Capstone Project

MAdhar Application

Source code

-> Backend

-> MyApplication.java

```
package com;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.boot.autoconfigure.domain.EntityScan;
import
org.springframework.data.jpa.repository.config.EnableJpaRepositories;
@SpringBootApplication(scanBasePackages = "com")
@EntityScan(basePackages = "com.aadhar.bean")
@EnableJpaRepositories(basePackages = "com.aadhar.repository")
public class MyAppApplication {
public static void main(String[] args) {
SpringApplication.run(MyAppApplication.class, args);
System.out.println("Server running on port number 9090.....");
}
}
```

-> Applications.java

```
package com.aadhar.bean;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
@Entity
public class Applications {
@Id
@GeneratedValue(strategy = GenerationType.IDENTITY)
private int appid;
private String fname;
private String lname;
private int age;
private String emailid;
private String address;
private String place;
public int getAppid() {
return appid;
}
```

```
public void setAppid(int appid) {
this.appid = appid;
}
public String getFname() {
return fname;
}
public void setFname(String fname) {
this.fname = fname;
}
public String getLname() {
return lname;
}
public void setLname(String lname) {
this.lname = lname;
}
public int getAge() {
return age;
}
public void setAge(int age) {
this.age = age;
}
public String getEmailid() {
return emailid;
}
public void setEmailid(String emailid) {
this.emailid = emailid;
}
public String getAddress() {
return address;
}
public void setAddress(String address) {
this.address = address;
}
public String getPlace() {
return place;
}
public void setPlace(String place) {
this.place = place;
}
@Override
public String toString() {
return "Applications [appid=" + appid + ", fname=" + fname +
", lname=" + lname + ", age=" + age + ", emailid="
+ emailid + ", address=" + address + ", place=" +
place + "]";
}
}
```

->Login.java

```
package com.aadhar.bean;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
@Entity
public class Login {
@Id
private String emailid;
private String password;
@Column(name = "typeofuser")
private String typeOfUser;
public String getEmailid() {
return emailid;
}
public void setEmailid(String emailid) {
this.emailid = emailid;
}
public String getPassword() {
return password;
}
public void setPassword(String password) {
this.password = password;
}
public String getTypeOfUser() {
return typeOfUser;
}
public void setTypeOfUser(String typeOfUser) {
this.typeOfUser = typeOfUser;
}
@Override
public String toString() {
return "Login [emailid=" + emailid + ", password=" + password + ",
typeOfUser=" + typeOfUser + "]";
}
}
```

-> ApplicationsController.java

```
package com.aadhar.controller;
import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.MediaType;
import org.springframework.web.bind.annotation.CrossOrigin;
import org.springframework.web.bind.annotation.DeleteMapping;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PatchMapping;
```

```
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
import com.aadhar.bean.Applications;
import com.aadhar.service.ApplicationsService;
@RestController
@RequestMapping("applications")
@CrossOrigin
public class ApplicationsController {
@Autowired
ApplicationsService applicationsService;
@PostMapping(value = "storeApplications",consumes =
MediaType.APPLICATION_JSON_VALUE)
public String storeApplications(@RequestBody Applications
applications) {
return applicationsService.storeApplications(applications);
}
@PatchMapping(value = "updateApplications",consumes =
MediaType.APPLICATION_JSON_VALUE)
public String updateApplications(@RequestBody Applications
applications) {
return applicationsService.updateApplications(applications);
}
@GetMapping(value="findAllApplications",produces =
MediaType.APPLICATION_JSON_VALUE)
public List<Applications> getAllProduct() {
return applicationsService.getAllApplications();
}
@GetMapping(value="findApplicationsById/{appid}")
public String findProductById(@PathVariable("appid") int appid) {
return applicationsService.findApplicationsById(appid);
}
@DeleteMapping(value="deleteApplications/{appid}")
public String deleteProductUsingId(@PathVariable("appid") int appid)
{
return applicationsService.deleteApplications(appid);
}
}
```

