

## SOURCE CODE: FSD JAVA CAPSTONE PROJECT PROBLEM STATEMENT - M-AADHARAPPLICATION – INSIST

## FRONT-END:

### Add-request.component.html:

```
<div>  
    <h2>Apply for Aadhar Card</h2><br>  
    <form [formGroup]="reqRef" (ngSubmit)="storeRequest()">  
        <!-- <label>Adharid</label>  
        <input type="text" formControlName="adharid"><br/> -->  
        <label>Email</label>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~  
        <input type="text" formControlName="emailid"><br/><br>  
        <label>Name</label>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~  
        <input type="text" formControlName="name"><br/><br>  
        <label>Age</label>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~  
        <input type="number" formControlName="age"><br/><br>  
        <label>Address</label>&nbsp;&nbsp;&~  
        <input type="text" formControlName="address"><br/><br>  
        <input type="submit" class="btn btn-success"  
value="APPLY"/>&nbsp;&nbsp;&~  
        <input type="reset" class="btn btn-dark" value="RESET"/><br/>  
    </form><br/>  
    <span style="color:red">{{storeMsg}}</span>  
</div>
```

### Add-request.component.ts:

```
import { Component, OnInit } from '@angular/core';
import { FormGroup, FormControl } from '@angular/forms'
import { RequestsService } from '../requests.service';
@Component({
  selector: 'app-add-request',
  templateUrl: './add-request.component.html',
  styleUrls: ['./add-request.component.css']
})
export class AddRequestComponent implements OnInit {

  reqRef = new FormGroup({
    // adharid:new FormControl(),
    emailid:new FormControl(),
    name:new FormControl(),
    age:new FormControl(),
    address:new FormControl()
  })

  storeMsg :string = ""
  constructor(public rs:RequestsService) { }
```

```
ngOnInit(): void {
}

storeRequest() {
  let request = this.reqRef.value;
  console.log(request);
  this.rs.storeRequest(request).subscribe({
    next:(result:any)=>this.storeMsg=result,
    error:(error:any)=>console.log(error),
    complete:()=>console.log("completed")
  })

  this.reqRef.reset();
}
}
```

### Admindashboard.component.html:

```
<div>  
    <h2>Welcome to home page admin {{user}}</h2>  
    <a routerLink="viewReq" class="btn btn-primary">View All  
Request</a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~  
    <a routerLink="viewall" class="btn btn-primary">Display all issued Aadhaar  
Card</a>  
    <br/>  
    <hr/>  
    <router-outlet></router-outlet>  
    <hr/>  
  
    <br/>  
    <input type="button" class="btn btn-dark" value="logout"  
(click)="logout()"/>  
</div>
```

### 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(private router:Router) { }

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



### displayalladhar.component.ts:

```
import { Component, OnInit } from '@angular/core';
import { Requests } from '../requests';
import { RequestsService } from '../requests.service';

@Component({
  selector: 'app-displayalladhar',
  templateUrl: './displayalladhar.component.html',
  styleUrls: ['./displayalladhar.component.css']
})
export class DisplayalladharComponent implements OnInit {
  requests: Array<Requests>=[]
  constructor(public rs:RequestsService) { }

  ngOnInit(): void {
    this.viewAllRequest();
  }
  viewAllRequest(){
    this.rs.findAllRequest().subscribe({
      next:(result:any)=>this.requests=result,
      error:(error:any)=>console.log(error),
      complete:()=>console.log("completed")

    })
  }
}
```

### Login.component.html:

[illegible]

</div>

### Login.component.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)=>{
        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.component.html:

```
<div>
  <h3>Create Account</h3>
  <form [formGroup]="signUpRef" (ngSubmit)="signUp()">
    <label>EmailId </label>&nbsp;
    <input type="email" formControlName="emailid"/><br/><br>
    <label>Password </label>&nbsp;
    <input type="password" formControlName="password"/><br/><br>
    <label>Type Of User</label>&nbsp;&nbsp;&nbsp;&nbsp;
    <input type="radio" name="typeOfUser" value="admin"
formControlName="typeOfUser"/>Admin&nbsp;
    <input type="radio" name="typeOfUser" value="user"
formControlName="typeOfUser"/>User<br/><br>
    <input type="submit" class="btn btn-success" value="SIGNUP">&nbsp;
    <input type="reset" class="btn btn-dark" value="RESET">
  </form>
  <a href="javascript:void(0)" (click)="goBack()">
    <-- Go to Login Page
  </a>
  <br/><br>
  <span style="color:red">{{msg}}</span>
  <span style="color:green;">{{user}}</span>
</div>
```

