

mAadhar Application Source Code:

Frontend:

admin.component.css:

```
.image{
    background-size:cover; width:auto;
    background-image:url('https://thumbs.dreamstime.com/b/colorful-bead-black-letter-word-admin-
abbreviation-administration-wood-background-187718269.jpg');
    height: 800px;
    background-position:89% 30%;
    background-size: 500px;
    background-repeat: no-repeat;
    background-color:antiquewhite;
}

input.ng-invalid.ng-touched {
    border: 1px solid red;
}
```

admin.component.html:

```
<div class="image">
  <br />
  <br />
  <div class="container d-flex justify-content-left">
    <form
      [formGroup]="loginForm"
      (ngSubmit)="onSubmit()"
      class="w-50 p-4"
      style="color: rgb(3, 11, 51)"
    >
      <h4 class="text-center">Admin Login</h4>

      <div class="row">
        <div class="form-group">
          <label for="username">Admin name</label>
          <input
            type="text"
            id="admin"
            class="form-control"
            formControlName="admin"
          />
          <span
            style="color: red"
            class="small"
            *ngIf="
              !loginForm.get('admin').valid && loginForm.get('admin').touched
            ">Please enter a valid admin name</span>
        </div>
      </div>
    </div>
  </div>
```

```

<div class="row">
  <div class="form-group">
    <label for="password">Password</label>
    <input
      type="password"
      id="password"
      class="form-control"
      formControlName="password"
    />
    <span
      style="color: red"
      class="small"
      *ngIf="
        !loginForm.get('password').valid &&
        loginForm.get('password').touched
      "
    >Please enter a valid password</span>
  >
<br />
<span
  style="color: red"
  class="small"
  *ngIf="loginForm.get('password').hasError('passwordStrength')"
>
  {{ loginForm.get("password").errors["passwordStrength"] }}
</span>
</div>
</div>
<br />
<button type="submit" class="btn btn-outline-danger">Login</button>
</form>
</div>
</div>

```

admin.component.ts:

```

import { HttpClient } from '@angular/common/http';
import { Component, OnInit } from '@angular/core';
import { FormBuilder, FormGroup, Validators } from '@angular/forms';
import { Router } from '@angular/router';
import { PasswordStrengthValidator } from "../password-strength.validators"

@Component({
  selector: 'app-admin',
  templateUrl: './admin.component.html',
  styleUrls: ['./admin.component.css']
})
export class AdminComponent implements OnInit {
  public loginForm!: FormGroup;
  url="http://localhost:8080/admin";
  submitted=false;

  constructor(private formBuilder: FormBuilder,private http: HttpClient , private router:Router) { }

  ngOnInit(): void {
    this.loginForm=this.formBuilder.group({
      admin : ['',Validators.required],
      password : ['',Validators.compose([

```

```

        Validators.required, PasswordStrengthValidator]])

    })
  }
  onSubmit() {
    this.submitted = true;
    this.login();
  }

  login(){
    this.http.get<any>(this.url).subscribe(res=>{
      const log = res.find((a:any)=>{
        return a.admin === this.loginForm.value.admin && a.password === this.loginForm.value.password
      });
      if(log){
        alert("Login Success,Click Ok to continue!!");
        this.loginForm.reset();
        this.router.navigate(['adminhome']);
      }else{
        alert("Try Again!!");
      }
    },err=>{
      alert("Something Went Wrong!!");
      this.loginForm.reset();
    })
  }

}

```

password-strength.validators.ts:

```

import { AbstractControl, ValidationErrors } from "@angular/forms"

export const PasswordStrengthValidator = function (control: AbstractControl): ValidationErrors | null {

  let value: string = control.value || '';

  if (!value) {
    return null
  }

  let upperCaseCharacters = /[A-Z]+/g
  if (upperCaseCharacters.test(value) === false) {
    return { passwordStrength: `Upper case required` };
  }

  let lowerCaseCharacters = /[a-z]+/g
  if (lowerCaseCharacters.test(value) === false) {
    return { passwordStrength: `lower case required` };
  }

  let numberCharacters = /[0-9]+/g
  if (numberCharacters.test(value) === false) {

```

```

    return { passwordStrength: `number required` };
  }

  let specialCharacters = /[!@#$$%^&*()_+\\-=\\[\\]\\{\\};':"\\|,.<>\\/?]+/
  if (specialCharacters.test(value) === false) {
    return { passwordStrength: `Special char required` };
  }
  return null;
}

```

adminhome.component.css:

```

.image{
  height:39em; background-size:cover; width:auto;
  background-image:url('aadhar.jpg');
  background-position:50% 50%;
}