-> LoginController.java

```
package com.aadhar.controller;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.MediaType;
import org.springframework.web.bind.annotation.CrossOrigin;
import org.springframework.web.bind.annotation.PostMapping;
```

```
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
import com.aadhar.bean.Login;
import com.aadhar.service.LoginService;
@RestController
@RequestMapping("login")
@CrossOrigin
public class LoginController {
@Autowired
LoginService loginService;
@PostMapping(value = "signIn",consumes =
MediaType.APPLICATION_JSON_VALUE)
public String signIn(@RequestBody Login login) {
System.out.println("I cam here");
return loginService.signIn(login);
}
@PostMapping(value = "signUp",consumes =
MediaType.APPLICATION_JSON_VALUE)
public String signUp(@RequestBody Login login) {
System.out.println(login);
return loginService.signUp(login);
}
}
```

-> Applications Repository.java

```
package com.aadhar.repository;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;
import com.aadhar.bean.Applications;
@Repository
public interface ApplicationsRepository extends
JpaRepository<Applications, Integer>{
}
```

-> Login Repository.java

```
package com.aadhar.repository;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;
import com.aadhar.bean.Login;
@Repository
public interface LoginRepository extends JpaRepository<Login, String>{
}
```

-> ApplicationsService.java

```
package com.aadhar.service;
import java.util.List;
import java.util.Optional;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import com.aadhar.bean.Applications;
import com.aadhar.repository.ApplicationsRepository;
@Service
public class ApplicationsService {
@Autowired
ApplicationsRepository applicationsRepo;
public String storeApplications(Applications applications) {
applicationsRepo.save(applications);
return "Applications details stored";
}
public List<Applications> getAllApplications() {
return applicationsRepo.findAll();
}
public String findApplicationsById(int appid) {
Optional<Applications> result =
applicationsRepo.findById(appid);
if(result.isPresent()) {
Applications a = result.get();
return a.toString();
}else {
return "Applications not present";
}
}
public String deleteApplications(int appid) {
Optional<Applications> result =
applicationsRepo.findById(appid);
if(result.isPresent()) {
Applications a = result.get();
applicationsRepo.delete(a);
return "Applications deleted successfully";
}else {
return "Applications not present";
}
}
public String updateApplications(Applications applications) {
Optional<Applications> result =
applicationsRepo.findById(applications.getAppid());
if(result.isPresent()) {
Applications a = result.get();
a.setEmailid(applications.getEmailid());
a.setAddress(applications.getAddress());
a.setPlace(applications.getPlace());
applicationsRepo.saveAndFlush(a);
```

```
return "Applications updated successfully";
}else {
return "Applications not present";
}
}
}
```

-> Loginservice.java

```
package com.aadhar.service;
import java.util.Optional;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import com.aadhar.bean.Login;
import com.aadhar.repository.LoginRepository;
@Service
public class LoginService {
@Autowired
LoginRepository loginRepository;
public String signIn(Login login) {
Optional<Login> result =
loginRepository.findById(login.getEmailid());
if(result.isPresent()) {
Login ll = result.get();
if(ll.getPassword().equals(login.getPassword())) {
if(login.getTypeOfUser().equals(ll.getTypeOfUser())
&& login.getTypeOfUser().equals("admin")) {
return "Admin sucessfully login";
}else
if(login.getTypeOfUser().equals(ll.getTypeOfUser()) &&
login.getTypeOfUser().equals("user")){
return "User successfully login";
}else {
return "Invalid details";
}
}else {
return "Invalid password";
}
}else {
return "Invalid emailId";
}
}
public String signUp(Login login) {
Optional<Login> result =
loginRepository.findById(login.getEmailid());
if(result.isPresent()) {
return "Email Id alreay exists";
}else {
```

```
if(login.getTypeOfUser().equals("admin")) {
return "You can't create admin account";
}else {
loginRepository.save(login);
return "Account created successfully";
}
}
}
}
```

Docker file

```
FROM openjdk:8
COPY target/aadhar-backend.jar .
CMD ["java","-jar","aadhar-backend.jar"]
```

->Pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion>
<parent>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-parent</artifactId>
<version>2.6.12</version>
<relativePath/> <!-- lookup parent from repository -->
</parent>
<groupId>com.example</groupId>
<artifactId>backend-app</artifactId>
<version>0.0.1-SNAPSHOT</version>
<name>MyApp</name>
<description>Demo project for Spring Boot</description>
<properties>
<java.version>1.8</java.version>
</properties>
<dependencies>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-web</artifactId>
</dependency>
<dependency>
```

<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-devtools</artifactId>
<scope>runtime</scope>
<optional>true</optional>
</dependency>
<dependency>
<groupId>mysql</groupId>
<artifactId>mysql-connector-java</artifactId>
<scope>runtime</scope>
</dependency>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-test</artifactId>
<scope>test</scope>
</dependency>
</dependencies>
<build>
<finalName>aadhar-backend</finalName>
<plugins>
<plugin>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-maven-plugin</artifactId>
</plugin>
</plugins>
</build>
</project>

-> Frontend

-> Admindashboard

-> Admindashboard.component.html

<h2>Welcome to home page {{user}}</h2>
<table border =1>
<th>Particulars</th>
<th>Actions </th>
<tr>
<td>View Everyone's Aadhar details: </td>
<td><a routerLink="findAllApplicationsById">View</td>
</tr>
</table>
<router-outlet> </router-outlet>

<input type="button" value="logout" (click)="logout()"/>

-> admindashboard.component.ts

```
import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
@Component({
selector: 'app-admindashboard',
templateUrl: './admindashboard.component.html',
styleUrls: ['./admindashboard.component.css']
})
export class AdmindashboardComponent implements OnInit {
user:string ="";
constructor(public router:Router) { }
ngOnInit(): void {
let obj = sessionStorage.getItem("userDetails");
if(obj!=null){
this.user=obj;
}
}
logout() {
sessionStorage.removeItem("userDetails");
this.router.navigate(["login"]);
}
}
```

-> Applications

-> Applications.component.html

```
<div>
<h2>Create a new Application: </h2>
<form [formGroup]="applicationRef" (ngSubmit)="storeApplications()">
<label>First Name</label>
<input type="text" formControlName="fname"><br/>
<label>Last Name</label>
<input type="text" formControlName="lname"><br/>
<label>Age</label>
<input type="text" formControlName="age"><br/>
<label>Email ID</label>
<input type="text" formControlName="emailid"><br/>
<label>Address</label>
<input type="text" formControlName="address"><br/>
<label>Place</label>
<input type="text" formControlName="place"><br/>
<input type="submit" value="Store Application"/><br/>
<input type="reset" value="reset"/><br/>
</form><br/>
<span style="color:red">{{storeMsg}}</span>
</div>
```

-> applications.companent.ts

```
import { Component, OnInit } from '@angular/core';
import {FormGroup,FormControl} from '@angular/forms'
import { ApplicationsService } from '../applications.service';
@Component({
selector: 'app-applications',
templateUrl: './applications.component.html',
styleUrls: ['./applications.component.css']
})
export class ApplicationsComponent implements OnInit {
applicationRef = new FormGroup({
appid:new FormControl(),
fname:new FormControl(),
lname:new FormControl(),
age:new FormControl(),
emailid:new FormControl(),
address:new FormControl(),
place:new FormControl(),
})
storeMsg :string =""
constructor(public as:ApplicationsService) { }
ngOnInit(): void {
}
storeApplications() {
let application = this.applicationRef.value;
this.as.storeApplications(application).subscribe({
next:(result:any)=>this.storeMsg=result,
error:(error:any)=>console.log(error),
complete:()=>console.log("completed")
})
this.applicationRef.reset();
}
findApplication(appid:number) {
this.as.findAllApplicationsById(appid).subscribe({
next:(result:any)=>console.log(result),
error:(error:any)=>console.log(error),
complete:()=>console.log("completed")
})
}
}
```

-> Applications-operations

-> ApplicationsOperations.component.html

