"

\*\*\*\*\* IMPORTANT :

The qid variable should be same as qid defined in local varible in the ts file.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

now to make the start component to land on clean page, we define it explicitly in routing.ts file.

That is it will go to app.component router outlet , not to user-dashboard.

Aslo need to change the url at the start quiz button.

Also attaach the can active attribute, so that direct access to page is avoided

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

to give confirmation on clicking start Quiz button , we have made a new func and aaded a click event on it.

In .ts file used The Router to access the required url .

// Method to start Quiz to add Functioanlity of Confirmation .

public startQuiz()

{

Swal.fire({

title: 'Do you want to Start the Quiz ?',

showCancelButton: true,

confirmButtonText: 'Start ',

icon:'info',

}).then((result) => {

if(result.isConfirmed)

{

this.\_router.navigate(['/start/'+ this.qid]);

}

else if (result.isDenied) {

Swal.fire('Changes are not saved', '', 'info')

}

});

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Now to BLOCK the back button ,once the quiz Started Followed code is used from google.

preventBackButton()

{

history.pushState(null,'null',location.href);

this.locationSt.onPopState(

()=>{

history.pushState(null,'null',location.href);

}

);

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Now fetch the quiz id which is comming in the url , need helpful to display the Questions of Quiz.

NOW SINCE WE NEED TO FETCH THE QUESTION IN QUIZ , so using the backend api function to fetch limited question at the time randomly ,

We also need to call this backend method in angular.serviec file, as we have not called this till now.

We have used the other fun to fetch all the Question of Quiz , but it was for admin.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Just just desing the page for showing the questions . \

Pay attention to the [name] given to the 4 options should be same as we are using the radio button here.

just using the index var i , to given name to all the four options in the given loop .

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

RADIO BUTTON POSSIBILITY OF ERROR

I have implemented the concept of selecting the option of the radio button , by clicking on the text.

By choosing the "id" field and choosing the <label> tag

If any problem arises ,just remove the id="" and the <label>

<div class="col-md-6">

<input type="radio"

id="{{q.option2}}+{{i}}"

[value]="q.option2"

[name]="i" >

<label for="{{q.option2}}+{{i}}">

{{q.option2}}

</label>

</div>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Video : 41

Now evaluating the report of the Quiz

marksGot :

Correct Answer :

Question Attempted :

Total Questions :

Success Ratio ;

Now we are taking the givenAnswer as the new field in Question entiry to hold the response of the selected option by user.

Later we can compare it with the original answer and give marks

this.questions.forEach( (q)=>{

q['givenAnswer']='';

} );

\_\_\_\_\_\_\_\_

now to bind the response selected by user with this givenAnswer var , we need to use the two way binding for it.

[(ngModel)]="q.givenAnswer" -- need to put this in all options input tag

Stopping from the ng Model , as name field is distrubed

// VIDEO : 44

Processing the Quiz on the Server Side

We need to make new handler method in back end , in my case client side is not working on adding the ngModel

// ERROR :

I have made change to the name attribute as the error was comming in loading the ngModel

[name]="i.toString()"

Temporary answer is comming in the console givenAnswer varible

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

As we have seen the user respnses are sent to the server , But at server we have the question id so we can fetch the correct answers and then can evalutate them at the server side

IN BACKEND WE Need to make a fucntion in the QuestionSerive interface to get the Question from the Question id , so we can fetch its answer.

// I had made a change in the Fetching codintion of Question for Current user randomly when more question are present.

Back Buutoon in Quiz is done using the routerLink attributre.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

VIDEO : 42

Implement the Timer using the progress bar.

Import the files related to Progress spinner and configure in app.module.ts file.

Basically desing the timer , to decrement acc to time and submit automatically when 0 0 left.

Modified originaml quizSubmit() , as made evalQuiz() , which gives the alert box on Submit button clicked, but not when submiteted by timer .

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

VIDEO 43 :

Showing the loader when request are made to server.

That is showing the loading when requests are made to backend.

USE Google and look for angular NGX ui loader.

And install them in the frontend project folder.

-install it from google.

npm install --save ngx-ui-loader

-apply it in app.module.ts file

import { NgxUiLoaderModule } from 'ngx-ui-loader';

-Need to place the tag in the root component. So that can be used anywhere.put it in app.component.html file

<ngx-ui-loader></ngx-ui-loader>

-To show the loader in any component we need to use its service and we can use various function of it.