```

adminhome.component.html:

```

<div class="image">
  <div>
    <h1 class="d-flex justify-content-center" style="color: crimson">
      AADHAR APPLICATION
    </h1>
  </div>
  <div class="container">
    <button class="btn btn-outline-danger mt-5" (click)="onFetchUsers()">
      View All
    </button>
    <div class="row m-3">
      <div class="col">
        <p *ngIf="fetchedUsers.length < 1">No Users Available !!</p>
        <ul class="list-group" *ngIf="fetchedUsers.length >= 1">
          <li class="list-group-item" *ngFor="let user of fetchedUsers">
            <p><b>Aadhar id:</b>&nbsp;{{ user.citizenId }}</p>
            <p><b>Name:</b>&nbsp;{{ user.name }}</p>
            <p><b>Date Of Birth:</b>&nbsp;{{ user.dob }}</p>
            <p><b>Email:</b>&nbsp;{{ user.emailid }}</p>

            <p><b>Address:</b>&nbsp;{{ user.address }}</p>
            <p><b>Mobile No:</b>&nbsp;{{ user.mobilenno }}</p>

            <p><b>Gender:</b>&nbsp;{{ user.gender }}</p>

            <button
              class="btn btn-sm btn-danger"
              (click)="onDeleteUser(user.citizenId)"
            >
              Delete
            </button>
          </li>
        </ul>
      </div>
    </div>
  </div>
</div>

```

</div>

adminhome.component.ts:

```
import { HttpClient } from '@angular/common/http';
import { Component, OnInit } from '@angular/core';
import { FormBuilder } from '@angular/forms';
import { map } from 'rxjs/operators';
import { ApiService } from '../service/api.service';
import { UserData } from '../userdata.model';

@Component({
  selector: 'app-adminhome',
  templateUrl: './adminhome.component.html',
  styleUrls: ['./adminhome.component.css']
})
export class AdminhomeComponent implements OnInit {

  fetchedUsers: UserData[] = [];
  backendurl = 'http://localhost:8080/citizens';
  constructor(private http:HttpClient,private formBuilder:FormBuilder,private api:ApiService) {}

  ngOnInit(): void {
  }

  onFetchUsers(){
    this.fetchUsers();
  }
  fetchUsers() {
    this.http
      .get(this.backendurl)
      .pipe(
        map((responseData) => {
          const userArray: UserData[] = [];
          for (const key in responseData) {
            var x = { ...responseData[key] };
            userArray.push(x);
          }
          return userArray;
        })
      )
      .subscribe((users) => {
        this.fetchedUsers = users;
      });
  }

  onDeleteUser(id: number) {
    this.http.delete(this.backendurl + '/' + id).subscribe((response) => {
      console.log('User deleted: ' + response);
      // this.fetchPosts();
    });
  }
}
```

userdata.model.ts:

```
}
home.component.css:
```

home.component.html:

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 {
    }
}

```

register.component.css:

```

.image{
    height:50em; background-size:cover; width:auto;
    background-image:url('https://img.etimg.com/thumb/width-1200,height-900,imgsize-341034,resizemode-75,msid-89920834/wealth/save/baal-aadhaar-how-to-apply-aadhaar-card-for-children.jpg');
    background-position:89% 30%;
    background-size: 550px;
    background-repeat: no-repeat;
    background-color:antiquewhite
}

```

register.component.html:

```

<div class="image">
  <br />
  <h3 style="color: rgba(42, 2, 71, 0.84)" class="text-center">
    Registration Page
  </h3>
  <ul class="nav " >
    <li class="nav-item" >
      <a routerLink="/home" class="nav-link" >Home</a>
    </li>
  </ul>
  <div class="container mt-3 d-flex justify-content-left">
    <form #f="ngForm" class="w-50 p-4" style="color: rgb(120, 55, 5)">
      <div class="row mb-2">
        <div class="form-group">
          <label for="name">Name</label>
          <input
            type="text"
            id="name"
            class="form-control"
            name="name"
            required
            ngModel
            #name="ngModel"
          />
        </div>
      </div>
      <div class="row mb-2">
        <div class="form-group">
          <label for="dob">Date Of Birth</label>
          <input
            type="text"
            id="dob"
            class="form-control"
            name="dob"
            ngModel
            required

```

```

        dob
        #dob="ngModel"
    />
    <span
        id="err-dob"
        style="color: red"
        class="small"
        *ngIf="!dob.valid && dob.touched"
    >Please enter a valid dob</span>
    </div>
</div>
<div class="row mb-2">
    <div class="form-group">
        <label for="emailid">Email</label>
        <input
            type="text"
            id="emailid"
            class="form-control"
            name="emailid"
            ngModel
            required
            emailid
            #emailid="ngModel"
        />
        <span
            id="err-email"
            style="color: red"
            class="small"
            *ngIf="!emailid.valid && emailid.touched"
        >Please enter a valid email</span>
    </div>
</div>
<div class="row mb-2">
    <div class="form-group">
        <label for="address">Address</label>
        <textarea
            type="text"
            id="address"
            class="form-control"
            name="address"
            ngModel
            required
            address
            #address="ngModel"
        ></textarea>
        <span
            id="err-address"
            style="color: red"
            class="small"
            *ngIf="!address.valid && address.touched"
        >Please enter a valid address</span>
    </div>
</div>
<div class="row mb-2">
    <div class="form-group">

```



