mAadhar Application

SOURCE CODE:

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" />
</head>
<body>
<app-root></app-root>
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.6/umd/popper.min.js"></script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/js/bootstrap.min.js"></script>
</body>
</html>
```

Admin.component.html:

```
<h1 class="text-center"><i>Admin Login</i></h1>
<div
class="container mx-200 mt-10 border justify-center"
style="width: 400px; background-color: darkgrey"
>
<h4 class="text-center">Login Page</h4>
<form [formGroup]="signupForm" (ngSubmit)="onSubmit()" class="form-center">
<div class="row">
<div class="form-group">
<label for="username">Username</label>
<input
type="text"
id="username"
```

```
class="form-control"
formControlName="username"
/>
<span
class="small"
*ngIf="
!signupForm.get('username').valid &&
signupForm.get('username').touched
"
>Please enter a valid username</span
>
</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
class="small"
*ngIf="
!signupForm.get('password').valid &&
signupForm.get('password').touched
"
>Incorrect Password!</span
>
</div>
</div>
<button
class="btn btn-primary m-2"
type="submit"
[disabled]="!signupForm.valid"
>
```

```
Submit
</button>
</form>
</div>
```

Home.component.html:

```
<h4 class="text-center">Welcome to mAadhar Application Portal!!</h4>
<hr class="new5" />
<p class="text-center"><b>Please Login to Continue!!</b></p>
<div class="container mt-50 text-center">
<button class="btn btn-info mt-5" (click)="gotoAdmin()">Admin LogIn</button
><br />
<button class="btn btn-success mt-5" (click)="gotoCitizen()">
Citizen LogIn</button
><br />
<button class="btn btn-primary mt-5" (click)="gotoRegister()">
Register New Citizen
</button>
</div>
```

Myaadhar.component.html:

```
<h1 class="text-center">
<i><u>Admin Portal</u></i>
</h1>
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
<a class="navbar-brand" style="color: white">AadharApplication</a>
<button
class="navbar-toggler"
type="button"
data-toggle="collapse"
data-target="#navbarColor02"
aria-controls="navbarColor02"
aria-expanded="false"
aria-label="Toggle navigation"
>
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarColor02">
<!-- <ul class="navbar-nav mr-auto">
```

```html
<li class="nav-item active">
<a class="nav-link" (click)="onOpenModal(null, 'add')"
>New Citizen Register <span class="sr-only">(current)</span></a
>
</li>
</ul> -->
<form class="form-inline my-2 my-lg-0">
<input
type="search"
(ngModelChange)="searchCitizen(key.value)"
#key="ngModel"
ngModel
name="key"
id="searchName"
class="form-control mr-sm-2"
placeholder="Search Citizen..."
required
/>
</form>
</div>
</nav>
<div class="container" id="main-container">
<div class="row">
<div *ngFor="let record of records" class="col-md-6 col-xl-3">
<div class="card m-b-30">
<div class="card-body row">
<div class="col-6">
<a href=""
><img
src="{{ record?.imageUrl }}"
alt=""
class="img-fluid rounded-circle w-60"
/></a>
</div>
<div class="col-6 card-title align-self-center mb-0">
<h5>{{ record?.fullname }}</h5>
<p class="m-0">{{ record?.gender }}</p>
```

```
</div>
</div>
<ul class="list-group list-group-flush">
<li class="list-group-item">
<i class="fa fa-phone float-right"></i>DOB :
{{ record?.dob }}
</li>
<li class="list-group-item">
<i class="fa fa-phone float-right"></i>Address :
{{ record?.address }}
</li>
<li class="list-group-item">
<i class="fa fa-envelope float-right"></i>{{ record?.email }}
</li>
<li class="list-group-item">
<i class="fa fa-phone float-right"></i>Phone : {{ record?.phone }}
</li>
</ul>
<div class="card-body">
<div class="float-right btn-group btn-group-sm">
<a
(click)="onOpenModal(record, 'edit')"
class="btn btn-primary tooltips"
data-placement="top"
data-original-title="Edit"
><i class="fa fa-pencil">Edit</i>
</a>
<a
(click)="onOpenModal(record, 'delete')"
class="btn btn-secondary tooltips"
data-placement="top"
data-original-title="Delete"
><i class="fa fa-times">Delete</i></a
>
</div>
</div>
</div>
```

