**Code:**

**Backend:**

**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>2.7.4</version>

<relativePath/> <!-- lookup parent from repository -->

</parent>

<groupId>com.example</groupId>

<artifactId>MAadharApp</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>MAadharApp</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>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>junit</groupId>

<artifactId>junit</artifactId>

<version>4.11</version>

<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>

</plugin>

</plugins>

</build>

</project>

**MAadharAppApplication.java:**

package com.aadhar;

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 MAadharAppApplication {

public static void main(String[] args) {

SpringApplication.*run*(MAadharAppApplication.class, args);

System.***out***.println("Server running on port number 9090");

}

}

**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 + "]";

}

}

**User.Java:**

package com.aadhar.bean;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

*@Entity*

public class User {

*@Id*

*@Column*(name="id")

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

private int id;

private String dob;

private String name;

private String address;

private String email;

private String mobile;

private String gender;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getDob() {

return dob;

}

public void setDob(String dob) {

this.dob = dob;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getAddress() {

return address;

}

public void setAddress(String address) {

this.address = address;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getMobile() {

return mobile;

}

public void setMobile(String mobile) {

this.mobile = mobile;

}

public String getGender() {

return gender;

}

public void setGender(String gender) {

this.gender = gender;

}

}

**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);

}

}

**UserController.java:**

package com.aadhar.controller;

import java.util.List;

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

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

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.PathVariable;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.PutMapping;

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.User;

import com.aadhar.service.UserService;

*@CrossOrigin*

*@RestController*

*@RequestMapping*("/api/user")

public class UserController {

*@Autowired*

private UserService service;

*@PostMapping*("/")

public ResponseEntity<User> adduser(*@RequestBody* User u){

User user= service.addUser(u);

if(user!=null)

return new ResponseEntity<User>(user,*HttpStatus*.***CREATED***);

else

return new ResponseEntity<User>(user,*HttpStatus*.***INTERNAL\_SERVER\_ERROR***);

}

*@GetMapping*("/")

public List<User> getAllUser(){

return service.getAllUser();

}

*@GetMapping*("/{id}")

public ResponseEntity<User>getUserById (*@PathVariable* int id){

User user= service.getUserById(id);

if(user!=null)

return new ResponseEntity<User>(user, *HttpStatus*.***FOUND***);

else

return new ResponseEntity<User>(user, *HttpStatus*.***NOT\_FOUND***);

}

*@PutMapping*("/{id}")

public ResponseEntity<Object> updateUser(*@PathVariable* int id,*@RequestBody* User newUser){

User user= service.updateUser(id, newUser);

if (user!=null)

return new ResponseEntity<Object>(user,*HttpStatus*.***OK***);

else

return new ResponseEntity<Object>("No User Available to Update",*HttpStatus*.***NOT\_FOUND***);

}

*@DeleteMapping*("/{id}")

public ResponseEntity<String>deleteUser(*@PathVariable* int id){

boolean result = service.deleteUser(id);

if(result)

return new ResponseEntity<String>("Object Deleted",*HttpStatus*.***OK***);

else

return new ResponseEntity<String>("NO user Found", *HttpStatus*.***NOT\_FOUND***);

}

}

**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*

public 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";

}

}

}

}

**UserService.java:**

package com.aadhar.service;

import java.util.List;

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

import org.springframework.stereotype.Service;

import com.aadhar.bean.User;

import com.aadhar.repository.UserRepo;

*@Service*

public class UserService {

*@Autowired*

private UserRepo repo;

public User addUser(User u) {

return repo.save(u);

}

public List<User> getAllUser(){

return repo.findAll();

}

public User getUserById(int id) {

if(repo.findById(id).isPresent()) {

return repo.findById(id).get();

}

else {

return null;

}

}

public User updateUser(int id, User newUser) {

if(repo.findById(id).isPresent()) {

User olduser= repo.findById(id).get();

olduser.setName(newUser.getName());

olduser.setDob(newUser.getDob());

olduser.setAddress(newUser.getAddress());

olduser.setEmail(newUser.getEmail());

olduser.setMobile(newUser.getMobile());

olduser.setGender(newUser.getGender());

return repo.save(olduser);

}

else {

return null;

}

}

public boolean deleteUser(int id) {

if(repo.findById(id).isPresent()) {

repo.deleteById(id);

return true;

}

else {

return false;

}

}

}

**LoginRepository.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>{

}

**UserRepo.java:**

package com.aadhar.repository;

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

import org.springframework.stereotype.Repository;

import com.aadhar.bean.User;