```

<label for="Phoneno">Mobile No:</label>
<input
  type="text"
  id="mobilenno"
  class="form-control"
  name="mobilenno"
  ngModel
  required
  mobilenno
  #mobilenno="ngModel"
/>
<span
  id="err-mobile"
  style="color: red"
  class="small"
  *ngIf="!mobilenno.valid && mobilenno.touched"
  >Please enter a valid Mobile Number</span>
>
</div>
</div>
<div class="row mb-2">
  <div class="form-group">
    <label for="gender">Gender</label>
    <input
      type="text"
      id="gender"
      class="form-control"
      name="gender"
      ngModel
      required
      gender
      #gender="ngModel"
    />
    <span
      id="err-gender"
      style="color: red"
      class="small"
      *ngIf="!gender.valid && gender.touched"
      >Please enter a valid Gender</span>
    >
  </div>
</div>
<div>
  <button
    class="btn btn-outline-danger m-2"
    [disabled]="!f.valid"
    (click)="onAddCitizen(f.value, f)"
    id="register"
  >
    Register
  </button>
</form>
</div>
</div>

```

register.component.ts:

```
import { HttpClient } from '@angular/common/http';
```

```

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

@Component({
  selector: 'app-register',
  templateUrl: './register.component.html',
  styleUrls: ['./register.component.css']
})
export class RegisterComponent implements OnInit {
  url='http://localhost:8080/citizens';
  constructor(private http:HttpClient) { }

  ngOnInit(): void {
  }
  onAddCitizen(citizenData:
{name:string,dob:string,emailid:string,gender:string,mobilenumber:string,address:string }, form: NgForm) {
    this.http.post(this.url, citizenData).subscribe((responseData) => {
      console.log(responseData);
      alert("Registered Successfully!!");
      form.reset();
    },err=>{
      alert("Something Happened!!")
    });
  }
}

```

api.service.ts:

```

import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { map } from 'rxjs/operators';

@Injectable({
  providedIn: 'root'
})
export class ApiService {

  constructor(private http:HttpClient) { }

  updateUser(data:any,id:number){
    return this.http.put<any>('http://localhost:8080/citizens',data)
    .pipe(map((res:any)=>{
      return res;
    })))
  }
}

```

user.component.css:

```

.image{
  height:39em; background-size:cover; width:auto;
  background-image:url('https://encrypted-
tbn0.gstatic.com/images?q=tbn:AND9GcQoe56UV1WpPue4uvKNEGweCaLMfIppfpYPA&usqp=CAU');
}

```

```

background-position:89% 30%;
background-size: 600px;
background-repeat: no-repeat;
background-color:antiquewhite;
}

```

user.component.html:

```

<div class="image">
  <br />
  <br />
  <div class="container d-flex justify-content-left">
    <form
      [formGroup]="loginForm"
      (ngSubmit)="onSubmit()"
      class="w-50 p-4"
      style="color: rgb(3, 11, 51)"
    >
      <h4 class="text-center">Citizen Login</h4>

      <div class="row">
        <div class="form-group">
          <label for="citizenId">AadharId</label>
          <br />
          <input
            type="number"
            id="citizen_id"
            class="form-control"
            formControlName="citizenId"
          />
        </div>
      </div>
    </div>
    <div class="row">
      <div class="form-group">
        <label for="password">Password</label>
        <br />
        <span class="small" style="color: darkgreen">
          >Your mobile number is your password</span>
      </div>

      <input
        type="password"
        id="password"
        class="form-control"
        formControlName="mobilenos"
      />
      <span
        style="color: red"
        class="small"
        *ngIf="
          !loginForm.get('mobilenos').valid &&
          loginForm.get('mobilenos').touched
        "
      >Please enter a valid password</span>
    </div>
  </div>
  <br />

```

```

        <button type="submit" class="btn btn-outline-danger">Login</button>
    </form>
</div>
</div>

```

User.component.ts:

```

import { HttpClient } from '@angular/common/http';
import { Component, OnInit } from '@angular/core';
import { FormBuilder, FormGroup, Validators } from '@angular/forms';
import { Router } from '@angular/router';

@Component({
  selector: 'app-user',
  templateUrl: './user.component.html',
  styleUrls: ['./user.component.css']
})
export class UserComponent implements OnInit {

  public loginForm!: FormGroup;
  url="http://localhost:8080/citizens";
  submitted=false;

  constructor(private formBuilder: FormBuilder,private http: HttpClient , private router:Router) { }

  ngOnInit(): void {
    this.loginForm=this.formBuilder.group({
      citizenId : ['',Validators.required],
      mobileno : ['',Validators.required]
    })
  }
  onSubmit() {
    this.submitted = true;
    this.login();
  }

  login(){
    this.http.get<any>(this.url).subscribe(res=>{
      const log = res.find((a:any)=>{
        return a.citizenId === this.loginForm.value.citizenId && a.mobileno ===
this.loginForm.value.mobileno
      });
      if(log){
        alert("Login Success,Click Ok to continue!!");
        this.loginForm.reset();
        this.router.navigate(['userhome']);
      }else{
        alert("Try Again!!");
      }
    },err=>{
      alert("Something Went Wrong!!");
      this.loginForm.reset();
    })
  }