```
</div>
</div>
<!-- Add Employee Modal -->
<div
class="modal fade"
id="addCitizenModal"
tabindex="-1"
role="dialog"
aria-labelledby="addEmployeeModalLabel"
aria-hidden="true"
>
<div class="modal-dialog" role="document">
<div class="modal-content">
<div class="modal-header">
<h5 class="modal-title" id="addEmployeeModalLabel">
New Citizen Register
</h5>
<button
type="button"
class="close"
data-dismiss="modal"
aria-label="Close"
>
<span aria-hidden="true">&times;</span>
</button>
</div>
<div class="modal-body">
<form #addForm="ngForm" (ngSubmit)="onAddCitizen(addForm)">
<div class="form-group">
<label for="fullname">Name</label>
<input
type="text"
ngModel
name="fullname"
class="form-control"
id="fullname"
placeholder="Name"
```

```
required
/>
</div>
<div class="form-group">
<label for="dob">D.O.B</label>
<input
type="date"
ngModel
name="dob"
class="form-control"
id="dob"
placeholder="dob"
required
/>
</div>
<div class="form-group">
<label for="email">Email Address</label>
<input
type="email"
ngModel
name="email"
class="form-control"
id="email"
placeholder="Email"
required
/>
</div>
<div class="form-group">
<label for="address">Home Address</label>
<input
type="text"
ngModel
name="address"
class="form-control"
id="address"
placeholder="Address"
required
```

```
/>
</div>
<div class="form-group">
<label for="gender">Gender</label>
<input
type="text"
ngModel
name="gender"
class="form-control"
id="gender"
placeholder="Gender"
required
/>
</div>
<div class="form-group">
<label for="phone">Phone</label>
<input
type="text"
ngModel
name="phone"
class="form-control"
id="phone"
placeholder="Phone"
required
/>
</div>
<div class="form-group">
<label for="phone">Image URL</label>
<input
type="text"
ngModel
name="imageUrl"
class="form-control"
id="imageUrl"
placeholder="Image URL"
required
/>
```

```
</div>
<div class="modal-footer">
<button
type="button"
id="add-record-form"
class="btn btn-secondary"
data-dismiss="modal"
>
Close
</button>
<button
[disabled]="addForm.invalid"
type="submit"
class="btn btn-primary"
>
Save changes
</button>
</div>
</form>
</div>
</div>
</div>
</div>
<!-- Edit Modal -->
<div
class="modal fade"
id="updateCitizenModal"
tabindex="-1"
role="dialog"
aria-labelledby="employeeEditModalLabel"
aria-hidden="true"
>
<div class="modal-dialog" role="document">
<div class="modal-content">
<div class="modal-header">
<h5 class="modal-title" id="updateEmployeeModalLabel">
Edit Citizen {{ editRecord?.fullname }}
```

```
</h5>
<button
type="button"
class="close"
data-dismiss="modal"
aria-label="Close"
>
<span aria-hidden="true">&times;</span>
</button>
</div>
<div class="modal-body">
<form #editForm="ngForm">
<div class="form-group">
<label for="fullname">Name</label>
<input
type="text"
ngModel="{{ editRecord?.fullname }}"
name="fullname"
class="form-control"
id="fullname"
aria-describedby="emailHelp"
placeholder="Name"
/>
</div>
<div class="form-group">
<label for="gender">Gender</label>
<input
type="text"
ngModel="{{ editRecord?.gender }}"
name="gender"
class="form-control"
id="gender"
aria-describedby="emailHelp"
placeholder="Gender"
/>
</div>
<input
```

```
type="hidden"
ngModel="{{ editRecord?.id }}"
name="id"
class="form-control"
id="id"
placeholder="Email"
/>
<div class="form-group">
<label for="email">Email Address</label>
<input
type="email"
ngModel="{{ editRecord?.email }}"
name="email"
class="form-control"
id="email"
placeholder="Email"
/>
</div>
<div class="form-group">
<label for="dob">D.O.B</label>
<input
type="date"
ngModel="{{ editRecord?.dob }}"
name="dob"
class="form-control"
id="dob"
placeholder="DOB"
/>
</div>
<div class="form-group">
<label for="address">Home Address</label>
<input
type="text"
ngModel="{{ editRecord?.address }}"
name="address"
class="form-control"
id="address"
```