*@Repository*

public interface UserRepo extends JpaRepository<User,Integer> {

}

**Application.properties:**

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.datasource.url=jdbc:mysql://localhost:3306/capstone

spring.datasource.username=root

spring.datasource.password=1204@koushi

spring.jpa.hibernate.ddl-auto=update

server.port=9090

**Testing:**

**MAadharAppApplicationTests.java:**

package com.aadhar;

import org.junit.jupiter.api.Test;

import org.springframework.boot.test.context.SpringBootTest;

*@SpringBootTest*

class MAadharAppApplicationTests {

*@Test*

void contextLoads() {

}

}

**MAadharAppApplicationTestsCode.java:**

package com.aadhar;

import org.testng.annotations.AfterMethod;

import org.testng.annotations.BeforeMethod;

import org.testng.annotations.Test;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class MAadharAppApplicationTestsCode {

WebDriver driver;

*@Test* (priority=0)

public void testing() {

long start = System.*currentTimeMillis*();

driver.manage().window().maximize();

driver.get("http://localhost:4200/home");

long finish = System.*currentTimeMillis*();

long totalTime = finish - start;

System.***out***.println("Total Time for main page load = "+(totalTime\*0.001)+" Seconds");

}

*@Test* (priority=1)

public void signup() {

driver.get("http://localhost:4200/signUp");

driver.findElement(By.*xpath*("//\*[@id=\"userName\"]")).sendKeys("u1");

driver.findElement(By.*xpath*("//\*[@id=\"userPassword\"]")).sendKeys("u1");

driver.findElement(By.*xpath*("/html/body/app-root/app-signup/div/div/form/input[3]")).click();

System.***out***.println("sucessfully signUp ");

}

*@Test* (priority=2)

public void signin() {

driver.get("http://localhost:4200/login");

driver.findElement(By.*xpath*("//\*[@id=\"userName\"]")).sendKeys("admin");

driver.findElement(By.*xpath*("//\*[@id=\"userPassword\"]")).sendKeys("admin");

driver.findElement(By.*xpath*("/html/body/app-root/app-login/div/div/form/input[3]")).click();

System.***out***.println("sucessfully logged in");

}

*@Test* (priority=4)

public void adminhome() {

long start = System.*currentTimeMillis*();

driver.manage().window().maximize();

driver.get("http://localhost:4200/adminHome");

long finish = System.*currentTimeMillis*();

long totalTime = finish - start;

System.***out***.println("Total Time for admin home page load = "+(totalTime\*0.001)+" Seconds");

}

*@Test* (priority=5)

public void userhome() {

long start = System.*currentTimeMillis*();

driver.manage().window().maximize();

driver.get("http://localhost:4200/userHome");

long finish = System.*currentTimeMillis*();

long totalTime = finish - start;

System.***out***.println("Total Time for user home page load = "+(totalTime\*0.001)+" Seconds");

}

*@Test* (priority=6)

public void loadusers() {

long start = System.*currentTimeMillis*();

driver.manage().window().maximize();

driver.get("http://localhost:4200/allusers");

long finish = System.*currentTimeMillis*();

long totalTime = finish - start;

System.***out***.println("Total Time for allusers page load = "+(totalTime\*0.001)+" Seconds");

}

*@Test* (priority=7)

public void approve() {

long start = System.*currentTimeMillis*();

driver.manage().window().maximize();

driver.get("http://localhost:4200/approve");

long finish = System.*currentTimeMillis*();

long totalTime = finish - start;

System.***out***.println("Total Time for approve page load = "+(totalTime\*0.001)+" Seconds");

}

*@BeforeMethod*

public void beforeMethod() {

System.*setProperty*("webdriver.chrome.driver", "C:\\\\Users\\\\durgasindhu\\\\Downloads\\\\chromedriver\_win32/chromedriver.exe");

driver = new ChromeDriver();

}

*@AfterMethod*

public void afterMethod() {

//driver.close();

driver = null;

}

}

**Test.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.aadhar.MAadharAppApplicationTestsCode"*/>

</**classes**>

</**test**> <!-- Test -->

</**suite**> <!-- Suite -->

**FrontEnd:**

**Index.html:**

<!doctype html>

<html lang="en">

<head>

    <meta charset="utf-8">

    <title>FrontendApp</title>

    <base href="/">

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel="icon" type="image/x-icon" href="favicon.ico">

    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min.css" integrity="sha384-xOolHFLEh07PJGoPkLv1IbcEPTNtaed2xpHsD9ESMhqIYd0nLMwNLD69Npy4HI+N" crossorigin="anonymous">

</head>

<body>

    <app-root></app-root>

    <script src="https://cdn.jsdelivr.net/npm/jquery@3.5.1/dist/jquery.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-Fy6S3B9q64WdZWQUiU+q4/2Lc9npb8tCaSX9FK7E8HnRr0Jz8D6OP9dO5Vg3Q9ct" crossorigin="anonymous"></script>