}

```

```
}
```

user.model.ts:

```
export class UserModel{
  citizenId:number;
  name:string;
  dob:string;
  mobileno:string;
  address:string;
  emailid:string;
  gender:string;
}
```

userdata.model.ts:

```
export interface UserData{
  citizenId:number,
  name:string,
  dob:string,
  mobileno:string,
  address:string,
  emailid:string,
  gender:string
}
```

userhome.component.css:

```
.image{
  height:39em; background-size:cover; width:auto;
  background-image:url('aadhar.jpg');
  background-position:50% 50%;
}
```

userhome.component.html:

```
<div class="image">
  <div class="container">
    <div class="h1 d-flex justify-content-center">AADHAR APPLICATION</div>
    <div class="col card d-flex justify-content-center" style="width: 25rem">
      <div class="card-body">
        <h5 class="card-title" style="color: red">Your Details</h5>

        <div class="card-text" *ngFor="let user of fetchedUsers">
          <p><b>Aadhar id:</b>&nbsp;&nbsp;&nbsp;{{ user.citizenId }}</p>
          <p><b>Name&nbsp;&nbsp;&nbsp;:</b>&nbsp;&nbsp;&nbsp;{{ user.name }}</p>
          <p><b>Date Of Birth:</b>&nbsp;&nbsp;&nbsp;{{ user.dob }}</p>
```

[illegible]

```
        required
        FormControlName="name"
    />
</div>
</div>
<div class="row mb-2">
    <div class="form-group">
        <label for="dob">Date Of Birth</label>
        <input
            type="text"
            id="dob"
            class="form-control"
            name="dob"
            FormControlName="dob"
            required
        />
    </div>
</div>
<div class="row mb-2">
    <div class="form-group">
        <label for="emailid">Email</label>
        <input
            type="text"
            id="emailid"
            class="form-control"
            name="emailid"
            FormControlName="emailid"
            required
        />
    </div>
</div>
<div class="row mb-2">
    <div class="form-group">
        <label for="address">Address</label>
        <textarea
            type="text"
            id="address"
            class="form-control"
            name="address"
            required
            FormControlName="address"
        ></textarea>
    </div>
</div>
<div class="row mb-2">
    <div class="form-group">
        <label for="Phoneno">Mobile No:</label>
        <input
            type="text"
            id="mobilenno"
            class="form-control"
            name="mobilenno"
            required
            FormControlName="mobilenno"
        />
    </div>
</div>
<div class="row mb-2">
```

```

        <div class="form-group">
            <label for="gender">Gender</label>
            <input
                type="text"
                id="gender"
                class="form-control"
                name="gender"
                required
                formControlName="gender"
            />
        </div>
    </div>
</form>
<div class="modal-footer">
    <button
        type="button"
        class="btn btn-secondary"
        data-bs-dismiss="modal"
    >
        Close
    </button>
    <button type="button" (click)="updateUser()" class="btn btn-primary">
        Save changes
    </button>
</div>
</div>
</div>
</div>
</div>

```

userhome.component.ts:

```

import { HttpClient } from '@angular/common/http';
import { Component, OnInit } from '@angular/core';
import { FormBuilder, FormGroup } from '@angular/forms';
import { map } from 'rxjs/operators';
import { ApiService } from '../service/api.service';
import { UserModel } from '../user.model';
import { UserData } from '../userdata.model';

@Component({
  selector: 'app-userhome',
  templateUrl: './userhome.component.html',
  styleUrls: ['./userhome.component.css']
})
export class UserhomeComponent implements OnInit {
  userModelObj: UserModel = new UserModel();
  public loginForm!: FormGroup;
  fetchedUsers: UserData[] = [];
  backendurl = 'http://localhost:8080/citizens';
  constructor(private http: HttpClient, private formBuilder: FormBuilder, private api: ApiService) {}

  ngOnInit(): void {
    this.loginForm = this.formBuilder.group({
      name: [''],
      dob: [''],
      emailid: [''],
      mobileno: [''],

```