```
placeholder="Address"
/>
</div>
<div class="form-group">
<label for="phone">Phone</label>
<input
type="text"
ngModel="{{ editRecord?.phone }}"
name="phone"
class="form-control"
id="phone"
name="phone"
placeholder="Phone"
/>
</div>
<div class="form-group">
<label for="phone">Image URL</label>
<input
type="text"
ngModel="{{ editRecord?.imageUrl }}"
name="imageUrl"
class="form-control"
id="imageUrl"
placeholder="Image URL"
/>
</div>
<div class="modal-footer">
<button
type="button"
id=""
data-dismiss="modal"
class="btn btn-secondary"
>
Close
</button>
<button
(click)="onUpdateCitizen(editForm.value)"
```

```
data-dismiss="modal"
class="btn btn-primary"
>
Save changes
</button>
</div>
</form>
</div>
</div>
</div>
</div>
<!-- Delete Modal -->
<div
class="modal fade"
id="deleteCitizenModal"
tabindex="-1"
role="dialog"
aria-labelledby="deleteModelLabel"
aria-hidden="true"
>
<div class="modal-dialog" role="document">
<div class="modal-content">
<div class="modal-header">
<h5 class="modal-title" id="deleteModelLabel">Delete Citizen</h5>
<button
type="button"
class="close"
data-dismiss="modal"
aria-label="Close"
>
<span aria-hidden="true">&times;</span>
</button>
</div>
<div class="modal-body">
<p>
Are you sure you want to delete
{{ deleteRecord?.fullname }}?
```

```
</p>
<div class="modal-footer">
<button
type="button"
class="btn btn-secondary"
data-dismiss="modal"
>
No
</button>
<button
(click)="onDeleteCitizen(deleteRecord?.id)"
class="btn btn-danger"
data-dismiss="modal"
>
Yes
</button>
</div>
</div>
</div>
</div>
</div>
</div>
<!-- Notification for no employees -->
<div *ngIf="records?.length == 0" class="col-lg-12 col-md-12 col-xl-12">
<div class="alert alert-info" role="alert">
<h4 class="alert-heading">NO EMPLOYEES!</h4>
<p>No Employees were found.</p>
</div>
</div>
```

Myaadhar2.component.html:

```
<h1 class="text-center">
<i><u>Citizen's Portal</u></i>
</h1>
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
<a class="navbar-brand" style="color: white">AadharApplication</a>
<button
class="navbar-toggler"
```

```
type="button"
data-toggle="collapse"
data-target="#navbarColor02"
aria-controls="navbarColor02"
aria-expanded="false"
aria-label="Toggle navigation"
>
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarColor02">
<ul class="navbar-nav mr-auto">
<li class="nav-item active">
<a class="nav-link" (click)="onOpenModal(null, 'add')"
>New Citizen Register <span class="sr-only">(current)</span></a
>
</li>
</ul>
<form class="form-inline my-2 my-lg-0">
<input
type="search"
(ngModelChange)="searchCitizen(key.value)"
#key="ngModel"
ngModel
name="key"
id="searchName"
class="form-control mr-sm-2"
placeholder="Enter your Name.."
required
/>
</form>
</div>
</nav>
<div class="container" id="main-container">
<div class="row">
<div *ngFor="let record of records" class="col-md-6 col-xl-3">
<div class="card m-b-30">
<div class="card-body row">
```

```html
<div class="col-6">
<a href=""
><img
src="{{ record?.imageUrl }}"
alt=""
class="img-fluid rounded-circle w-60"
/></a>
</div>
<div class="col-6 card-title align-self-center mb-0">
<h5>{{ record?.fullname }}</h5>
<p class="m-0">{{ record?.gender }}</p>
</div>
</div>
<ul class="list-group list-group-flush">
<li class="list-group-item">
<i class="fa fa-phone float-right"></i>DOB :
{{ record?.dob }}
</li>
<li class="list-group-item">
<i class="fa fa-phone float-right"></i>Address :
{{ record?.address }}
</li>
<li class="list-group-item">
<i class="fa fa-envelope float-right"></i>{{ record?.email }}
</li>
<li class="list-group-item">
<i class="fa fa-phone float-right"></i>Phone : {{ record?.phone }}
</li>
</ul>
<div class="card-body">
<div class="float-right btn-group btn-group-sm">
<a
(click)="onOpenModal(record, 'edit')"
class="btn btn-primary tooltips"
data-placement="top"
data-original-title="Edit"
><i class="fa fa-pencil">Edit</i>
```