    <!-- JavaScript Bundle with Popper -->

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3" crossorigin="anonymous"></script>

</body>

</html>

**Admindashboard(component)**

**Admindashboard.component.html:**

<div class="fluid container mt-5 ">

    <h1 class="text-center">Welcome Aadhaar Admin</h1><br><br>

    <div class="row text-center">

      <div class="col text-center">

        <div class="card text-white bg-success mb-3" style="max-width: 18rem;">

            <div class="card-body">

              <h5 class="card-title"><a routerLink="/allusers" style="color:white ;">List All Users</a></h5>

              <p class="card-text">Here we Can view all the available Users in System</p>

            </div>

      </div>

      </div>

      <div class="col text-center">

        <div class="card text-white bg-info mb-3" style="max-width: 18rem;">

            <div class="card-body">

              <h5 class="card-title"><a routerLink="/approve" style="color:white ;">Approve New Users</a></h5>

              <p class="card-text">Here we can approve and review modifications</p>

            </div>

      </div>

      </div>

    </div>

  </div>

**Admindashboard.component.spec.ts:**

import { ComponentFixture, TestBed } from '@angular/core/testing';

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

describe('AdmindashboardComponent', () => {

  let component: AdmindashboardComponent;

  let fixture: ComponentFixture<AdmindashboardComponent>;

  beforeEach(async () => {

    await TestBed.configureTestingModule({

      declarations: [ AdmindashboardComponent ]

    })

    .compileComponents();

    fixture = TestBed.createComponent(AdmindashboardComponent);

    component = fixture.componentInstance;

    fixture.detectChanges();

  });

  it('should create', () => {

    expect(component).toBeTruthy();

  });

});

**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){

      this.user=obj;

    }

  }

}

**Allusers(component):**

**Allusers.component.html:**

<div class="container mt-5">

    <table class="table">

        <thead class="thead-dark">

          <tr>

            <th scope="col">Id</th>

            <th scope="col">Name</th>

            <th scope="col">DOB</th>

            <th scope="col">Address</th>

            <th scope="col">Email id</th>

            <th scope="col">Mobile Number </th>

            <th scope="col">Gender</th>

          </tr>

        </thead>

        <tbody>

            <tr \*ngFor="let u of users">

                <td>{{u.id}}</td>

                <td>{{u.name}}</td>

                <td>{{u.dob}}</td>

                <td>{{u.address}}</td>

                <td>{{u.email}}</td>

                <td>{{u.mobile}}</td>

                <td>{{u.gender}}</td>

            </tr>

        </tbody>

      </table>

</div>

**Allusers.component.spec.ts:**

import { ComponentFixture, TestBed } from '@angular/core/testing';

import { AllusersComponent } from './allusers.component';

describe('AllusersComponent', () => {

  let component: AllusersComponent;

  let fixture: ComponentFixture<AllusersComponent>;

  beforeEach(async () => {

    await TestBed.configureTestingModule({

      declarations: [ AllusersComponent ]

    })

    .compileComponents();

    fixture = TestBed.createComponent(AllusersComponent);

    component = fixture.componentInstance;

    fixture.detectChanges();

  });

  it('should create', () => {

    expect(component).toBeTruthy();

  });

});

**Allusers.component.ts:**

import { ProductService } from './../product.service';

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

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

import { Observable } from 'rxjs';

import { UserClass } from '../UserClass';

@Component({

  selector: 'app-allusers',

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

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

})

export class AllusersComponent implements OnInit {

  allusers:Observable<UserClass[]>;

  constructor(private router:Router, private service:ProductService) { }

  users:UserClass[];

  ngOnInit(): void {

    this.getusers();

  }

  getusers(){

    this.service.getAllUser().subscribe(result=>this.users=result);

  }

}

**Editusers(component)**

**Editusers.component.html:**

  <!-- Modal -->

  <div class="modal fade" id="exampleModal" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">