```

        address : [''],
        gender : [''],
    })
    this.fetchUsers();
}

fetchUsers() {
    this.http
        .get(this.backendurl)
        .pipe(
            map((responseData) => {
                const userArray: UserData[] = [];
                for (const key in responseData) {
                    var x = { ...responseData[key] };
                    userArray.push(x);
                }
                return userArray;
            })
        )
        .subscribe((users) => {
            this.fetchedUsers = users;
        });
}

onEdit(user:any){
    this.userModelObj.citizenId=user.citizenId;
    this.loginForm.controls['name'].setValue(user.name);
    this.loginForm.controls['dob'].setValue(user.dob);
    this.loginForm.controls['emailid'].setValue(user.emailid);
    this.loginForm.controls['mobilenos'].setValue(user.mobilenos);
    this.loginForm.controls['address'].setValue(user.address);
    this.loginForm.controls['gender'].setValue(user.gender);
}

updateUser(){
    this.userModelObj.name=this.loginForm.value.name;
    this.userModelObj.emailid=this.loginForm.value.emailid;
    this.userModelObj.gender=this.loginForm.value.gender;

    this.userModelObj.address=this.loginForm.value.address;
    this.userModelObj.mobilenos=this.loginForm.value.mobilenos;
    this.userModelObj.dob=this.loginForm.value.dob;

    this.api.UpdateUser(this.userModelObj,this.userModelObj.citizenId)
    .subscribe(res=>{
        alert("Updated Successfully!!");
        this.loginForm.reset();
        this.fetchUsers();
    },err=>{
        alert("Something Went Wrong:/")
    })
}
}

```

app.component.css:

```

.image{
    height:39em; background-size:cover; width:auto;

```

```

    background-image:url('aadhar.jpg');
    background-position:50% 50%;
}

```

app.component.html:

```
<router-outlet></router-outlet>
```

app.component.spec.ts:

```

import { TestBed } from '@angular/core/testing';
import { AppComponent } from './app.component';

describe('AppComponent', () => {
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      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 'AadharApp'`, () => {
    const fixture = TestBed.createComponent(AppComponent);
    const app = fixture.componentInstance;
    expect(app.title).toEqual('AadharApp');
  });

  it('should render title', () => {
    const fixture = TestBed.createComponent(AppComponent);
    fixture.detectChanges();
    const compiled = fixture.nativeElement as HTMLElement;
    expect(compiled.querySelector('.content span')?.textContent).toContain('AadharApp 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 = 'AadharApp';
}

```

app.module.ts:

```
import { NgModule } from '@angular/core';
import { FormsModule, ReactiveFormsModule } from '@angular/forms';
import { BrowserModule } from '@angular/platform-browser';
import { RouterModule, Routes } from '@angular/router';
import { HttpClientModule } from '@angular/common/http'

import { AppComponent } from './app.component';
import { AdminComponent } from './admin/admin.component';
import { HomeComponent } from './home/home.component';
import { RegisterComponent } from './register/register.component';
import { UserComponent } from './user/user.component';
import { AdminhomeComponent } from './adminhome/adminhome.component';
import { UserhomeComponent } from './userhome/userhome.component';

const appRoutes:Routes=[
  { path: '', redirectTo: 'home' , pathMatch:'full' },
  { path: 'adminlogin', component: AdminComponent },
  { path: 'home', component: HomeComponent },
  { path : 'register' , component: RegisterComponent},
  { path : 'userlogin' , component :UserComponent},
  { path : 'adminhome' , component: AdminhomeComponent},
  { path : 'userhome' , component: UserhomeComponent}

]

@NgModule({
  declarations: [
    AppComponent,
    AdminComponent,
    HomeComponent,
    RegisterComponent,
    UserComponent,
    AdminhomeComponent,
    UserhomeComponent
  ],
  imports: [
    BrowserModule,FormsModule,
    RouterModule.forRoot(appRoutes),
    ReactiveFormsModule,
    HttpClientModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
```

```

<title>AadharApp</title>
<base href="/" />
<meta name="viewport" content="width=device-width, initial-scale=1" />
<link rel="icon" type="image/x-icon" href="favicon.ico" />
<style>
  body {
    background-image: url("");
    background-repeat: no-repeat;
    background-attachment: fixed;
    background-size: cover;
  }
</style>
</head>
<body>
  <app-root></app-root>
</body>
</html>

```

dockerfile:

```

FROM node:16.14.0-alpine AS build
RUN npm i -g @angular/cli@13.1.0
WORKDIR /app
COPY package.json package-lock.json ./
RUN npm install
COPY . .
RUN ng build --prod

FROM nginx
COPY --from=build /app/dist/AadharApp/ /usr/share/nginx/html/

```

BackEnd:

aadharbackendapplication.java

```
package com.example.capstone;
```

```
import org.springframework.boot.SpringApplication;
```

```
import org.springframework.boot.autoconfigure.SpringBootApplication;
```

```
@SpringBootApplication
```

```
public class AadharBackendApplication {
```

```
    public static void main(String[] args) {
```

```
        SpringApplication.run(AadharBackendApplication.class, args);
    }

}
```

aadharapiexception.java

```
package com.example.capstone.AadharRestApi.Controller;

import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.ControllerAdvice;
import org.springframework.web.bind.annotation.ExceptionHandler;

import com.examples.capstone.AadharRestApi.Exception.AdminNotFoundException;
import com.examples.capstone.AadharRestApi.Exception.CitizenNotFoundException;
import com.examples.capstone.AadharRestApi.Exception.UserNotFoundException;

