

FSD Java Capstone Project Problem Statement - m-AadharApplication

BackEnd Code :-

Aadhar.java

```
package com.mAadhar.bean;
```

```
import javax.persistence.Entity;
```

```
import javax.persistence.GeneratedValue;
```

```
import javax.persistence.GenerationType;
```

```
import javax.persistence.Id;
```

```
@Entity
```

```
public class Aadhar {
```

```
    @Id
```

```
    @GeneratedValue(strategy = GenerationType.IDENTITY)
```

```
    private int cid;
```

```
    private String cname;
```

```
    private String dob;
```

```
    private String address;
```

```
    private String emailid;
```

```
    private String number;
```

```
    private String gender;
```

```
    public int getCid() {
```

```
        return cid;
```

```
    }
```

```
    public void setCid(int cid) {
```

```
        this.cid = cid;
    }
    public String getCname() {
        return cname;
    }
    public void setCname(String cname) {
        this.cname = cname;
    }
    public String getDob() {
        return dob;
    }
    public void setDob(String dob) {
        this.dob = dob;
    }
    public String getAddress() {
        return address;
    }
    public void setAddress(String address) {
        this.address = address;
    }
    public String getEmailid() {
        return emailid;
    }
    public void setEmailid(String emailid) {
        this.emailid = emailid;
    }
    public String getNumber() {
        return (String) number;
```

```

    }
    public void setNumber(String number) {
        this.number = (String) number;
    }
    public String getGender() {
        return gender;
    }
    public void setGender(String gender) {
        this.gender = gender;
    }
    @Override
    public String toString() {
        return "Aadhar [cid=" + cid + ", cname=" + cname + ", dob="
+ dob + ", address=" + address + ", emailid="
        + emailid + ", number=" + number + ", gender=" +
gender + "]\n";
    }

}

```

login.java

```

package com.mAadhar.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 setEmailis(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;
```

```
    }
```

```
}
```

AadharController.java

```
package com.mAadhar.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.mAadhar.bean.Aadhar;
```

```
import com.mAadhar.service.AadharService;
```

```
@RestController
```

```
@RequestMapping("aadhar")
```

```
@CrossOrigin
```

```

public class AadharController {
    @Autowired
    AadharService aadharService;

    @PostMapping(value = "storeAadhar", consumes =
MediaType.APPLICATION_JSON_VALUE)
    public String storeAadhar(@RequestBody Aadhar aadhar) {
        return aadharService.storeAadhar(aadhar);
    }

    @PatchMapping(value = "updateAadhar", consumes =
MediaType.APPLICATION_JSON_VALUE)
    public String updateAadhar(@RequestBody Aadhar aadhar) {
        return aadharService.updateAadhar(aadhar);
    }

    @GetMapping(value = "findAllAadhar", produces =
MediaType.APPLICATION_JSON_VALUE)
    public List<Aadhar> getAllAadhar(){
        return aadharService.getAllAadhar();
    }

    @GetMapping(value = "findAllAadhar/{cid}")
    public String findAadharById(@PathVariable("cid") int cid) {
        return aadharService.findAadharById(cid);
    }

    @DeleteMapping(value = "deleteAadhar/{cid}")
    public String deleteAadharUsingId(@PathVariable("cid") int cid) {
        return aadharService.deleteAadhar(cid);
    }
}

```

LoginController.java

```
package com.mAadhar.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.mAadhar.bean.Login;
import com.mAadhar.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) {
        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);
    }
}

```

MAadharAppApplication.java

```

package com.mAadhar.main;
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.mAadhar.bean")
@EnableJpaRepositories(basePackages = "com.mAadhar.repository")
public class MAadharAppApplication {

    public static void main(String[] args) {
        SpringApplication.run(MAadharAppApplication.class, args);
        System.out.println("Server running on port number 9091");
    }
}

```



```
}
```

AadharService.java

```
package com.mAadhar.service;
```

```
import java.util.List;
```

```
import java.util.Optional;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.stereotype.Service;
```

```
import com.mAadhar.bean.Aadhar;
```

```
import com.mAadhar.repository.AadharRepository;
```

```
@Service
```

```
public class AadharService {
```

```
    @Autowired
```

```
    AadharRepository aadharRepository;
```

```
    public String storeAadhar(Aadhar aadhar) {
```

```
        aadharRepository.save(aadhar);
```

```
        return "Aadhar Details Submitted Successfully";
```

```
    }
```

```

public List<Aadhar> getAllAadhar(){
    return aadharRepository.findAll();
}

public String findAadharById(int cid) {
    Optional<Aadhar> result = aadharRepository.findById(cid);
    if(result.isPresent()) {
        Aadhar a = result.get();
        return a.toString();
    }else {
        return "Aadhar Details not found";
    }
}

public String deleteAadhar(int cid) {
    Optional<Aadhar> result = aadharRepository.findById(cid);
    if(result.isPresent()) {
        Aadhar a = result.get();
        aadharRepository.delete(a);
        return "Aadhar Deleted Succesfully";
    }else {
        return "Aadhar details didn't match with record";
    }
}

public String updateAadhar(Aadhar aadhar) {
    Optional<Aadhar> result =
aadharRepository.findById(aadhar.getCid());
    if(result.isPresent()) {
        Aadhar a = result.get();
        a.setCname(aadhar.getCname());
    }
}