### Sinup.component.ts:

```
import { Component, OnInit } from '@angular/core';
import { FormGroup, FormControl } from '@angular/forms';
import { Router } from '@angular/router';
import { LoginService } from '../login.service';
import { timer } from 'rxjs';
import { Location } from '@angular/common';
@Component({
  selector: 'app-signup',
  templateUrl: './signup.component.html',
  styleUrls: ['./signup.component.css']
})
export class SignupComponent implements OnInit {
  signUpRef = new FormGroup({
    emailid: new FormControl(),
    password: new FormControl(),
    typeOfUser: new FormControl()
  });
  msg: string = ""
  constructor(public ls: LoginService, public router: Router, private location:
Location) { }
  user: string = "";
  ngOnInit(): void {

  }
}
```







[illegible]

```

    </span>
    <br/><br>
    <span style="color:red">{{msg}}</span>
</div>

```

### View-request.component.ts:

```

import { Component, OnInit } from '@angular/core';
import { Requests } from '../requests';
import { RequestsService } from '../requests.service';

@Component({
  selector: 'app-view-request',
  templateUrl: './view-request.component.html',
  styleUrls: ['./view-request.component.css']
})
export class ViewRequestComponent implements OnInit {
  msg:string=""
  requests:Array<Requests>=[]

  flag:boolean = false;
  adharid:string=""
  emailid:string=""
  name:string=""
  age:number=0
  address:string=""

  constructor(public rs:RequestsService) { }

  ngOnInit(): void {

    this.viewAllRequest();
  }

  viewAllRequest(){
    this.rs.findAllRequest().subscribe({
      next:(result:any)=>this.requests=result,
      error:(error:any)=>console.log(error),
      complete:()=>console.log("completed")
    })
  }

  denyRequest(adharid:string){
    //console.log(pid)
    this.rs.denyRequest(adharid).subscribe({
      next:(result:any)=>this.msg=result,
      error:(error:any)=>console.log(error),

```



[illegible]

```

        <input type="reset" class="btn btn-dark" value="reset" /><br />
    </form>
    <br />
</div>
</div>

```

### Viewadhar.component.ts:

```

import { Component, OnInit } from '@angular/core';
import { Requests } from '../requests';
import { RequestsService } from '../requests.service';

@Component({
  selector: 'app-viewadhar',
  templateUrl: './viewadhar.component.html',
  styleUrls: ['./viewadhar.component.css']
})
export class ViewadharComponent implements OnInit {
  msg:string=""
  requests:Array<Requests>=[]
  myadhar:any
  flag:boolean = false;
  adharid:string=""
  emailid:string=""
  name:string=""
  age:number=0
  address:string=""
  constructor(public rs:RequestsService) { }
  ngOnInit(): void {
    let obj = sessionStorage.getItem("userDetails");
    if(obj!=null){
      this.emailid=obj;
    }
    this.viewmyadhar(this.emailid);
  }

  approvereq(request:any){
    this.flag= true;
    this.adharid=request.adharid;
    this.emailid=request.emailid;
    this.name=request.name;
    this.age=request.age;
    this.address=request.address;
  }

  viewmyadhar(emailid:string){
    this.rs.viewmyAdhar(this.emailid).subscribe({
      next:(result:any)=>this.myadhar=JSON.parse(result),
      error:(error:any)=>console.log(error),
    })
  }
}

```

```

        complete:()=>console.log("completed")

    })

}

saveReqtoDB(){
    let request =
{adharid:this.adharid,emailid:this.emailid,name:this.name,age:this.age,address
:this.address};
    this.rs.updateAdhar(request).subscribe({
        next:(result:any)=>this.msg=result,
        error:(error:any)=>console.log(error),
        complete:()=>{"completed"
        }
    })
    this.flag=false;
}
}

```

#### App.component.html:

```

<div align="center">
  <h2 align="left" >
     <b><em>mAdhar
Application</em></b> </h2>
    <hr />
    <router-outlet></router-outlet>
  </div>