Constuctor(

private \_loader:NgxUiLoaderService,

){

}

- To do the start and stop of loader automatically , when moving on http request so we need to import 1 more module in app.module.ts file

NgxUiLoaderHttpModule

\*\*\*\*\*

LOADER NOT WORKING HERE SO REMOVING THEM FROM HERE>>

\*\*\*\* Rich Text Editor also not worked ,

May be the issue is with latert angular version.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

// VIDEO 44 SERVER SIDE EVALUTAION OF QUIZ DONE .

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//VIDEO 45 : Solving Issuses

//TO remove the answer varible from the console loaded with the object .

SO we are manually removing it form backend in Video 46

// When normal data converted to json we call it as serialization and vice versa (deserialization json -> object )

// When we are trying to serialize , we need to ignore answer, but when deserializing ie json back to obj we needed "answer" to check the user response and do marking.

// One less-efficient solution is to use the

@JsonIgnore in getter of Answer() , so this will not be fetched on the console.

@JsonProperty on setter , this will be valid and it will set the answer in database.

// But Manually emptying the answer field is fruitfull here..

list.forEach( (q)->{

q.setAnswer("");

});

\*\*\* Also specifying the type<Question > in the List..

// Since we have single model of Question here ,

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

When we use

@Transient - hibernate igonore it to store it in database.

@JsonIgnore - we ignore this as client means no getter used

Error in video : 45

Do not take JsonIgnore on answer field, because it will affect the admin also, it anwer will not be saved for new Question added there.

\*\*\* Generating the PDF Og page , on clicking print.

(click)="printPage()" on Print Buton .

public printPage()

{

window.print();

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//Setting the instuctions on the page, while giving the Quiz on left part of page.

//Printing two digits fomating in success %

Number(data.marksGot).toFixed(2);

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

VIDEO TILL 46 Completed :

Didnt worked with Rich Editor . For Quesiton description .

Didnt worked with the Loader NgxUiLoader / loading process.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

DEPLOYING OUR PROJECT TO AMAZON EC2

VIDEO 47 :

Deploying the Exam Portal project on AWS EC2 :

Various cloud platforms like Azure , Digital Ocean , Google cloud

Using AWS cloud Ec2 instance.

Steps invloved .

1. Exam portal is build with java , mysql , nodejs(angular).

2. Create the account on AWS then sign in (24 hr) .

3. Then crate the new EC2 instance from Services. (Img 2 )

4. Choose the instance of Ubuntu Server 20.04 Free Tier eligible. Then launch it.

5\*. Now we need to create the key pair .This key is VERY imporatant to login and perform operation.

6.Rename it ,Now wait till the current instance come in running state.

7.Now click on instace id and connect the instance. Now teminal will be opened in browser.

8.Try sudu su , clear , apt update commnads.

9. We need to install the java and mysql here. SO just type java and it will give suggestion to download java-14 - same as springboot project.Copy till apt install openjdk-14-jre-headeless and paste there . Type java -version.

10.But jshell not working , Also install the jdk, which give tools to compile java . so install the jdk 14 from suggestion by copying . Now check jshell , and /exit after checking.

11.Now to install the MySql , first do apt update , then run the command "apt install mysql-server"

And check it "mysql" , "show databases"

.

12. Now creating the new user so that we can use it from anywhere(%) and aslo from the MySql workbench of localcomputer.durgesh-password.

create user 'examuser'@'%' identified by 'durgesh'

13.Now we need to give permissions .

grant all privileges on \*.\* to 'examuser'@'%' with grant option;

14. Now do "flush privileges" and "exit"

15. Now we can login with created user and give password "durgesh".

mysql -u examuser -p

16. Now we can use the mysql of Ubuntu from local Workbench . Need to do some setting in edit inbound rules.

We need to go configuration:

Go to instances -> select id of instance -> Go to security tag -> click security Groups id -> click Edit inbound rules and add a rule .-> give port 3306 and Custom TCP type 0.0.0/0 means from anywhere access is allowed.Then save.

17.Now open workbecnch on local comp and create new connection.