    <div class="modal-dialog" role="document">

      <div class="modal-content">

        <div class="modal-header">

          <h5 class="modal-title" id="exampleModalLabel">Sucess</h5>

          <button type="button" class="close" data-dismiss="modal" aria-label="Close">

            <span aria-hidden="true">&times;</span>

          </button>

        </div>

        <div class="modal-body">

           <h1>Your Request Processed Sucessfully</h1>

        </div>

        <div class="modal-footer">

          <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>

        </div>

      </div>

    </div>

  </div>

  <div class="container mt-5">

    <table class="table">

        <thead class="thead-dark">

          <tr>

            <th scope="col">Id</th>

            <th scope="col">Name</th>

            <th scope="col">DOB</th>

            <th scope="col">Address</th>

            <th scope="col">Email id</th>

            <th scope="col">Mobile Number </th>

            <th scope="col">Gender</th>

            <th scope="col">Action</th>

          </tr>

        </thead>

        <tbody>

            <tr \*ngFor="let u of users">

                <td>{{u.id}}</td>

                <td>{{u.name}}</td>

                <td>{{u.dob}}</td>

                <td>{{u.address}}</td>

                <td>{{u.email}}</td>

                <td>{{u.mobile}}</td>

                <td>{{u.gender}}</td>

                <td>

                <div class="row">

                  <div class="col"><button  type="button" class="btn btn-success" data-toggle="modal" data-target="#exampleModal">Approve</button></div>

                  <div class="col"><button type="button" (click)='BookDelete(u.id)' class="btn btn-danger">Delete</button></div>

             </div>

            </td>

            </tr>

        </tbody>

      </table>

</div>

**Editusers.component.spec.ts:**

import { ComponentFixture, TestBed } from '@angular/core/testing';

import { EditusersComponent } from './editusers.component';

describe('EditusersComponent', () => {

  let component: EditusersComponent;

  let fixture: ComponentFixture<EditusersComponent>;

  beforeEach(async () => {

    await TestBed.configureTestingModule({

      declarations: [ EditusersComponent ]

    })

    .compileComponents();

    fixture = TestBed.createComponent(EditusersComponent);

    component = fixture.componentInstance;

    fixture.detectChanges();

  });

  it('should create', () => {

    expect(component).toBeTruthy();

  });

});

**Editusers.component.ts:**

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

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

import { Observable } from 'rxjs';

import { ProductService } from '../product.service';

import { UserClass } from '../UserClass';

@Component({

  selector: 'app-editusers',

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

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

})

export class EditusersComponent implements OnInit {

  allusers:Observable<UserClass[]>;

  constructor(private router:Router, private service:ProductService) { }

  users:UserClass[];

  ngOnInit(): void {

    this.getusers();

  }

  getusers(){

    this.service.getAllUser().subscribe(result=>this.users=result);

  }

  BookDelete(id:number){

    this.service.deletePost(id);

    console.log("user deleted");

    this.onReload();

}

onReload(){

  this.router.navigate(['/approve']);

  this.ngOnInit();

}

}

**Footer(component):**

**Footer.component.html:**

  <style media="screen">

    footer{

        position:absolute;

        bottom: 0;

        left: 0;

        right: 0;

        background: #111;

        height: auto;

        width: 100vw;

        color: #fff;

        padding: top 5px;;

    }

    .footer-content{

        display: flex;

        align-items: center;

        justify-content: center;

        flex-direction: column;

        text-align: center;

    }

    .footer-content p{

        max-width: 500px;

        line-height: 20px;

        font-size: 14px;

        color: #cacdd2;

    }

    @media (max-width:500px) {

    .footer-menu ul{

      display: flex;

      margin-top: 10px;

      margin-bottom: 20px;

    }

    }

        </style>

        <footer>

            <div class="footer-content">

                <h4>M-Aadhaar</h4>

                <p>copyright &copy; <a href="#"> Developers</a>  </p>

            </div>

        </footer>

**Footer.component.spec.ts:**

import { ComponentFixture, TestBed } from '@angular/core/testing';

import { FooterComponent } from './footer.component';

describe('FooterComponent', () => {

  let component: FooterComponent;

  let fixture: ComponentFixture<FooterComponent>;

  beforeEach(async () => {

    await TestBed.configureTestingModule({

      declarations: [ FooterComponent ]

    })

    .compileComponents();

    fixture = TestBed.createComponent(FooterComponent);

    component = fixture.componentInstance;

    fixture.detectChanges();

  });

  it('should create', () => {

    expect(component).toBeTruthy();

  });

});

**Footer.component.ts:**

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

@Component({

  selector: 'app-footer',

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

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

})

export class FooterComponent implements OnInit {

  constructor() { }

  ngOnInit(): void {

  }

}

**Header(component)**

**Header.component.html:**

<nav class="navbar navbar-expand-lg navbar-dark bg-primary">

    <div class="container-fluid">

      <a class="navbar-brand" href="#">M-Aadhaar</a>

      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

        <span class="navbar-toggler-icon"></span>

      </button>

      <div class="collapse navbar-collapse" id="navbarSupportedContent">

        <ul class="navbar-nav me-auto mb-2 mb-lg-0">

          <li class="nav-item">

            <a class="nav-link active" aria-current="page" routerLink="/home">Home</a>

          </li>

          <li class="nav-item">

            <a class="nav-link active" aria-current="page" routerLink="/adminHome"