```
</a>
<a
(click)="OnDeleteRequest()"
class="btn btn-secondary tooltips"
data-placement="top"
data-original-title="Delete"
><i class="fa fa-times">Delete</i></a
>
</div>
</div>
</div>
</div>
</div>
<!-- Add Employee Modal -->
<div
class="modal fade"
id="addCitizenModal"
tabindex="-1"
role="dialog"
aria-labelledby="addEmployeeModalLabel"
aria-hidden="true"
>
<div class="modal-dialog" role="document">
<div class="modal-content">
<div class="modal-header">
<h5 class="modal-title" id="addEmployeeModalLabel">
New Citizen Register
</h5>
<button
type="button"
class="close"
data-dismiss="modal"
aria-label="Close"
>
<span aria-hidden="true">&times;</span>
</button>
</div>
```

```
<div class="modal-body">
<form #addForm="ngForm" (ngSubmit)="onAddCitizen(addForm)">
<div class="form-group">
<label for="fullname">Name</label>
<input
type="text"
ngModel
name="fullname"
class="form-control"
id="fullname"
placeholder="Name"
required
/>
</div>
<div class="form-group">
<label for="dob">D.O.B</label>
<input
type="date"
ngModel
name="dob"
class="form-control"
id="dob"
placeholder="dob"
required
/>
</div>
<div class="form-group">
<label for="email">Email Address</label>
<input
type="email"
ngModel
name="email"
class="form-control"
id="email"
placeholder="Email"
required
/>
```

```
</div>
<div class="form-group">
<label for="address">Home Address</label>
<input
type="text"
ngModel
name="address"
class="form-control"
id="address"
placeholder="Address"
required
/>
</div>
<div class="form-group">
<label for="gender">Gender</label>
<input
type="text"
ngModel
name="gender"
class="form-control"
id="gender"
placeholder="Gender"
required
/>
</div>
<div class="form-group">
<label for="phone">Phone</label>
<input
type="text"
ngModel
name="phone"
class="form-control"
id="phone"
placeholder="Phone"
required
/>
</div>
```

```
<div class="form-group">
<label for="phone">Image URL</label>
<input
type="text"
ngModel
name="imageUrl"
class="form-control"
id="imageUrl"
placeholder="Image URL"
required
/>
</div>
<div class="modal-footer">
<button
type="button"
id="add-record-form"
class="btn btn-secondary"
data-dismiss="modal"
>
Close
</button>
<button
[disabled]="addForm.invalid"
type="submit"
class="btn btn-primary"
>
Save changes
</button>
</div>
</form>
</div>
</div>
</div>
</div>
<div
class="modal fade"
id="updateCitizenModal"
```

```
tabindex="-1"
role="dialog"
aria-labelledby="employeeEditModalLabel"
aria-hidden="true"
>
<div class="modal-dialog" role="document">
<div class="modal-content">
<div class="modal-header">
<h5 class="modal-title" id="updateEmployeeModalLabel">
Edit Citizen {{ editRecord?.fullname }}
</h5>
<button
type="button"
class="close"
data-dismiss="modal"
aria-label="Close"
>
<span aria-hidden="true">&times;</span>
</button>
</div>
<div class="modal-body">
<form #editForm="ngForm">
<div class="form-group">
<label for="fullname">Name</label>
<input
type="text"
ngModel="{{ editRecord?.fullname }}"
name="fullname"
class="form-control"
id="fullname"
aria-describedby="emailHelp"
placeholder="Name"
/>
</div>
<div class="form-group">
<label for="gender">Gender</label>
<input
```

```
type="text"
ngModel="{{ editRecord?.gender }}"
name="gender"
class="form-control"
id="gender"
aria-describedby="emailHelp"
placeholder="Gender"
/>
</div>
<input
type="hidden"
ngModel="{{ editRecord?.id }}"
name="id"
class="form-control"
id="id"
placeholder="Email"
/>
<div class="form-group">
<label for="email">Email Address</label>
<input
type="email"
ngModel="{{ editRecord?.email }}"
name="email"
class="form-control"
id="email"
placeholder="Email"
/>
</div>
<div class="form-group">
<label for="dob">D.O.B</label>
<input
type="date"
ngModel="{{ editRecord?.dob }}"
name="dob"
class="form-control"
id="dob"
placeholder="DOB"
```