```

#### App.component.ts:

```

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

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'mAdharFrontend-app';
}

```

#### 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 { UserdashboardComponent } from
'./userdashboard/userdashboard.component';
import { AdmindashboardComponent } from
'./admindashboard/admindashboard.component';
import { AddRequestComponent } from './add-request/add-request.component';
import { ViewRequestComponent } from './view-request/view-request.component';
import { ViewadharComponent } from './viewadhar/viewadhar.component';
import { DisplayalladharComponent } from
'./displayalladhar/displayalladhar.component';

@NgModule({
  declarations: [
    AppComponent,
    LoginComponent,
    SignupComponent,
    UserdashboardComponent,
    AdmindashboardComponent,
    AddRequestComponent,
    ViewRequestComponent,
    ViewadharComponent,
    DisplayalladharComponent,
  ],
  imports: [
    BrowserModule,
    AppRoutingModuleModule,HttpClientModule,FormsModule,ReactiveFormsModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }

```

#### 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:9191/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"});
    }
}

```

### Request.service.ts:

```

import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { Observable } from 'rxjs';
import { Requests } from './requests';
@Injectable({
    providedIn: 'root'
})
export class RequestsService {
    baseUrl:string ="http://localhost:9191/request"
    constructor(public http:HttpClient) { }

    storeRequest(requests:any):Observable<string> {
        return
this.http.post(this.baseUrl+"/apply",requests,{responseType:"text"});
    }

    findAllRequest():Observable<Requests[]> {
        return this.http.get<Requests[]>(this.baseUrl+"/viewAllRequest");
    }
    denyRequest(adharid:string):Observable<string> {
        return
this.http.delete(this.baseUrl+"/deny/"+adharid,{responseType:"text"});
    }

    approvereq(request:any):Observable<string> {
        return
this.http.patch(this.baseUrl+"/approve",request,{responseType:"text"});
    }

    viewmyAdhar(emailid:String):Observable<string> {
        return
this.http.get(this.baseUrl+"/viewmyadhar/"+emailid,{responseType:"text"});
    }

    updateAdhar(request:any):Observable<string> {

```



```

        return
    this.http.patch(this.baseUrl+"/updateadhar",request,{responseType:"text"});
    }
}

```

## BACK-END:

### MAadharbackendAppApplication.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.mAadhar.bean")
@EnableJpaRepositories(basePackages = "com.mAadhar.repository")
public class MAadharbackendAppApplication {

    public static void main(String[] args) {
        SpringApplication.run(MAadharbackendAppApplication.class, args);
        System.out.println("Server Running on 9191");
    }

}

```

### 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 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;
    }
}

```

### Requests.java:

```

package com.mAadhar.bean;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

import org.hibernate.annotations.GenericGenerator;
@Entity
public class Requests {

    @GeneratedValue(generator = "uuid")
    @GenericGenerator(name = "uuid", strategy = "uuid")
    @Column(columnDefinition = "CHAR(32)")
    @Id
    private String adharid;
    private String emailid;
    private String name;
    private int age;
    private String address;
    public String getEmailid() {
        return emailid;
    }
    public void setEmailid(String emailid) {
        this.emailid = emailid;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public int getAge() {
        return age;
    }
    public void setAge(int age) {
        this.age = age;
    }
    public String getAddress() {
        return address;
    }
    public void setAddress(String address) {
        this.address = address;
    }
    public String getAdharid() {
        return adharid;
    }
    public void setAdharid(String adharid) {
        this.adharid = adharid;
    }
}
@Override

```

```

        public String toString() {
            return "Requests [adharid=" + adharid + ", emailid=" + emailid + ",
name=" + name + ", age=" + age
                + ", address=" + address + "]";
        }
    }
}