            \*ngIf="isAdmin()"

            >Admin Dashboard</a>

          </li>

          <li class="nav-item">

            <a class="nav-link active" aria-current="page" routerLink="/userHome"

            \*ngIf="isUser()"

            >User Dashboard</a>

          </li>

        </ul>

       </div>

       <button \*ngIf="!isLoggedIn()" type="button" class="btn btn-success" routerLink="/login">Login</button>

       <button \*ngIf="isLoggedIn()" type="button" class="btn btn-danger" (click)="logout()" >Logout</button>

    </div>

  </nav>

**Header.component.spec.ts:**

import { ComponentFixture, TestBed } from '@angular/core/testing';

import { HeaderComponent } from './header.component';

describe('HeaderComponent', () => {

  let component: HeaderComponent;

  let fixture: ComponentFixture<HeaderComponent>;

  beforeEach(async () => {

    await TestBed.configureTestingModule({

      declarations: [ HeaderComponent ]

    })

    .compileComponents();

    fixture = TestBed.createComponent(HeaderComponent);

    component = fixture.componentInstance;

    fixture.detectChanges();

  });

  it('should create', () => {

    expect(component).toBeTruthy();

  });

});

**Header.component.ts:**

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

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

@Component({

  selector: 'app-header',

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

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

})

export class HeaderComponent implements OnInit {

  constructor(public router:Router) { }

  ngOnInit(): void {

  }

  isAdmin(){

    if(sessionStorage.getItem("userDetails")=="admin")

      return true;

  }

  isUser(){

    if(sessionStorage.getItem("userDetails")=="user")

       return true;

  }

  isLoggedIn(){

      if(sessionStorage.getItem("userDetails"))

          return true;

      else

         return false;

  }

  logout() {

    sessionStorage.removeItem("userDetails");

    this.router.navigate(["home"]);

  }

}

**Home(component)**

[**Home.component.html**](http://home.component.html/)**:**

<div class="container">

    <div class="text-center"><img src="assets/images/demo.png" class="img-fluid" alt="..."></div>

    <h3>Aadhaar is</h3>

    <ul class="aad-is-dv1">

       <li><span class="li-text">A 12-digit unique identity for every Indian individual, including children and infants</span></li>

       <li><span class="li-text">Enables identification for every resident Indian</span></li>

       <li><span class="li-text">Establishes uniqueness of every individual on the basis of demographic and biometric information</span></li>

       <li><span class="li-text">It is a voluntary service that every resident can avail irrespective of present documentation </span></li>

       <li><span class="li-text">Each individual will be given a single unique Aadhaar ID number</span></li>

       </ul>

 </div>

**Home.component.spec.ts:**

import { ComponentFixture, TestBed } from '@angular/core/testing';

import { HomeComponent } from './home.component';

describe('HomeComponent', () => {

  let component: HomeComponent;

  let fixture: ComponentFixture<HomeComponent>;

  beforeEach(async () => {

    await TestBed.configureTestingModule({

      declarations: [ HomeComponent ]

    })

    .compileComponents();

    fixture = TestBed.createComponent(HomeComponent);

    component = fixture.componentInstance;

    fixture.detectChanges();

  });

  it('should create', () => {

    expect(component).toBeTruthy();

  });

});

[**Home.component. HYPERLINK "http://home.component.ts/"ts**](http://home.component.ts/)**:**

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

@Component({

  selector: 'app-home',

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

  styleUrls: ['./home.component.css']

})

export class HomeComponent implements OnInit {

  constructor() { }

  ngOnInit(): void {

  }

}

**Login(component):**

**Login.component.html:**

<div class="container mt-5">

    <div class="card" style="padding: 20px;">

        <form [formGroup]="loginRef" (ngSubmit)="signIn()">

            <input type="text"

            ngModel

            name="userName"

            id="userName"

            placeholder="Enter your username"

            class="form-control mb-3"

            formControlName="emailid"

            >

            <input type="password"

            ngModel

            name="userPassword"

            id="userPassword"

            placeholder="Enter your password"

            class="form-control mb-3"

            formControlName="password"

            >

            <div class="row text-center">

                <div class="col"><input type="radio" name="typeOfUser" value="admin" formControlName="typeOfUser"/><h5>admin</h5>

                </div>

                <div class="col"><input type="radio" name="typeOfUser" value="user" formControlName="typeOfUser"/><h5>user</h5>