@ControllerAdvice
public class AadharApiException {

    @ExceptionHandler(value=AdminNotFoundException.class)
    public ResponseEntity<Object> handleException(AdminNotFoundException ex) {
        return new ResponseEntity<Object>(ex.getMessage(), HttpStatus.NOT_FOUND);
    }

    @ExceptionHandler(value=UserNotFoundException.class)
```

```
public ResponseEntity<Object> handleException(UserNotFoundException ex) {  
    return new ResponseEntity<Object>(ex.getMessage(), HttpStatus.NOT_FOUND);  
}
```

```
@ExceptionHandler(value=CitizenNotFoundException.class)  
public ResponseEntity<Object> handleException(CitizenNotFoundException ex) {  
    return new ResponseEntity<Object>(ex.getMessage(), HttpStatus.NOT_FOUND);  
}
```

```
}
```

restapicontroller.java

```
package com.example.capstone.AadharRestApi.Controller;
```

```
import org.springframework.http.HttpStatus;
```

```
import org.springframework.http.ResponseEntity;
```

```
import org.springframework.web.bind.annotation.ControllerAdvice;
```

```
import org.springframework.web.bind.annotation.ExceptionHandler;
```

```
import com.examples.capstone.AadharRestApi.Exception.AdminNotFoundException;
```

```
import com.examples.capstone.AadharRestApi.Exception.CitizenNotFoundException;
```

```
import com.examples.capstone.AadharRestApi.Exception.UserNotFoundException;
```

```
@ControllerAdvice
```

```
public class AadharApiException {
```

```
@ExceptionHandler(value=AdminNotFoundException.class)

public ResponseEntity<Object> handleException(AdminNotFoundException ex) {

    return new ResponseEntity<Object>(ex.getMessage(), HttpStatus.NOT_FOUND);

}
```

```
@ExceptionHandler(value=UserNotFoundException.class)

public ResponseEntity<Object> handleException(UserNotFoundException ex) {

    return new ResponseEntity<Object>(ex.getMessage(), HttpStatus.NOT_FOUND);

}
```

```
@ExceptionHandler(value=CitizenNotFoundException.class)

public ResponseEntity<Object> handleException(CitizenNotFoundException ex) {

    return new ResponseEntity<Object>(ex.getMessage(), HttpStatus.NOT_FOUND);

}
```

```
}
```

admin.java

```
package com.example.capstone.AadharRestApi.Entity;
```

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

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

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

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

@Entity

public class Admin {

 @Id

 @GeneratedValue(strategy = GenerationType.IDENTITY)

 private int id;

 private String admin;

 private String password;

 public int getId() {

 return id;

 }

 public void setId(int id) {

 this.id = id;

 }

 public String getAdmin() {

 return admin;

 }

 public void setAdmin(String admin) {

 this.admin = admin;

 }

 public String getPassword() {

 return password;

 }

 public void setPassword(String password) {


```

        this.password = password;
    }

    @Override
    public String toString() {
        return "Admin [id=" + id + ", admin=" + admin + ", password=" + password + "]";
    }

}

```

citizens.java

```
package com.example.capstone.AadharRestApi.Entity;
```

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

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

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

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

```
@Entity
```

```
public class Citizens {
```

```
    @Id
```

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

```
    private int citizenId;
```

```
private String name;

private String dob;

private String emailid;

private String mobilenno;

private String gender;

private String address;

public int getCitizenId() {

    return citizenId;

}

public void setCitizenId(int citizenId) {

    this.citizenId = citizenId;

}

public String getName() {

    return name;

}

public void setName(String name) {

    this.name = name;

}

public String getDob() {

    return dob;

}

public void setDob(String dob) {

    this.dob = dob;

}

public String getEmailid() {
```

```
        return emailid;
    }

    public void setEmailid(String emailid) {

        this.emailid = emailid;
    }

    public String getMobilenno() {

        return mobilenno;
    }

    public void setMobilenno(String mobilenno) {

        this.mobilenno = mobilenno;
    }

    public String getGender() {

        return gender;
    }

    public void setGender(String gender) {

        this.gender = gender;
    }

    public String getAddress() {

        return address;
    }

    public void setAddress(String address) {

        this.address = address;
    }

    @Override

    public String toString() {
```

```

        return "Citizens [citizenId=" + citizenId + ", name=" + name + ", dob=" + dob + ",
emailid=" + emailid
        + ",mobilen=" + mobilen + ", gender=" + gender + ", address=" +
address
        + "];
    }

```

```

}

```

user.java

```

package com.example.capstone.AadharRestApi.Entity;

```

```

import javax.persistence.Entity;

```

```

import javax.persistence.GeneratedValue;

```

```

import javax.persistence.GenerationType;

```

```

import javax.persistence.Id;

```

```

@Entity

```

```

public class User {

```

```

    @Id

```

```

    @GeneratedValue(strategy = GenerationType.IDENTITY)

```

```
private int id;

private String emailId;

private String password;

public int getId() {

    return id;

}

public void setId(int id) {

    this.id = id;

}

public String getEmail() {

    return emailId;

}

public void setEmail(String email) {

    this.emailId = email;