```
<div>
<h2>Aadhar Application Details</h2>
```

```
<div *ngIf="flag">
<h2>Update Aadhar details</h2>
<form (ngSubmit)="updateDataFromDb()">
<label>Application Id</label>
<input type="number" name="appid" [(ngModel)]="appid"
readonly/><br/>
<label>First name</label>
<input type="text" name="fname" [(ngModel)]="fname"/><br/>
<label>Last Name</label>
<input type="text" name="lname" [(ngModel)]="lname"/><br/>
<label>Age</label>
<input type="number" name="age" [(ngModel)]="age"/><br/>
<label>Email ID</label>
<input type="text" name="emailid"
[(ngModel)]="emailid"/><br/>
<label>Address</label>
<input type="text" name="address"
[(ngModel)]="address"/><br/>
<label>Place</label>
<input type="text" name="place" [(ngModel)]="place"/><br/>
<input type="submit" value="update data"/>
<input type="reset" value="reset"/>
</form>
</div>
<span *ngFor="let a of applications">
<span>
<table border="1">
<th>Application ID</th>
<th>First name</th>
<th>Last name</th>
<th>Age</th>
<th>Email ID</th>
<th>Address</th>
<th>Place</th>
<tr>
<td>{{a.appid}}</td>
<td>{{a.fname}}</td>
<td>{{a.lname}}</td>
<td>{{a.age}}</td>
<td>{{a.emailid}}</td>
<td>{{a.address}}</td>
<td>{{a.place}}</td>
</tr>
</table>
</span>
<input type="button" value="delete"
(click)="deleteApplications(a.appid)"/>
<input type="button" value="update"
```

(click)="updateApplications(a)"/>

-> ApplicationsOperatioms.component.ts

```
import { Component, OnInit } from '@angular/core';
import { Applications } from '../applications';
import { ApplicationsService } from '../applications.service';
@Component({
selector: 'app-applications-operations',
templateUrl: './applications-operations.component.html',
styleUrls: ['./applications-operations.component.css']
})
export class ApplicationsOperationsComponent implements OnInit {
applications:Array<Applications>=[];
constructor(public ps:ApplicationsService) { }
ngOnInit(): void {
this.findAllApplications();
}
flag:boolean = false;
appid:number =0;
fname:string="";
lname:string="";
age:number=0;
emailid:string=""
address:string="";
place:string ="";
findAllApplications() {
this.ps.findAllApplications().subscribe({
next:(result:any)=>this.applications=result,
error:(error:any)=>console.log(error),
complete:()=>console.log("completed")
})
}
findApplication(appid:number) {
this.ps.findAllApplicationsById(appid).subscribe({
next:(result:any)=>console.log(result),
error:(error:any)=>console.log(error),
complete:()=>{
this.findAllApplications();
}
})
}
deleteApplications(appid:number){
//console.log(pid)
this.ps.deleteApplicationsById(appid).subscribe({
next:(result:any)=>console.log(result),
```

```
error:(error:any)=>console.log(error),
complete:()=>{
this.findAllApplications();
}
})
}
updateApplications(application:any){
this.flag= true;
this.appid=application.appid;
this.fname=application.fname;
this.lname=application.lname;
this.age=application.age;
this.emailid=application.emailid;
this.address=application.address;
this.place=application.place;
}
updateDataFromDb(){
let application =
{appid:this.appid,fname:this.fname,lname:this.lname,age:this.age,emailid:t
his.emailid,address:this.address,place:this.place};
this.ps.updateApplications(application).subscribe({
next:(result:any)=>console.log(result),
error:(error:any)=>console.log(error),
complete:()=>{
this.findAllApplications();
}
})
this.flag=false;
}
}
```

-> Login

-> Login.component.html

```
<div>
<h2>Login Page</h2>
<form [formGroup]="loginRef" (ngSubmit)="signIn()">
<label>EmailId</label>
<input type="email" formControlName="emailid"/><br/>
<label>Password</label>
<input type="password" formControlName="password"/><br/>
<label>TypeOfUser</label>
<input type="radio" name="typeOfUser" value="admin"
formControlName="typeOfUser"/>admin
<input type="radio" name="typeOfUser" value="user"
formControlName="typeOfUser"/>user<br/>
<input type="submit" value="signIn"/>
```