--> Fill name as learn-aws

Before This go to AWS sevice and allocate IP . Click on Elastic IP addresses.

And rename it to "learn-aws"

Now Actions -> associate the IP Address ->choose instance and ip address and then associate it.

Now go to instance and Copy the public IPv4 "5.129.240.240" -to be used to connect with workbench.

Refer screenshot-10

-->enter the SSH Hostname as copied ip address with :22

--> SSH username: ubuntu

--> SSH Password : is password of AWS account.

-->SSH file : select the key pair downloaded at time of account creation.

--> Now at mySQL , usename=examuser , password=root (password of MySQL)

-->now test connection.

NOW we have successfully connected the Local workbench with Online AWS- ubuntu MySQL

.........................

Check by reconnecting

- sudu su

-mysql -u examuser -p and enter password 'durgesh'

-Now show databases and cross verify each other.

-creat the database named "exam"

+++++++++++++++++++++++++++++++++++++++++++++++++++++

======================================================

VIDEO : 48

Deploying the springboot on EC2 instance "learn-aws"

1.Need to upload the jar file , before that we need to connect the instance with terminal.

If we direcly click on instance and connect button its terminal is opned on browser but we need to type every thing manually there.

(DONE IN LATER VIDEOS )

/\*So to save this we are connecting instance with the Git Bash (download git from net).\*/

2.Start with connecting with browser Connect button.

Then sudu su

-Now we go back to intellij and correct the database port to 3306 under application.properties file,also change username, password .

3.Also we are make a new api(open to all users ) to test the working on online system , ie our backend system is responding or not.

SO create a method test() , which is unauthenticated used to test.

Url will be "/user/test"

@GetMapping("/test")

public String test("/test")

{ return "welcome to backend api of Examportal";}

-Also we need to give access to url in MySecurityConfig file. ie

.antMathers("/generate-token","/user/","/user/test").permitAll()

4. Now crate the jar file of project from

Maven tag -> LifeCycle -> install

This jar file will be in target folder (open in file explorer) "examserver (0.0.1-SNAPSHOT.jar)"

Rename it as "examportal.jar"

5. Now go back to EC2 teminal ,

sudu su , apt update

If already inside /home/ubuntu#

-Then make folder :

mkdir project

-and give all permissions

chmod 777 project/

6. Now to upload jar.file we need 3 party software "FileZilla" download it from google.

7.Now launch Filezilla then file ->site manager -> new site ->name it and fill the details (screenshot img 12)

Use Protol : SFTP ,

Host : IP address of EC2 instance.

Port : 22 (port of SSH is 22 )

Logon: KeyFile

User : Ubuntu

Select the key pair file downloaded at time of aws account creation.

Now connect to the learn-aws again from start.

Go to project folder in ubuntu server (right side)

8. Now we need to drag - drop the jar file into the project folder of Ubuntu.

Now we can verify it from the online-terminal, ls command -> cd project -> ls .

9.Now we can run the jar file

"java -jar examportal.jar"

Projec is running on port 8080

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

10. Now we need to opne the port 8080 of EC2 instance Dashboard ->instances -> inst ID -> Security ->Security Gruops ->Edit inbound rules ->add rule to port 8080 from anywhere(0.0.0) and save.

11. Now check directly on google : IP Address : 8080

=> 18.119.154.231:8080/user/test - to verify springboot is responding.

12. Now problem is "teminal is tempory" so need to run it as backend service. Also we can attach the domain name to ip:

Also we can verify from workbench tables are created.

++++++++++++++++++++++++++++++++++++++++++++++++++++++++

=======================================================================

VIDEO 49 :

Now creating the service in Ubuntu , to make the springboot application run in background of EC2 instance.

1.Need to do change in pom.xml , need to add <configuration> tag in build/plugins

<configuration>

<executable> true

</executable>

</configuration>

2.Now again generate the .rar file using the maven -> lifecycle ->install

Now rename the generated file as examportal.rar

3.Now go to the online terminal of EC2 instance , and login using super user

sudo su

clear

Connect back the filezilla and drag and drop the new examserver file in project folder and confirm the overwrite previous file.