```
/>
</div>
<div class="form-group">
<label for="address">Home Address</label>
<input
type="text"
ngModel="{{ editRecord?.address }}"
name="address"
class="form-control"
id="address"
placeholder="Address"
/>
</div>
<div class="form-group">
<label for="phone">Phone</label>
<input
type="text"
ngModel="{{ editRecord?.phone }}"
name="phone"
class="form-control"
id="phone"
name="phone"
placeholder="Phone"
/>
</div>
<div class="form-group">
<label for="phone">Image URL</label>
<input
type="text"
ngModel="{{ editRecord?.imageUrl }}"
name="imageUrl"
class="form-control"
id="imageUrl"
placeholder="Image URL"
/>
</div>
<div class="modal-footer">
```

```
<button
type="button"
id=""
data-dismiss="modal"
class="btn btn-secondary"
>
Close
</button>
<button
(click)="OnUpdateRequest()"
data-dismiss="modal"
class="btn btn-primary"
>
Save changes
</button>
</div>
</form>
</div>
</div>
</div>
</div>
</div>
```

User.component.html:

```
<h1 class="text-center"><i>Citizen Login</i></h1>
<div
class="container mx-200 mt-10 border justify-center"
style="width: 400px; background-color: darkgrey"
>
<h4 class="text-center">Login Page</h4>
<form [formGroup]="signupForm" (ngSubmit)="onSubmit()" class="form-center">
<div class="row">
<div class="form-group">
<label for="username">Username</label>
<input
type="text"
id="username"
class="form-control"
```

```
formControlName="username"
/>
<span
class="small"
*ngIf="
!signupForm.get('username').valid &&
signupForm.get('username').touched
"
>Please enter a valid username</span
>
</div>
</div>
<div class="row">
<div class="form-group">
<span><i>Enter Mobile number as Password</i></span
><br />
<label for="password">Password</label>
<input
type="text"
id="password"
class="form-control"
formControlName="password"
/>
<span
class="small"
*ngIf="
!signupForm.get('password').valid &&
signupForm.get('password').touched
"
>Incorrest Password!</span
>
</div>
</div>
<button
class="btn btn-primary m-2"
type="submit"
[disabled]="!signupForm.valid"
```

```
>
Submit
</button>
</form>
</div>
```

Delete.java:

```
package com.example.testng;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.Test;
public class Delete {
WebDriver webdriver = null;
String URL = "http://localhost:4200/";
@Test
public void delete() {
System.setProperty("webdriver.chrome.driver",
"C:\\Users\\hpp\\Downloads\\chromedriver_win32\\chromedriver.exe");
WebDriver webdriver = new ChromeDriver();
webdriver.get(URL);
webdriver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
webdriver.manage().window().maximize();
webdriver.findElement(By.xpath("//button[normalize-space()='Admin
LogIn']")).click();
webdriver.findElement(By.xpath("//input[@id='username']")).sendKeys("Admin");
webdriver.findElement(By.xpath("//input[@id='password']")).sendKeys("1234");
webdriver.findElement(By.xpath("//button[normalizespace()='Submit']")).click();
webdriver.findElement(By.xpath("//body[1]/approot[1]/div[1]/div[2]/div[1]/appmyaa
dhar[1]/div[1]/div[1]/div[8]/div[1]/div[2]/div[1]/a[2]/i[1]")).click();
webdriver.findElement(By.xpath("//button[normalizespace()='Yes']")).click();
System.out.println("Deleted Successfully");
}
}
```

Registration.java:

```
package com.example.testng;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.Test;
public class Registration {
WebDriver webdriver = null;
String URL = "http://localhost:4200/";
@Test
public void registration() {
System.setProperty("webdriver.chrome.driver",
"C:\\Users\\hpp\\Downloads\\chromedriver_win32\\chromedriver.exe");
WebDriver webdriver = new ChromeDriver();
webdriver.get(URL);
webdriver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
webdriver.manage().window().maximize();
webdriver.findElement(By.xpath("//button[normalize-space()='Register New
Citizen']")).click();
webdriver.findElement(By.xpath("//ul[@class='navbar-nav
mrauto']//a[@class='nav-link']")).click();
webdriver.findElement(By.xpath("//form[@class='ng-untouched ng-pristine
ng-invalid']//input[@id='fullname']"))
.sendKeys("Jack Miller");
webdriver.findElement(By.xpath("(//input[@id='dob'])[1]"))
.sendKeys("2/2/2008");
webdriver.findElement(By.xpath("(//input[@id='email'])[1]"))
.sendKeys("jack@example.com");
webdriver.findElement(By.xpath("(//input[@id='address'])[1]"))
.sendKeys("A-10,CityCenter,Puri");
webdriver.findElement(By.xpath("(//input[@id='gender'])[1]"))
.sendKeys("Male");
webdriver.findElement(By.xpath("(//input[@id='phone'])[1]"))
.sendKeys("99999777");
webdriver.findElement(By.xpath("(//input[@id='imageUrl'])[1]"))
.sendKeys("https://bootdey.com/img/Content/avatar/avatar6.png");
webdriver.findElement(By.xpath("//button[@type='submit']")).click();
System.out.println("Registration Tested Succesfully");
}
}
```

Update.java:

```
package com.example.testng;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.Test;
public class Update {
WebDriver webdriver = null;
String URL = "http://localhost:4200/";
@Test
public void update() {
System.setProperty("webdriver.chrome.driver",
"C:\\Users\\hpp\\Downloads\\chromedriver_win32\\chromedriver.exe");
WebDriver webdriver = new ChromeDriver();
webdriver.get(URL);
webdriver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
webdriver.manage().window().maximize();
webdriver.findElement(By.xpath("//button[normalize-space()='Admin
LogIn']")).click();
webdriver.findElement(By.xpath("//input[@id='username']")).sendKeys("Admin");
webdriver.findElement(By.xpath("//input[@id='password']")).sendKeys("1234");
webdriver.findElement(By.xpath("//button[normalizespace()='Submit']")).click();
webdriver.findElement(By.xpath("//div[8]//div[1]//div[2]//div[1]//a[1]//i[1]")).click();
webdriver.findElement(By.xpath("//input[@ng-reflectmodel='https://bootdey.com/i
mg/Conten']")).clear();
webdriver.findElement(By.xpath("//input[@ng-reflectmodel='https://bootdey.com/i
mg/Conten']"))
.sendKeys("https://bootdey.com/img/Content/avatar/avatar7.png");
webdriver.findElement(By.xpath("//button[@datadismiss='modal'][normalize-spac
e()='Save changes']")).click();
System.out.println("Updation Tested Succesfully");
}
}
```

AppController.java:

```
package com.example.demo.controller;
```

```
import java.util.List;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
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.example.demo.entity.myaadhar;
import com.example.demo.service.AppService;
@RestController
@RequestMapping("/aadhar")
public class AppController {
private final AppService appservice;
public AppController(AppService appservice) {
this.appservice=appservice;
}
@GetMapping("/all")
public ResponseEntity<List<myaadhar>> getrecords(){
List<myaadhar> records=appservice.findallRecords();
return new ResponseEntity<>(records,HttpStatus.OK);
}
@GetMapping("/find/{id}")
public ResponseEntity<myaadhar> getaadharById(@PathVariable("id") int id){
myaadhar x=appservice.findmyaadharById(id);
return new ResponseEntity<>(x,HttpStatus.OK);
}
@PostMapping("/add")
ResponseEntity<myaadhar> addRequest(@RequestBody myaadhar x){
myaadhar y=appservice.add(x);
return new ResponseEntity<>(y,HttpStatus.CREATED);
}
@PutMapping("/update")
ResponseEntity<myaadhar> updateRequest(@RequestBody myaadhar x){
```

```
myaadhar y=appservice.update(x);
return new ResponseEntity<>(y,HttpStatus.OK);
}
@DeleteMapping("/delete/{id}")
ResponseEntity<?> deleteRequest(@PathVariable("id") int id){
appservice.delete(id);
return new ResponseEntity<>(HttpStatus.OK);
}
}
```

Myaadhar.java:

```
package com.example.demo.entity;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
@Entity
public class myaadhar {
@Id
@GeneratedValue(strategy = GenerationType.IDENTITY)
private int id;
private String fullname;
private String gender;
private String address;
private String dob;
private int phone;
private String email;
private String imageUrl;
public String getImageUrl() {
return imageUrl;
}
public void setImageUrl(String imageUrl) {
this.imageUrl = imageUrl;
}
public int getId() {
return id;
}
public void setId(int id) {
```