```

#### 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) {
        return loginService.signUp(login);
    }
}

```

#### RequestController.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.mAdhar.bean.Requests;
import com.mAdhar.service.RequestService;

@RestController
@RequestMapping("request")
@CrossOrigin
public class RequestController {

    @Autowired
    RequestService reqservice;

    @PostMapping(value = "apply", consumes = MediaType.APPLICATION_JSON_VALUE)
    public String requestadhar(@RequestBody Requests req) {
        return reqservice.storeRequest(req);
    }

    @GetMapping(value="viewAllRequest", produces =
MediaType.APPLICATION_JSON_VALUE)
    public List<Requests> getAllReq() {
        return reqservice.getAllRequests();
    }

    @DeleteMapping(value="deny/{adharid}")
    public String deleteReq(@PathVariable("adharid") String adharid) {
        return reqservice.denyRequest(adharid);
    }

    @PatchMapping(value = "approve", consumes =
MediaType.APPLICATION_JSON_VALUE)
    public String approveRequest(@RequestBody Requests req) {
        return reqservice.approveRequest(req);
    }

    @PatchMapping(value = "updateadhar", consumes =
MediaType.APPLICATION_JSON_VALUE)
    public String updateadhar(@RequestBody Requests req) {
        return reqservice.updateAdhar(req);
    }

    @GetMapping(value="viewmyadhar/{emailid}", produces =
MediaType.APPLICATION_JSON_VALUE)
    public Requests viewmyAdhar(@PathVariable("emailid") String emailid) {
        return reqservice.viewmyAdhar(emailid);
    }
}

```

#### LoginRepository.java:

```

package com.mAdhar.repository;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

import com.mAdhar.bean.Login;

@Repository
public interface LoginRepository extends JpaRepository<Login, String>{

}

```

### RequestRepository.java:

```
package com.mAdhar.repository;

import java.util.List;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.query.Param;
import org.springframework.stereotype.Repository;

import com.mAdhar.bean.Requests;

@Repository
public interface RequestRepository extends JpaRepository<Requests, String>{

    //JPQL
    @Query("select r from Requests r where r.emailid = :emailid")
    public Requests viewmyAdhar(@Param("emailid") String emailid);
}
```

### LoginService.java:

```
package com.mAdhar.service;

import java.util.Optional;

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

import com.mAdhar.bean.Login;
import com.mAdhar.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())) {

            if(login.getTypeOfUser().equals("admin")) {
return "Admin sucessfully
login";
}
else{
return "User successfully
login";
}
}else {
return "Invalid Credentials! Try
Again!";
}
}else {
```

```

        return "Invalid Password! Try Again!";
    }
    }else{
        return "Invalid EmailId! Try Again!";
    }
}

    public String signUp(Login login) {
        Optional<Login> result =
loginRepository.findById(login.getEmailid());
        if(result.isPresent()) {
            return "Email Id already exist";
        }else {
            if(login.getTypeOfUser().equals("admin")) {
                return "You can't create admin account";
            }else {
                loginRepository.save(login);
                return "Account created successfully";
            }
        }
    }
}

```

#### **RequestService.java:**

```

package com.mAdhar.service;

import java.util.List;
import java.util.Optional;

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

import com.mAdhar.bean.Requests;
import com.mAdhar.repository.RequestRepository;

@Service
public class RequestService {

    @Autowired
    RequestRepository reqrepo;

    public String storeRequest(Requests req) {
        reqrepo.save(req);
        return "Aadhar Application Request Submitted";
    }

    public List<Requests> getAllRequests() {
        return reqrepo.findAll();
    }

    public String denyRequest(String adharid) {
        Optional<Requests> result = reqrepo.findById(adharid);
        if(result.isPresent()) {
            Requests p = result.get();
            reqrepo.delete(p);
            return "Application Denied";
        }else {
            return "Application not present";
        }
    }
}

```

```

    }
    public String approveRequest(Requests req) {
        Optional<Requests> result = reqrepo.findById(req.getAdharid());
        if(result.isPresent()) {
            Requests p = result.get();
            reqrepo.saveAndFlush(p);
            return "Application Approved";
        }else {
            return "Application not present";
        }
    }

    public String updateAdhar(Requests req) {
        Optional<Requests> result = reqrepo.findById(req.getAdharid());
        if(result.isPresent()) {
            Requests p = result.get();
            p.setAge(req.getAge());
            p.setName(req.getName());
            p.setAddress(req.getAddress());
            reqrepo.saveAndFlush(p);
            return "Aadhar Card Updation Requested";
        }else {
            return "Aadhar not present";
        }
    }

    public Requests viewmyAdhar(String emailid) {
        return reqrepo.viewmyAdhar(emailid);
    }
}