3.Now write following script in notepad.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

[Unit]

Description=A Spring Boot Application for examportal

After=syslog.target

[Service]

User=ubuntu

ExecStart=/home/ubuntu/project/examportal.jar

SuccessExitStatus=143

[Install]

WantedBy=multi-user.target

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4.Now we need to save this file in project subfoldder

Path: /etc/sustemd/system

name: examportal.service

5.Go to specified path in teminal project/etc/systemd/system and create file.

vim examportal.service

Now we need to paste the code of Note pad in this file. but copy/paste wont work here.

6.Now go to Online EC2 instance and connect option , and copy the link to connect with the SSH Client .

NOW GO TO THE SAME FOLDER WHERE WE DOWNLOADED THE KEY PAIR(learn-aws.pem) at the time of making AWS account (downloads)

And OPEN the git-bash here.(Need to download the git software)

7.Now copy the SSH connection link in gitbase teminal.So your Online EC2 teminal will be opened here in gitbash.

Login with super user.

sudu su

clear

8.No go to same folder from home/ubuntu#

cd etc/systemd/system

then : vim examportal.service and paste the nodepad text here (if error use i to go to insert mode then paste)

9.Now quit and save using= :wq!

10. Now need to enalble this service , so it automatically starts.

Other commands

-Enable : systemctl enable examportal

-Start : systemctl start examportal

-Status : systemctl status examportal

-Stop : systemctl stop examportal

-Restart : systemctl restart examportal

11: Now enable and start the service using the above commands and then check the status of it.

12.if error comes ,we need to give permission to new jar file using

chmod 777 /home/ubuntu/project/examportal.jar

13.Now again start it and see the status.

14.Now we can exit the terminal , and service will keep on running .ie backend springboot is active.We need to explicitly stop the service using above command.

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

=========================================================

VIDEO : 50

Buying the domain name , to point the IP address in the url.

1.Sign up and sign in on GoDaddy.com and search the domain name according your choice and buy it (.com => 900 rs / year ).

2.Now go to MyProducts section on Godaddy and open the domain name -> Manage DNS .

3. Copy the IP address from the EC2 instance of AWS.

4. Now change the value of @ with IP address u have copied under the Points to option and Save.(Wait 1 hr or try in Incognito mode of Chrome)

-Now check it : www.codewithdurgesh:8080/user/test

5.By default port is 80 , otherwise need to mention expicity as in above case.

Basically we deploy the frontend part on the port 80 , so it can domain name alone can point to the project.

SO TILL NOW OUT BACKEND service is running perfectly on 8080 port.

6.Now go to VSCode and open the helper.ts file of examfront project and need to change the address of backend in angular project as it need to point the backend of springboot project deployed on EC2 instance with given domain name and port.

//Locally we have : let baseUrl='http://localhost:8080'

In deploying :

let baseUrl='http://codewithdurgesh.com:8080'

export default baseUrl;

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

=============================================================

VIDEO : 51 :

DEPLOYING frontend / Angular on EC2 instance.

1. We are going to use Nginx server

2. Create the compressed folder "dist" of angular project using terminal command.

ng build --prod

3.Need to install the nginx server in online EC2 instance GitBash terminal.

If disconnected , Reconnent it using same SSH link of EC2 instance copying it ->Then opeing the Git Bash in folder where pair key is kept. -> Login using the sudu su ->clear -> apt update

4.Install : apt install nginx

5. Now need to enable & start - this server in backend using the command same as we have used for examservice and then check the staus of server.

Enable - systemctl enable nginx

Start - systemctl start nginx

Stop- systemctl stop nginx

Restart - systemctl restart nginx

Status - systemctl status nginx

6.Now go to EC2 instance of AWS and open the 80 port no , so it can be accessed from anywhere.

Go to EC2 Instance -> Security -> Security group -> Edit inbound rule -> add new rule -> Type : Custom TCP and 0.0.0.0/0 (open form anywhere) , port 80 : and save.

7. NOW we need to make a file to tell which directory should server serve on port 80 .

8. If we see the angular the dist folder will be created which has the index.html file inside the examfront.

9.Just Drag and drop the "examfront" folder of dist folder using the FileZilla. (If filezilla was disconnect connect it from the File -> connect to learn-aws)

10. Now in GitBash : go to given path and create file with given name and copy the text given below.

path : /etc/nginx/conf.d/

name : codewithdurgesh.com.conf

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

server {

listen 0.0.0.0:80;

root /home/ubuntu/project/examfront;

server\_name codewithdurgesh.com;

location / {

try\_files $uri $uri/ /index.html;

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

11. Now create the file inside /etc/nginx/conf.d# using :

vim codewithdurgesh.com.conf

--> paste the content and if INSERT is visible at bottom left - press the ESC key. and :wq!