                </div>

            </div>

            <input type="submit"

             value="signIn"

            class="btn btn-success form-control rounded-0">

            <br><br>

            <input type="reset" value="reset"

            class="btn btn-primary form-control rounded-0">

        </form>

        <br>

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

        <a class="text-center" routerLink="/signUp"><h4>SignUp</h4></a>

    </div>

</div>

**Login.component.spec.ts:**

import { ComponentFixture, TestBed } from '@angular/core/testing';

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

describe('LoginComponent', () => {

  let component: LoginComponent;

  let fixture: ComponentFixture<LoginComponent>;

  beforeEach(async () => {

    await TestBed.configureTestingModule({

      declarations: [ LoginComponent ]

    })

    .compileComponents();

    fixture = TestBed.createComponent(LoginComponent);

    component = fixture.componentInstance;

    fixture.detectChanges();

  });

  it('should create', () => {

    expect(component).toBeTruthy();

  });

});

**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)=>{

        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(component)**

**Signup.component.html:**

<div class="container mt-5">

    <h2 class="text-center">Create your Account </h2>

    <div class="card" style="padding: 20px;">

        <form [formGroup]="loginRef" (ngSubmit)="signUp()">

            <input type="text"

            ngModel

            name="userName"

            id="userName"

            placeholder="Enter your username"

            class="form-control mb-3"

            formControlName="emailid"

            >

            <input type="password"

            ngModel

            name="userPassword"

            id="userPassword"

            placeholder="Enter your password"

            class="form-control mb-3"

            formControlName="password"

            >

            <div class="row text-center">

                <div class="col"><input type="radio" name="typeOfUser" value="admin" formControlName="typeOfUser"/><h5>admin</h5>

                </div>

                <div class="col"><input type="radio" name="typeOfUser" value="user" formControlName="typeOfUser"/><h5>user</h5>

                </div>

            </div>

            <input type="submit"

            value="signUp"

            class="btn btn-success form-control rounded-0">

            <br><br>

            <input type="reset" value="reset"

            class="btn btn-primary form-control rounded-0">

        </form>

        <br>

        <span style="color:red" class="text-center">{{msg}}</span>

        <a class="text-center" routerLink="/login"><h4>Login</h4></a>

    </div>

</div>

**Signup.component.spec.ts:**

import { ComponentFixture, TestBed } from '@angular/core/testing';

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

describe('SignupComponent', () => {

  let component: SignupComponent;

  let fixture: ComponentFixture<SignupComponent>;

  beforeEach(async () => {

    await TestBed.configureTestingModule({

      declarations: [ SignupComponent ]

    })

    .compileComponents();

    fixture = TestBed.createComponent(SignupComponent);

    component = fixture.componentInstance;

    fixture.detectChanges();

  });

  it('should create', () => {

    expect(component).toBeTruthy();

  });

});

**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(component)**

**Userdashboard.component.html:**

<style>

    .custom {

    width: 200px !important;

}

</style>

<div class="fluid-container text-center p-5 " style="background-color:rgb(222, 235, 208)">

    <h1>Welcome to M-Aadhaar page</h1>

    <br><br><br>

    <div class="row">

      <div class="col">

        <img src="assets/images/aadhar.png" class="img-fluid" alt="...">

      </div>

      <div class="col">

        <div class="d-grid gap-2 col-8 mx-auto">

            <button type="button" class="btn btn-primary  btn-lg rounded custom"  data-toggle="modal" data-target="#exampleModal">Update Aadhaar</button><br><br>

            <button type="button" class="btn btn-secondary btn-lg rounded custom" data-toggle="modal" data-target="#exampleModal2">Duplicate Aadhaar</button><br><br>

            <button type="button" class="btn btn-success btn-lg rounded custom" data-toggle="modal" data-target="#exampleModal">New Aadhaar</button><br><br>

            <button type="button" class="btn btn-danger btn-lg rounded custom" data-toggle="modal" data-target="#exampleModal3">Delete Aadhaar</button><br><br>

        </div>

      </div>

    </div>

  </div>

  <!-- Modal for creation -->

  <div class="modal fade" id="exampleModal" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">

    <div class="modal-dialog" role="document">

      <div class="modal-content">

        <div class="modal-header">

          <h5 class="modal-title" id="exampleModalLabel">Modal title</h5>

          <button type="button" class="close" data-dismiss="modal" aria-label="Close">

            <span aria-hidden="true">&times;</span>

          </button>

        </div>

        <div style="width: 500px; margin: auto;" class="modal-body">

            <div class="submit-form">

              <div \*ngIf="!isBookAdded">

                <div class="form-group">

                  <label for="name">Name</label>

                  <input class="form-control" id="name" required [(ngModel)]="user.name" name="email"/>