```
this.id = id;
}
public String getFullname() {
return fullname;
}
public void setFullname(String fullname) {
this.fullname = fullname;
}
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;
}
public String getDob() {
return dob;
}
public void setDob(String dob) {
this.dob = dob;
}
public int getPhone() {
return phone;
}
public void setPhone(int phone) {
this.phone = phone;
}
public String getEmail() {
return email;
}
public void setEmail(String email) {
this.email = email;
```

```
}
@Override
public String toString() {
return "myaadhar [id=" + id + ", fullname=" + fullname + ", gender=" + gender + ",
address=" + address
+ ", dob=" + dob + ", phone=" + phone + ", email=" + email + ", imageUrl=" +
imageUrl + "]";
}
}
```

Myaadharrepository.java:

```
package com.example.demo.repository;
import java.util.Optional;
import org.springframework.data.jpa.repository.JpaRepository;
import com.example.demo.entity.myaadhar;
public interface MyAadharRepository extends JpaRepository<myaadhar,
Integer> {
Optional<myaadhar> findById(int id);
}
```

Appservice.java:

```
package com.example.demo.service;
import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import com.example.demo.entity.myaadhar;
import com.example.demo.exceptions.UserNotFoundException;
import com.example.demo.repository.MyAadharRepository;
@Service
public class AppService {
private final MyAadharRepository repo;
@Autowired
public AppService(MyAadharRepository repo) {
this.repo = repo;
}
public List<myaadhar> findallRecords(){
return repo.findAll();
}
public myaadhar add(myaadhar x) {
```

```
return repo.save(x);
}
public myaadhar update(myaadhar x) {
return repo.save(x);
}
public myaadhar findmyaadharById(int id) {
return repo.findById(id)
.orElseThrow(() -> new UserNotFoundException("User by id " + id + " not
found"));
}
public void delete(int id) {
repo.deleteById(id);
}
}
```

MyApplication.java:

```
package com.example.demo;
import java.util.Arrays;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.Bean;
import org.springframework.web.cors.CorsConfiguration;
import org.springframework.web.cors.UrlBasedCorsConfigurationSource;
import org.springframework.web.filter.CorsFilter;
@SpringBootApplication
public class MyAppApplication {
public static void main(String[] args) {
SpringApplication.run(MyAppApplication.class, args);
}
@Bean
public CorsFilter corsFilter() {
CorsConfiguration corsconfiguration = new CorsConfiguration();
corsconfiguration.setAllowCredentials(true);
corsconfiguration.setAllowedOrigins(Arrays.asList("http://localhost:4200"));
corsconfiguration.setAllowedHeaders(Arrays.asList("Origin",
"Access-Control-AllowOrigin", "Content-Type",
"Accept", "Authorization", "Origin, Accept", "X-Requested-With",
"AccessControl-Request-Method",
"Access-Control-Request-Headers"));
corsconfiguration.setExposedHeaders(Arrays.asList("Origin", "Content-Type",
"Accept", "Authorization",
"Access-Control-Allow-Origin", "Access-Control-Allow-Origin",
"Access-ControlAllow-Credentials"));
corsconfiguration.setAllowedMethods(Arrays.asList("GET", "PUT", "POST",
"DELETE", "OPTIONS"));
UrlBasedCorsConfigurationSource urlconfig = new
UrlBasedCorsConfigurationSource();
urlconfig.registerCorsConfiguration("/**", corsconfiguration);
return new CorsFilter(urlconfig);
}
}
```

Application.properties:

```
server.port=8081
spring.datasource.url=jdbc:mysql://localhost:3306/aadhar
spring.datasource.username=root
spring.datasource.password=Daga#2023
spring.jpa.properties.hibernate.format_sql=true
spring.jpa.hibernate.ddl-auto=update
spring.mvc.view.prefix=/WEB-INF/views/
spring.mvc.view.suffix=.jsp
logging.level.org.hibernate=INFO
logging.level.org.hibernate.SQL=INFO
logging.level.org.springframework=ERROR
logging.level.org.apache=ERROR
spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL5Dialect
spring.jpa.properties.hibernate.globally_quoted_identifiers=true
```