(Make sure http )

12. Now restart the nginx server using command and check the status (Wait 1 hr and stop-start try ).

We can see that exam portal is live on port:80 or directly on domain name.

http:codewithdurgesh.com

NOW WHOLE PROJECT IS WORKING FINE.

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

================================================================

VIDEO : 52

Attaching SSL / making https

1.SSL (Secure Socket Layer) secures that the data is ecrypted and then data is shared to server.

2.Connect the EC2 Instance using the SSH link on Git Bash (in folder of key pair). Then login the super user (sudu su)

3.Now check backend service and nginx server is running or not - ie http:codewithdurgesh.com is running or not.

systemctl status examportal

systemctl status nginx

4.To get the SSL certificate (make https) we are using CloudFlare (Third party) Login and signin in it.

5.After creating account : Give the site name :

codewithdurgesh.com -> select free plan -> continue and follow the instuctions to removee the nameservers from Godaddy and paste new nameservers there .

6.Now login back to Godaddy.com -> My Products ->Manage DNS and click Nameservers change (take screenshot of old name servers )

7.Enter your own name servers -> copy paste the nameservers of Cloudfare here ->Save (Wait 1 hr ).

8.Follow the Quick guide and click options visely

Always https - yes

Auto minfy - html,javascript,css and Save all.

9.ALso confim the IP address is correct in CloudFlare under content section(DNS management tag).

10.After 10 miniutes , Frontend worked well (but backend is not working )

BASICALLY ERROR is :

Frontend is working fine with SSL , but backend is not working , so we have used the nginx server for redirecting the request so that backend also work with SSL security.

We are desinging the .conf file with required code , That is when we make request to api.codewithdurgesh.com then it redirects to localhost:8080 (which is actual backend of EC2 instance ) and also SSL is ensured.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Working.

NOW we are simplifying the working of loading backend with https also we are doing mapping .

--We will make request " api.codewithdurgesh.com " to the Nginx server and then nginx server will redirect to http://codewithdurgesh.com

--Now we need to make file using the name under the path given below and paste the text there.

path : /etc/nginx/conf.d/

name: api.codewithdurgesh.com.conf

code:::

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

server {

listen 80;

server\_name api.codewithdurgesh.com;

location / {

proxy\_pass "http://localhost:8080";

proxy\_set\_header Host api.codewithdurgesh.com;

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Basically the request come to nginx at 80 port and redirect to localhost:8080

11 .

Now login to GitBash using the SSH link of EC2 instance. paste the link and login with sudu su

12. Make the file in folder (already 1 file is present , check by ls commnad).

ubuntu : cd/etc/nginx/conf.d

13.Make file with command , and copy paste the content then close it using :wq!

vim api.codewithdurgesh.com.conf

14.Then restart it : systemctl restart nginx ,also check status .

15. Now go to the CLOUDFLARE , CLICK ON DNS ICON .

Now add the record with name=api and the required same ip address and save it.

16.Wait 20 minutes and make the seperate call to the backend site. https://api.codewithdurgesh.com/user/test

NOW SSL IS ALSO APPLIED ON BACKEND BUT we have used the nginx subdomain for it .

So we need to again update the Angular - VSCOde helper.ts file with this new url for accessing the backend service.

17. just Again make change to baseUrl

let baseUrl='https://api.codewithduresh.com';

export default baseUrl;

18: build the file using command

ng build --prod

19.Now drag and drop this new files under the dist/examfront folder to already uploaded examfront folder under project of Ubuntu EC2 instance using the FileZilla app.

Basically Drag - drop new files to overwrite the old files of examfront

20: Now restart the nginx server and check status of both service and nginx server (wait 30 min and try incognito mode )

. Now frontend and backend both are working fine

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

DONE WITH DEPLOYMENT.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Extra

Added images in app component

Added mt in dashboard admin

-Features of App.

bcrypt\_password\_encryption