```
<input type="reset" value="reset"/>
</form>
<br/>
<span style="color:red">{{msg}}</span><br/>
<a routerLink="/signUp">SignUp</a>
</div>
```

-> Login.componenet.ts

```
import { Component, OnInit } from '@angular/core';
import {FormGroup,FormControl} from '@angular/forms';
import { Router } from '@angular/router';
import { LoginService } from '../login.service';
@Component({
selector: 'app-login',
templateUrl: './login.component.html',
styleUrls: ['./login.component.css']
})
export class LoginComponent implements OnInit {
loginRef = new FormGroup({
emailid:new FormControl(),
password:new FormControl(),
typeOfUser:new FormControl()
});
msg:string=""
constructor(public ls:LoginService,public router:Router) { }
ngOnInit(): void {
}
signIn(){
let login = this.loginRef.value;
console.log(login);
this.ls.signIn(login).subscribe({
next:(result:any)=>{
console.log(result);
if(result=="Admin sucessfully login"){
sessionStorage.setItem("userDetails",login.emailid);
this.router.navigate(["adminHome"])
}else if(result=="User successfully login"){
sessionStorage.setItem("userDetails",login.emailid);
this.router.navigate(["userHome"])
}else {
this.msg=result;
}
},
error:(error:any)=>console.log(error),
complete:()=>console.log("completed")
})
}
```

```
}
```

-> Signup

-> Signup.component.html

```
<div>
<h2>Account Create</h2>
<form [formGroup]="loginRef" (ngSubmit)="signUp()">
<label>EmailId</label>
<input type="email" formControlName="emailid"/><br/>
<label>Password</label>
<input type="password" formControlName="password"/><br/>
<label>TypeOfUser</label>
<input type="radio" name="typeOfUser" value="admin"
formControlName="typeOfUser"/>admin
<input type="radio" name="typeOfUser" value="user"
formControlName="typeOfUser"/>user<br/>
<input type="submit" value="signUp"/>
<input type="reset" value="reset"/>
</form>
<br/>
<span style="color:red">{{msg}}</span><br/>
<a routerLink="/login">login</a>
</div>
```

-> signup.component.ts

```
import { Component, OnInit } from '@angular/core';
import {FormGroup,FormControl} from '@angular/forms';
import { LoginService } from '../login.service';
@Component({
selector: 'app-signup',
templateUrl: './signup.component.html',
styleUrls: ['./signup.component.css']
})
export class SignupComponent implements OnInit {
loginRef = new FormGroup({
emailid:new FormControl(),
password:new FormControl(),
typeOfUser:new FormControl()
});
msg:string=""
constructor(public ls:LoginService) { }
ngOnInit(): void {
}
signUp() {
let login = this.loginRef.value;
```

```
this.ls.signUp(login).subscribe({
next:(result:any)=>this.msg=result,
error:(error:any)=>console.log(error),
complete:()=>console.log("completed")
})
}
}
```

-> Userdashboard

-> Userdashboard.componenet.html

```
<div>
<h2>Welcome to home page {{user}}</h2>
<table border =1>
<th>What would you like to do?</th>
<th>Actions: </th>
<tr>
<td>Apply for new Aadhar card: </td>
<td><a routerLink="storeApplications">Apply</a></td>
</tr>
<tr>
<td>View Aadhar details: </td>
<td><a routerLink="findAllApplicationsById/">View</a></td>
</tr>
</table>
<router-outlet> </router-outlet>
<br/>
<input type="button" value="logout" (click)="logout()"/>
</div>
```

-> userdashboard.component.ts

```
import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
@Component({
selector: 'app-userdashboard',
templateUrl: './userdashboard.component.html',
styleUrls: ['./userdashboard.component.css']
})
export class UserdashboardComponent implements OnInit {
user:string ="";
constructor(public router:Router) { }
ngOnInit(): void {
let obj = sessionStorage.getItem("userDetails");
if(obj!=null){
this.user=obj;
}
```

```
}
logout() {
sessionStorage.removeItem("userDetails");
this.router.navigate(["login"]);
}
}
```

->App-routing.module.ts