```

#### Application.properties:

```

spring.datasource.username=root
spring.datasource.password=Ajitdk@26042000
spring.datasource.url=jdbc:mysql://localhost:3306/maadhar
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
server.port=8082

```

#### TESTING:

##### MAadharbackendAppApplicationTests.java:

```

package com;

import org.junit.jupiter.api.Test;
import org.springframework.boot.test.context.SpringBootTest;

@SpringBootTest
class MAadharbackendAppApplicationTests {

    @Test
    void contextLoads() {
    }

}

```

**LoginServiceTest.java:**

```
package com.mAdhar.testing;

import org.testng.annotations.Test;

import io.github.bonigarcia.wdm.WebDriverManager;

import org.testng.annotations.BeforeClass;
import org.testng.annotations.BeforeTest;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.chrome.ChromeOptions;
import org.testng.annotations.AfterClass;
import org.testng.annotations.AfterTest;

public class LoginServiceTest {
    WebDriver driver;

    @BeforeTest
    public void config() {
        WebDriverManager.chromedriver().setup();
        driver=new ChromeDriver();
        driver.manage().window().maximize();
    }

    @AfterTest
    public void afterClass() {
        driver.close();
    }

    @Test
    public void signInTest() {
        driver.get("http://localhost:4200/login");
    }

    @Test
    public void signUpTest() {
        driver.get("http://localhost:4200/signup");
    }
}
```

**RequestServiceTest.java:**

```
package com.mAdhar.testing;

import org.testng.annotations.Test;

import io.github.bonigarcia.wdm.WebDriverManager;

import org.testng.annotations.BeforeClass;
import org.testng.annotations.BeforeTest;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.chrome.ChromeOptions;
import org.testng.annotations.AfterClass;
import org.testng.annotations.AfterTest;
```



```

public class RequestServiceTest {
    WebDriver driver;

    @BeforeTest
    public void config() {
        WebDriverManager.chromedriver().setup();
        driver=new ChromeDriver();
        driver.manage().window().maximize();
    }

    @AfterTest
    public void afterClass() {
        driver.close();
    }

    @Test
    public void approveRequestTest() {
        driver.get("http://localhost:4200/approve");
    }

    @Test
    public void denyRequestTest() {
        driver.get("http://localhost:4200/deny");
    }

    @Test
    public void getAllRequestsTest() {
        driver.get("http://localhost:4200/veiwAll");
    }

    @Test
    public void storeRequestTest() {
        driver.get("http://localhost:4200/apply");
    }

    @Test
    public void updateAdharTest() {
        driver.get("http://localhost:4200/updateaadhar");
    }

    @Test
    public void viewmyAdharTest() {
        driver.get("http://localhost:4200/viewmyadhar");
    }
}

```

#### Docker File:

FROM openjdk:8

COPY target/mAadharbackend-app-0.0.1-SNAPSHOT.jar .

CMD ["java", "-jar", "mAadharbackend-app-0.0.1-SNAPSHOT.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>mAadharbackend-app</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>mAadharbackend-app</name>
  <description>Demo project for Spring Boot</description>
  <properties>
    <java.version>11</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>io.github.bonigarcia</groupId>
      <artifactId>webdrivermanager</artifactId>
      <version>5.3.0</version>
    </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>
    <dependency>
      <groupId>org.seleniumhq.selenium</groupId>
      <artifactId>selenium-java</artifactId>
      <version>4.2.1</version>
    </dependency>
    <dependency>
      <groupId>org.testng</groupId>
```

```

        <artifactId>testng</artifactId>
        <version>7.5</version>
        <scope>test</scope>
    </dependency>
</dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
                <configuration>
                    <excludes>
                        <exclude>
                            <groupId>org.projectlombok</groupId>
                            <artifactId>lombok</artifactId>
                        </exclude>
                    </excludes>
                </configuration>
            </plugin>
        </plugins>
    </build>
</project>

```

### Testing.xml:

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="Suite">
    <test thread-count="5" name="Test">
        <classes>
            <class name="com.mAadhar.testing.LoginServiceTest"/>
            <class name="com.mAadhar.testing.RequestServiceTest"/>
        </classes>
    </test> <!-- Test -->
</suite> <!-- Suite -->

```

### MySql:

create database maadhar;

use maadhar;