                </div>

                <div class="form-group">

                    <label for="dob">Date of birth</label>

                    <input class="form-control" id="dob" required [(ngModel)]="user.dob" name="dob"/>

                  </div>

                  <div class="form-group">

                    <label for="address">Address</label>

                    <input class="form-control" id="address" required [(ngModel)]="user.address" name="address" />

                  </div>

                <div class="form-group">

                  <label for="email">Email</label>

                  <input class="form-control" id="email" required [(ngModel)]="user.email" name="email" />

                </div>

                <div class="form-group">

                    <label for="mobile">Mobile</label>

                    <input class="form-control" id="mobile" required [(ngModel)]="user.mobile" name="mobile"/>

                  </div>

                <div class="form-group">

                    <label for="gender">Gender</label>

                    <input class="form-control" id="gender" required [(ngModel)]="user.gender" name="gender" />

                  </div>

                <button (click)="addUser()" class="btn btn-success">Create</button>

              </div>

              <div \*ngIf="isBookAdded">

                <h3>Data added SUccessFully</h3>

              </div>

            </div>

          </div>

      </div>

    </div>

  </div>

  <!-- Modal for duplicate -->

  <div class="modal fade" id="exampleModal2" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">

    <div class="modal-dialog" role="document">

      <div class="modal-content">

        <div class="modal-header">

          <h5 class="modal-title" id="exampleModalLabel">Modal title</h5>

          <button type="button" class="close" data-dismiss="modal" aria-label="Close">

            <span aria-hidden="true">&times;</span>

          </button>

        </div>

        <div class="modal-body">

          <h4 style="color:green ;">Your Aadhhar card has been successfully duplicated. It will be delivered to a registered address in 30 working days.

            </h4><br>

            <h4 style="color:rgb(126, 126, 246);">Thank you for using m-Aadhaar services!</h4>

        </div>

        <div class="modal-footer">

          <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>

        </div>

      </div>

    </div>

  </div>

  <!-- Modal for delete -->

  <div class="modal fade" id="exampleModal3" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">

    <div class="modal-dialog" role="document">

      <div class="modal-content">

        <div class="modal-header">

          <h5 class="modal-title" id="exampleModalLabel">Delete Aadhaar Card</h5>

          <button type="button" class="close" data-dismiss="modal" aria-label="Close">

            <span aria-hidden="true">&times;</span>

          </button>

        </div>

        <div class="modal-body">

          <h4 style="color:green ;">Your request was placed successfully</h4><br>

          <h4 style="color: red;">The Aadhaar card gets deleted once the admin gets noticed</h4>

        </div>

        <div class="modal-footer">

          <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>

        </div>

      </div>

    </div>

  </div>

**Userdashboard.component.spec.ts:**

import { ComponentFixture, TestBed } from '@angular/core/testing';

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

describe('UserdashboardComponent', () => {

  let component: UserdashboardComponent;

  let fixture: ComponentFixture<UserdashboardComponent>;

  beforeEach(async () => {

    await TestBed.configureTestingModule({

      declarations: [ UserdashboardComponent ]

    })

    .compileComponents();

    fixture = TestBed.createComponent(UserdashboardComponent);

    component = fixture.componentInstance;

    fixture.detectChanges();

  });

  it('should create', () => {

    expect(component).toBeTruthy();

  });

});

**Userdashboard.components.ts:**

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

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

import { ProductService } from '../product.service';

@Component({

  selector: 'app-userdashboard',

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

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

})

export class UserdashboardComponent implements OnInit {

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

  ngOnInit(): void {

  }

  user = {

    name: '',

    dob:'',

    address:'',

    email: '',

    mobile:'',

    gender:'',

  };

  isBookAdded = false;

  // Add New

  addUser(): void {

    const data = {

      name: this.user.name,

      dob: this.user.dob,

      address: this.user.address,

      email: this.user.email,

      mobile: this.user.mobile,

      gender:this.user.gender,

    };

    if (!data.name) {

      alert('Please add title!');

      return;

    }

    this.service.create(data)

      .subscribe(

        response => {

          console.log(response);

          this.isBookAdded = true;

        },

        error => {

          console.log(error);

        });

  }

}

**App-routing.module.ts:**

import { EditusersComponent } from './editusers/editusers.component';

import { AllusersComponent } from './allusers/allusers.component';

import { HomeComponent } from './home/home.component';

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

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

  {path:"userHome",component:UserdashboardComponent},

  {path:"signUp",component:SignupComponent},

  {path:"home",component:HomeComponent},

  {path:"allusers",component:AllusersComponent},

  {path:"approve",component:EditusersComponent},

  {path:"",redirectTo:"home",pathMatch:"full"}

];