```

```

        a.setAddress(aadhar.getAddress());
        a.setDob(aadhar.getDob());
        a.setEmailid(aadhar.getEmailid());
        a.setNumber(aadhar.getNumber());
        aadharRepository.saveAndFlush(a);
        return "Aadhar Details Updated Succesfully";
    }else {
        return "Aadhar details not present";
    }
}
}

```

LoginService.java

```

package com.mAadhar.service;

import java.util.Optional;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.mAadhar.bean.Login;
import com.mAadhar.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.findByEmail(login.getEmail());
        if(result.isPresent()) {
            return "Email Id already exists";
        }else {
            if(login.getTypeOfUser().equals("admin")) {
                return "You can't create admin account";
            }else {
                loginRepository.save(login);
                return "Account created successfully";
            }
        }
    }
}

```

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>mAadhar App</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>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
        </plugin>
    </plugins>
</build>

</project>
```

FrontEnd

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 { AppRoutingModuleModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { LoginComponent } from './login/login.component';
import { SignupComponent } from './signup/signup.component';
import { AdmindashboardComponent } from './admindashboard/
admindashboard.component';
import { UserdashboardComponent } from './userdashboard/
userdashboard.component';
import { AddAadharComponent } from './add-aadhar/add-
aadhar.component';
import { AdminAadharViewComponent } from './admin-aadhar-view/
admin-aadhar-view.component';
import { UpdateAadharComponent } from './update-aadhar/update-
aadhar.component';
import { DeleteAadharComponent } from './delete-aadhar/delete-
aadhar.component';
```

```
@NgModule({
  declarations: [
    AppComponent,
```



```

    LoginComponent,
    SignupComponent,
    AdminDashboardComponent,
    UserDashboardComponent,
    AddAadharComponent,
    AdminAadharViewComponent,
    UpdateAadharComponent,
    DeleteAadharComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule, HttpClientModule, FormsModule,
    ReactiveFormsModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }

```

app.routing modules

```

import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { AddAadharComponent } from './add-aadhar/add-aadhar.component';
import { AdminAadharViewComponent } from './admin-aadhar-view/admin-aadhar-view.component';

```

```
import { AdmindashboardComponent } from './admindashboard/
admindashboard.component';

import { DeleteAadharComponent } from './delete-aadhar/delete-
aadhar.component';

import { LoginComponent } from './login/login.component';

import { SignupComponent } from './signup/signup.component';

import { UpdateAadharComponent } from './update-aadhar/update-
aadhar.component';

import { UserdashboardComponent } from './userdashboard/
userdashboard.component';
```

```
const routes: Routes = [
  {path: "login", component:LoginComponent},
  {path:"adminHome",component:AdmindashboardComponent},
  {path:"userHome",component:UserdashboardComponent},
  {path:"", redirectTo:"login",pathMatch:"full"},
  {path:"applyforaadhar",component:AddAadharComponent},
  {path: "signUp", component:SignupComponent},
  {path:"updateAadhar", component:UpdateAadharComponent},
  {path:"deleteAadhar",component:DeleteAadharComponent},
  //{path:"adminHome",component:AdminAadharViewComponent}
  {path:"findAllAadhar",component:AdminAadharViewComponent}
];
```

```
@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```

Sign Up page(Typescript)

```
import { Component, OnInit } from '@angular/core';
import { FormControl, FormGroup } 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")
    })
}

}

```

login page (typescript)

```

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 Successfully 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")
  })
}
}

```

Admin dashborad

```

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

```

```
@Component({
  selector: 'app-admindashboard',
  templateUrl: './admindashboard.component.html',
  styleUrls: ['./admindashboard.component.css']
})
export class AdmindashboardComponent implements OnInit {
  user:string="";
  constructor() { }

  ngOnInit(): void {
    let obj = sessionStorage.getItem("userDetails");
    if(obj!=null){
      this.user=obj;
    }
  }
}
```