Source code:

Bank end :

Admin entity:

```
package com.simplilearn.demo.entity;


import javax.persistence.Column;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;


public class Admin {


	@Id

	@Column(name="id")

	@GeneratedValue(strategy = GenerationType.IDENTITY)

	private int id;

	private String email;

	private String passwod;

	public int getId() {

		return id;

	}

	public void setId(int id) {

		this.id = id;

	}

	public String getEmail() {

		return email;

	}
```

```
    public void setEmail(String email) {
        this.email = email;
    }
    public String getPasswod() {
        return passwod;
    }
    public void setPasswod(String passwod) {
        this.passwod = passwod;
    }

}
```

User entity:

```
package com.simplilearn.demo.entity;



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 name;
private String address;
private String email;
private String gender;

public int getId() {
        return id;
}
public void setId(int id) {
        this.id = id;
}
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 getGender() {
		return gender;
	}
	public void setGender(String gender) {
		this.gender = gender;
	}
	@Override
	public String toString() {
		return "User [id=" + id + ", name=" + name + ", address=" + address + ", email=" + email + ", gender=" + gender
				+ "]";
	}


	public User() {
		super();
		// TODO Auto-generated constructor stub
	}

}
```

Admin repo:

```
package com.simplilearn.demo.repo;


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

import org.springframework.stereotype.Repository;


import com.simplilearn.demo.entity.Admin;


@Repository

public interface AdminRepo extends JpaRepository<Admin,Integer> {


}
```

User repo:

```
package com.simplilearn.demo.repo;


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

import org.springframework.data.jpa.repository.config.JpaRepositoryConfigExtension;

import org.springframework.stereotype.Repository;

import com.simplilearn.demo.entity.User;


@Repository

public interface UserRepo extends JpaRepository<User,Integer> {


}
```

Admin service:

```
package com.simplilearn.demo.service;
```

```
import java.util.List;

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

import com.simplilearn.demo.entity.Admin;

import com.simplilearn.demo.repo.AdminRepo;

@Service
public class AdminService {

    @Autowired
    private AdminRepo repo;

    public Admin addAdmin(Admin u) {
        return repo.save(u);
    }

    public List<Admin> getAllAdmin(){
        return repo.findAll();
    }

    public Admin getAdminById(int id) {
        if(repo.findById(id).isPresent()) {
            return repo.findById(id).get();
        }
        else {
            return null;
        }

    }
    public boolean deleteAdmin(int id) {
        if(repo.findById(id).isPresent()) {
            repo.deleteById(id);
            return true;
        }

        else {
            return false;
        }
    }


}
```

User service:

```
package com.simplilearn.demo.service;



import java.util.List;


import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import com.simplilearn.demo.entity.User;
import com.simplilearn.demo.repo.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.setAddress(newUser.getAddress());
            olduser.setEmail(newUser.getEmail());
            //olduser.setMobileno(newUser.getMobileno());
            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;
        }
    }

}
```

Admin controller:

```
package com.simplilearn.demo;


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.RequestBody;

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

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


import com.simplilearn.demo.entity.Admin;

import com.simplilearn.demo.service.AdminService;




@CrossOrigin("http://localhost:4200")

@RequestMapping("/api/admin")

public class AdminController {


	@Autowired

	private AdminService service;
```

```
    @PostMapping("/")
    public ResponseEntity<Admin> addadmin(@RequestBody Admin u){

        Admin admin= service.addAdmin(u);

        if(admin!=null)
            return new ResponseEntity<Admin>(admin,HttpStatus.CREATED);
        else
            return new
ResponseEntity<Admin>(admin,HttpStatus.INTERNAL_SERVER_ERROR);

    }
    @GetMapping("/")
    public List<Admin> getAllAdmin(){
        return service.getAllAdmin();
    }

    @DeleteMapping("/{id}")
    public ResponseEntity<String>deleteAdmin(@PathVariable int id){
        boolean result = service.deleteAdmin(id);
        if(result)
            return new ResponseEntity<String>("Object Deleted",HttpStatus.OK);
        else
            return new ResponseEntity<String>("NO user Found",
HttpStatus.NOT_FOUND);

    }
```

```
}
```

User controller:

```
package com.simplilearn.demo;



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.simplilearn.demo.entity.User;
import com.simplilearn.demo.service.UserService;
@CrossOrigin("http://localhost:4200/")
@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);

    }

}
```

My aadar application:

```
package com.simplilearn.demo;


import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;


@SpringBootApplication

public class MAadharApplication {


	public static void main(String[] args) {

		SpringApplication.run(MAadharApplication.class, args);

	}


}
```

Application:

```
spring.datasource.username=root

spring.datasource.password=root

spring.datasource.url=jdbc:mysql://localhost:3307/capstonedemo

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect

spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true

server.port=8082
```

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.simplilearn.demo</groupId>
    <artifactId>MAadharApplication</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>MAadharApplication</name>
    <description>Capstone Project MAadhar</description>
    <properties>
        <java.version>17</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>mysql</groupId>
                <artifactId>mysql-connector-java</artifactId>
                <scope>runtime</scope>
            </dependency>



            <!-- https://mvnrepository.com/artifact/org.projectlombok/lombok-maven-plugin --
>
<dependency>
  <groupId>org.projectlombok</groupId>
  <artifactId>lombok-maven-plugin</artifactId>
  <version>1.18.20.0</version>
  <scope>provided</scope>
</dependency>

<!--Cross origin Dependency-->
    <dependency>
      <groupId>com.allenru</groupId>
      <artifactId>cross-origin</artifactId>
      <version>1.0.1</version>
    </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>
```

Front end:

Admin dashboard.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.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;
    }
```

```
 }



}
```

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.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);
    });
 }


}
```

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.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.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.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.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="#"> Varniraj Service PVT. LTD </a>  </p>
            </div>
        </footer>
```

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>
```

Home.component.html

```
<div class="container">


<div>

    <img class="image" style="margin-top: 50px;" src="https://www.91-cdn.com/hub/wp-
content/uploads/2018/01/Aadhaar-logo-Featured.jpg" alt="">

</div>

<h1 style="color: black;top: 52%; left:37%; position: absolute;">Aadhaar Application</h1>

</div>
```

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.html:

```
<h2 style="text-align: center; margin-top: 30px;">Aadhaar Login</h2>

<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.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,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.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")
    })
  }
}
```

App-rounting.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.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 { }
```

app testing:

```
package com.springbot.practice;

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 MAadhaarTestscode {



            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("guru");

                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", "F:\\Java FSD
Software\\chrome\\chromedriver.exe");
```

```
                 driver = new ChromeDriver();
        }


        @AfterMethod
        public void afterMethod() {
                //driver.close();
                 driver = null;
        }
}
```