}

public String getPassword() {

    return password;

}

public void setPassword(String password) {

    this.password = password;

}

@Override

public String toString() {

    return "User [id=" + id + ", email=" + emailId + ", password=" + password + "];"

}
```

```
}
```

adminrepo.java

```
package com.example.capstone.AadharRestApi.Repository;
```

```
import org.springframework.data.repository.CrudRepository;
```

```
import com.example.capstone.AadharRestApi.Entity.Admin;
```

```
public interface AdminRepo extends CrudRepository<Admin, Integer>{
```

```
}
```

citizensrepo.java

```
package com.example.capstone.AadharRestApi.Repository;
```

```
import org.springframework.data.repository.CrudRepository;
```

```
import com.example.capstone.AadharRestApi.Entity.Citizens;
```

```
public interface CitizensRepo extends CrudRepository<Citizens, Integer>{
```

```
}
```

userrepo.java

```
package com.example.capstone.AadharRestApi.Repository;

import org.springframework.data.repository.CrudRepository;

import com.example.capstone.AadharRestApi.Entity.User;

public interface UserRepo extends CrudRepository<User, Integer>{

}
```

adminnotfoundexception.java

```
package com.examples.capstone.AadharRestApi.Exception;

public class AdminNotFoundException extends RuntimeException{

    public AdminNotFoundException(int id) {

        super("Admin with id " + id + " is not Found.Pls Give another id!!");

    }

}
```

citizensnotfoundexception.java

```
package com.examples.capstone.AadharRestApi.Exception;

public class CitizenNotFoundException extends RuntimeException{

    public CitizenNotFoundException(int id) {
```

```
        super("Citizen with id " + id + " is not Found.Pls Give another id!!");  
    }  
  
}
```

usernotfoundexception.java

```
package com.examples.capstone.AadharRestApi.Exception;  
  
public class UserNotFoundException extends RuntimeException{  
  
    public UserNotFoundException(int id) {  
        super("User with id " + id + " is not Found.Pls Give another id!!");  
    }  
  
}
```

application.properties

```
server.port=8080  
  
spring.datasource.url=jdbc:mysql://localhost/maadhar_db  
  
spring.datasource.username=root  
  
spring.datasource.password=abdul123@  
  
spring.jpa.properties.hibernate.show_sql=true  
  
spring.jpa.properties.hibernate.format_sql=true  
  
spring.jpa.hibernate.ddl-auto=update  
  
logging.level.org.hibernate=INFO  
  
logging.level.org.hibernate.SQL=INFO  
  
logging.level.org.hibernate.type.descriptor.sql.BasicBinder=TRACE
```


logging.level.org.springframework=INFO

logging.level.org.apache=INFO

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  
    http://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>AadharBackend</artifactId>
```

```
    <version>0.0.1-SNAPSHOT</version>
```

```
    <name>AadharBackend</name>
```

```
    <description>Demo project for Spring Boot</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>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</artifactId>

</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>
```

dockerfile

```
FROM openjdk:17-slim

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

EXPOSE 8080

ENTRYPOINT ["java", "-jar", "app.jar"]
```

Testing:

aadharadmin.java:

```
package com.capstone;
```

```
import org.openqa.selenium.By;
```

```
import org.openqa.selenium.WebDriver;
```

```
import org.openqa.selenium.chrome.ChromeDriver;
```

```
import org.testng.annotations.BeforeMethod;
```

```
import org.testng.annotations.Test;
```

```
public class AadharAdmin {
```

```
    static WebDriver webdriver = null;
```

```
    @Test
```

```
    public void ChromeBrowser() {
```

```
        System.setProperty("webdriver.chrome.driver",
```

```
        "C:\\Users\\kvsivasankar\\eclipse-  
workspace\\Downloads\\chromedriver_win32\\chromedriver.exe");
```

```
        webdriver = new ChromeDriver();
```

```
        webdriver.manage().window().maximize();
```

```
        System.out.println("Chrome Browser is Opened!!");
```

```
    }
```

```
    @Test
```

```
public void TestHome() {  
    webdriver.get("http://localhost:4200/home");  
    try {  
        Thread.sleep(3000);  
    } catch (InterruptedException e) {  
        e.printStackTrace();  
    }  
    webdriver.findElement(By.xpath("/html[1]/body[1]/app-root[1]/app-home[1]/div[1]/div[1]/a[3]")).click();  
    try {  
        Thread.sleep(3000);  
    } catch (InterruptedException e) {  
        e.printStackTrace();  
    }  
    webdriver.get("http://localhost:4200/adminlogin");  
    try {  
        Thread.sleep(3000);  
    } catch (InterruptedException e) {  
        e.printStackTrace();  
    }  
    webdriver.findElement(By.id("admin")).sendKeys("admin");  
    webdriver.findElement(By.id("password")).sendKeys("12345");  
    webdriver.findElement(By.xpath("//button[normalize-space()='Login']")).click();  
    try {  
        Thread.sleep(3000);  
    } catch (InterruptedException e) {
```

```
        e.printStackTrace();  
    }  
  
}
```

```
}
```