@NgModule({

  imports: [RouterModule.forRoot(routes)],

  exports: [RouterModule]

})

export class AppRoutingModule { }

**App-component.html:**

  <app-header></app-header>

  <router-outlet></router-outlet>

  <app-footer></app-footer>

**App-component.spec.ts:**

import { TestBed } from '@angular/core/testing';

import { RouterTestingModule } from '@angular/router/testing';

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

describe('AppComponent', () => {

  beforeEach(async () => {

    await TestBed.configureTestingModule({

      imports: [

        RouterTestingModule

      ],

      declarations: [

        AppComponent

      ],

    }).compileComponents();

  });

  it('should create the app', () => {

    const fixture = TestBed.createComponent(AppComponent);

    const app = fixture.componentInstance;

    expect(app).toBeTruthy();

  });

  it(`should have as title 'frontend-app'`, () => {

    const fixture = TestBed.createComponent(AppComponent);

    const app = fixture.componentInstance;

    expect(app.title).toEqual('frontend-app');

  });

  it('should render title', () => {

    const fixture = TestBed.createComponent(AppComponent);

    fixture.detectChanges();

    const compiled = fixture.nativeElement as HTMLElement;

    expect(compiled.querySelector('.content span')?.textContent).toContain('frontend-app app is running!');

  });

});

**App-component.ts:**

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

@Component({

  selector: 'app-root',

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

  styleUrls: ['./app.component.css']

})

export class AppComponent {

  title = 'frontend-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 { 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 { HeaderComponent } from './header/header.component';

import { FooterComponent } from './footer/footer.component';

import { HomeComponent } from './home/home.component';

import { AllusersComponent } from './allusers/allusers.component';

import { EditusersComponent } from './editusers/editusers.component';

@NgModule({

  declarations: [

    AppComponent,

    LoginComponent,

    SignupComponent,

    UserdashboardComponent,

    AdmindashboardComponent,

    HeaderComponent,

    FooterComponent,

    HomeComponent,

    AllusersComponent,

    EditusersComponent

  ],

  imports: [

    BrowserModule,

    AppRoutingModule,HttpClientModule,FormsModule,ReactiveFormsModule

  ],

  providers: [],

  bootstrap: [AppComponent]

})

export class AppModule { }

**Login.service.spec.ts:**

import { TestBed } from '@angular/core/testing';

import { LoginService } from './login.service';

describe('LoginService', () => {

  let service: LoginService;

  beforeEach(() => {

    TestBed.configureTestingModule({});

    service = TestBed.inject(LoginService);

  });

  it('should be created', () => {

    expect(service).toBeTruthy();

  });

});

**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"});

  }

}

**Login.spec.ts:**

import { Login } from './login';

describe('Login', () => {

  it('should create an instance', () => {

    expect(new Login()).toBeTruthy();

  });

});

**Login.ts:**

// map to entity class or json data.

export class Login {

    constructor(public emailid:string,

        public password:string,

        public typeOfUser:string){}

}

**Product.service.spec.ts:**

import { TestBed } from '@angular/core/testing';

import { ProductService } from './product.service';

describe('ProductService', () => {

  let service: ProductService;

  beforeEach(() => {

    TestBed.configureTestingModule({});

    service = TestBed.inject(ProductService);

  });

  it('should be created', () => {

    expect(service).toBeTruthy();

  });

});

**Product.service.ts:**

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

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

import { catchError, Observable, throwError } from 'rxjs';

import { UserClass } from './UserClass';

@Injectable({

  providedIn: 'root'

})

export class ProductService {

  url:string="http://localhost:9090/api/user/";

  headers = new HttpHeaders().set('Content-Type', 'application/json');

  constructor(public http:HttpClient) { }

  //get all users

  getAllUser():Observable<UserClass[]>{

    return this.http.get<UserClass[]>(this.url);

}

//delete user by id

deletePost(id:number) {

  let endPoints = id;

  this.http.delete(this.url + endPoints).subscribe(data => {

     return this.getAllUser();

  });

}

//post

create(data: any): Observable<any> {

  return this.http.post(this.url, data).pipe(

    catchError(this.handleError)

    );

}

 // Handle API errors

handleError(error: HttpErrorResponse) {

  if (error.error instanceof ErrorEvent) {

    console.error('An error occurred:', error.error.message);

  } else {

    console.error(

      `Backend returned code ${error.status}, ` +

      `body was: ${error.error}`);

  }

  return throwError(

    'Something bad happened; please try again later.');

};

}

**UserClass.ts:**

export class UserClass{

    id: number;

    name: string;

    dob:string;

    address:string;

    email: string;

    mobile:string;

    gender:string;

}