```
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { ApplicationsComponent } from
'./applications/applications.component';
import { ApplicationsOperationsComponent } from
'./applications-operations/applications-operations.component';
import { AdmindashboardComponent } from
'./admindashboard/admindashboard.component';
import { LoginComponent } from './login/login.component';
import { SignupComponent } from './signup/signup.component';
import { UserdashboardComponent } from
'./userdashboard/userdashboard.component';
const routes: Routes = [
{path:"login",component:LoginComponent},
{path:"adminHome",component:AdmindashboardComponent,children:[
{path:"findAllApplicationsById",component:ApplicationsOperationsComponent}
,
]},
{path:"userHome",component:UserdashboardComponent,children:
[{path:"storeApplications",component:ApplicationsComponent},
{path:"findAllApplicationsById",component:ApplicationsOperationsComponent}
,
{path:"updateApplications",component:ApplicationsOperationsComponent},
{path:"deleteApplications",component:ApplicationsOperationsComponent},]},
{path:"signUp",component:SignupComponent},
{path:"",redirectTo:"login",pathMatch:"full"}
];
@NgModule({
imports: [RouterModule.forRoot(routes)],
exports: [RouterModule]
})
export class AppRoutingModule { }
```

App.component.html

```
<div align="center">
<h2>Aadhar Application</h2>
<hr/>
<router-outlet></router-outlet>
</div>
```

->App.module.ts

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import {HttpClientModule} from '@angular/common/http';
import { FormsModule,ReactiveFormsModule } from '@angular/forms';
import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { LoginComponent } from './login/login.component';
import { SignupComponent } from './signup/signup.component';
import { UserdashboardComponent } from
'./userdashboard/userdashboard.component';
import { AdmindashboardComponent } from
'./admindashboard/admindashboard.component';
import { ApplicationsComponent } from
'./applications/applications.component';
import { ApplicationsOperationsComponent } from
'./applications-operations/applications-operations.component';
@NgModule({
declarations: [
AppComponent,
LoginComponent,
SignupComponent,
UserdashboardComponent,
AdmindashboardComponent,
ApplicationsComponent,
ApplicationsOperationsComponent
],
imports: [
BrowserModule,
AppRoutingModule,HttpClientModule,FormsModule,ReactiveFormsModule
],
providers: [],
bootstrap: [AppComponent]
})
export class AppModule { }
Applications.service.ts
import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { Observable } from 'rxjs';
import { Applications } from './applications';
@Injectable({
providedIn: 'root'
})
export class ApplicationsService {
baseUrl:string ="http://localhost:9090/applications/"
constructor(public http:HttpClient) { }
storeApplications(applications:any):Observable<string> {
```

```
return
this.http.post(this.baseUrl+"storeApplications",applications,{responseType
:"text"});
}
updateApplications(applications:any):Observable<string> {
return
this.http.patch(this.baseUrl+"updateApplications",applications,{responseTy
pe:"text"});
}
findAllApplications():Observable<Applications[]> {
return
this.http.get<Applications[]>(this.baseUrl+"findAllApplications");
}
findAllApplicationsById(appid:number):Observable<string> {
return
this.http.get(this.baseUrl+"findAllApplicationsById/"+appid,{responseType:
"text"});
}
deleteApplicationsById(appid:number):Observable<string> {
return
this.http.delete(this.baseUrl+"deleteApplications/"+appid,{responseType:"t
ext"});
}
}
```

->Applications.ts

```
export class Applications {
constructor(public appid:number,
public fname:string,
public lname:string,
public age:number,
public emailid:string,
public address:string,
public place:string,){}
}
```

->Login.service.ts

```
import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { Observable } from 'rxjs';
@Injectable({
providedIn: 'root'
})
export class LoginService {
baseURL:string ="http://localhost:9090/login";
constructor(public http:HttpClient) { }
```

```
signIn(login:any):Observable<string> {
return
this.http.post(this.baseURL+"/signIn",login,{responseType:"text"});
}
signUp(login:any):Observable<string> {
return
this.http.post(this.baseURL+"/signUp",login,{responseType:"text"});
}
}
```