aadharcitizen.java:

```
package com.capstone;
```

```
import org.openqa.selenium.By;
```

```
import org.openqa.selenium.WebDriver;
```

```
import org.openqa.selenium.chrome.ChromeDriver;
```

```
import org.testng.annotations.BeforeMethod;
```

```
import org.testng.annotations.Test;
```

```
public class AadharCitizen {
```

```
    static WebDriver webdriver = null;
```

```
    @Test
```

```
    public void ChromeBrowser() {
```

```
        System.setProperty("webdriver.chrome.driver",  
        "C:\\Users\\kvsivasankar\\eclipse-  
workspace\\Downloads\\chromedriver_win32\\chromedriver.exe");  
  
        webdriver = new ChromeDriver();  
  
        webdriver.manage().window().maximize();  
  
        System.out.println("Chrome Browser is Opened!!");  
  
    }
```

@Test

```
public void CitizenCheck() {  
    webdriver.get("http://localhost:4200/home");  
  
    try {  
        Thread.sleep(3000);  
    } catch (InterruptedException e) {  
        e.printStackTrace();  
    }  
  
    webdriver.findElement(By.xpath("/html[1]/body[1]/app-root[1]/app-  
home[1]/div[1]/div[1]/a[1]")).click();  
  
    try {  
        Thread.sleep(3000);  
    } catch (InterruptedException e) {  
        e.printStackTrace();  
    }  
  
    webdriver.get("http://localhost:4200/userlogin");  
}
```

```

        try {

            Thread.sleep(3000);

        } catch (InterruptedException e) {

            e.printStackTrace();

        }

        webdriver.findElement(By.id("citizen_id")).sendKeys("2");

        webdriver.findElement(By.id("password")).sendKeys("8096956409");

        webdriver.findElement(By.xpath("//button[normalize-space()='Login']")).click();

    }

}

```

aadhar register.java:

```

package com.capstone;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.Test;

public class AadharRegister {

    static WebDriver webdriver = null;

```



```

@Test

public void ChromeBrowser() {

    System.setProperty("webdriver.chrome.driver",

        "C:\\Users\\kvsivasankar\\eclipse-
workspace\\Downloads\\chromedriver_win32\\chromedriver.exe");

    webdriver = new ChromeDriver();

    webdriver.manage().window().maximize();

    System.out.println("Chrome Browser is Opened!!");

}

@Test

public void TestHome() {

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

    try {

        Thread.sleep(3000);

    } catch (InterruptedException e) {

        e.printStackTrace();

    }

    webdriver.findElement(By.xpath("/html[1]/body[1]/app-root[1]/app-
home[1]/div[1]/div[1]/a[2]")).click();

    webdriver.get("http://localhost:4200/register");

    try {

        Thread.sleep(3000);

    } catch (InterruptedException e) {

        e.printStackTrace();

    }

```

```

    }

    webdriver.findElement(By.id("name")).sendKeys("sivasankar");

    webdriver.findElement(By.id("dob")).sendKeys("2001-01-21");

    webdriver.findElement(By.id("emailid")).sendKeys("s@gmail.com");

    webdriver.findElement(By.id("address")).sendKeys("main raod ");

    webdriver.findElement(By.id("mobilenno")).sendKeys("8096956409");

    webdriver.findElement(By.id("gender")).sendKeys("male");

}

```

```

}

```

testngadmin.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.capstone.AadharAdmin"/>

        </classes>

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

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

```

testngcitizen.java:

```

<?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.capstone.AadharCitizen"/>

    </classes>

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

</suite> <!-- Suite -->
```

testngreg.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.capstone.AadharRegister"/>

    </classes>

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

</suite> <!-- Suite -->
```

pom.xml:

```
<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>

<groupId>org.mycompany</groupId>

<artifactId>AadharTesting</artifactId>

<version>0.0.1-SNAPSHOT</version>

<build>

  <sourceDirectory>src</sourceDirectory>

  <plugins>

    <plugin>

      <artifactId>maven-compiler-plugin</artifactId>

      <version>3.8.1</version>

      <configuration>

        <release>15</release>

      </configuration>

    </plugin>

    <plugin>

      <groupId>org.apache.maven.plugins</groupId>

      <artifactId>maven-surefire-plugin</artifactId>

      <version>3.0.0-M5</version>

      <configuration>

        <suiteXmlFiles>

          <suiteXmlFile>testng.xml</suiteXmlFile>

        </suiteXmlFiles>

      </configuration>

    </plugin>

  </plugins>
```

</build>

<dependencies>

<dependency>

 <groupId>org.seleniumhq.selenium</groupId>

 <artifactId>selenium-java</artifactId>

 <version>4.1.2</version>

</dependency>

<dependency>

 <groupId>org.testng</groupId>

 <artifactId>testng</artifactId>

 <version>7.5</version>

</dependency>

</dependencies>

</project>