

# // Source code for mAadhar Application

## //Backend-code

### // application.properties

```
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.url=jdbc:mysql://localhost:3306/aadharapp
spring.datasource.username=root
spring.datasource.password=1234
#spring.jpa.show-sql=true
#spring.jpa.hibernate.ddl-auto=update
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect
```

### //Login.java

```
package com.ecommerce.entity;

import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
```

@Entity

public class Login {

    @Id

    private String emailid;

    @GeneratedValue(strategy = GenerationType.AUTO)

    private int id;

    private String password;

    @Column(name = "typeofuser")

    private String typeOfUser;

    private String name;

    private String address;

    private double phone;

    private String dob;

    public String getEmailid() {

        return emailid;

    }

```
public void setEmailid(String emailid) {  
    this.emailid = emailid;  
}
```

```
public String getPassword() {  
    return password;  
}
```

```
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 double getPhone() {  
    return phone;  
}  
  
public void setPhone(double phone) {  
    this.phone = phone;  
}  
  
public String getDob() {  
    return dob;  
}  
  
public void setDob(String dob) {  
    this.dob = dob;  
}  
  
public void setPassword(String password) {  
    this.password = password;  
}  
  
public String getTypeOfUser() {  
    return typeOfUser;  
}
```

```
}
```

```
public void setTypeOfUser(String typeOfUser) {
```

```
    this.typeOfUser = typeOfUser;
```

```
}
```

```
    public Login(int id, String emailid, String password, String  
typeOfUser, String name, String address, double phone,
```

```
        String dob) {
```

```
    super();
```

```
    this.id = id;
```

```
    this.emailid = emailid;
```

```
    this.password = password;
```

```
    this.typeOfUser = typeOfUser;
```

```
    this.name = name;
```

```
    this.address = address;
```

```
    this.phone = phone;
```

```
    this.dob = dob;
```

```
}
```

```
public Login() {
```

```
    super();
```

```
        // TODO Auto-generated constructor stub
    }

    @Override
    public String toString() {
        return "Login [id=" + id + ", emailid=" + emailid + ",
password=" + password + ", typeOfUser=" + typeOfUser
        + ", name=" + name + ", address=" + address +
        ", phone=" + phone + ", dob=" + dob + "]";
    }

}
```

### **//LoginController.java**

```
package com.ecommerce.controllers;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.MediaType;
import org.springframework.web.bind.annotation.CrossOrigin;
import org.springframework.web.bind.annotation.GetMapping;
```

```
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.ecommerce.entity.Login;
import com.ecommerce.services.LoginService;
```

```
@RestController
@RequestMapping("login")
@CrossOrigin
public class LoginController {
```

```
    @Autowired
    LoginService ls;
```

```
    @GetMapping(value = "note")
    public String note() {
        return ls.note();
    }
```

```
        @PostMapping(value = "signIn", consumes =
MediaType.APPLICATION_JSON_VALUE)

        public String signIn(@RequestBody Login login) {

            System.out.println(login);

            return ls.signIn(login);

        }
```

```
        @PostMapping(value = "signUp", consumes =
MediaType.APPLICATION_JSON_VALUE)

        public String signUp(@RequestBody Login login) {

            return ls.signUp(login);

        }
```

```
}
```

**//OperationController.java**

```
package com.ecommerce.controllers;
```



```
import java.util.List;
```

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

```
import org.springframework.http.MediaType;
```

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

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

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

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

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

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

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

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

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

```
import com.ecommerce.entity.Login;
```

```
import com.ecommerce.services.LoginService;
```

```
@RestController
```

```
@RequestMapping("operation")
```

```
@CrossOrigin
```

```
public class OperationController {
```

@Autowired

LoginService ls;

@PostMapping(value = "apply", consumes =  
MediaType.APPLICATION\_JSON\_VALUE)

```
public String apply(@RequestBody Login login) {  
    return ls.storeUser(login);  
}
```

@PatchMapping(value = "update", consumes =  
MediaType.APPLICATION\_JSON\_VALUE)

```
public String update(@RequestBody Login login) {  
    return ls.update(login);  
}
```

@GetMapping(value = "getAll", produces =  
MediaType.APPLICATION\_JSON\_VALUE)

```
public List<Login> getAllUser(){  
    return ls.getAll();  
}
```

```
    @GetMapping(value = "getUser/{id}", produces =  
MediaType.APPLICATION_JSON_VALUE)
```

```
    public List<Login> getOneUser(@PathVariable("id") int id){  
        return ls.findById(id);  
    }
```

```
    @DeleteMapping(value = "delete/{emailid}")
```

```
    public String deleteUser(@PathVariable("emailid") String  
emailid){  
        return ls.delete(emailid);  
    }
```

```
}
```

```
//LoginRepository
```

```
package com.ecommerce.repositries;
```

```
import java.util.List;
```

```
import java.util.Optional;
```

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

```
import org.springframework.data.jpa.repository.Query;
```

```
import org.springframework.data.repository.query.Param;
```

```
import org.springframework.stereotype.Repository;
```

```
import com.ecommerce.controllers.LoginController;
```

```
import com.ecommerce.entity.Login;
```

```
@Repository
```

```
public interface LoginRepository extends JpaRepository<Login,  
String> {
```

```
    @Query("select l from Login l where l.id = :id")
```

```
    public List<Login> userById(@Param("id") int id);
```

```
    public void save1(LoginController citizen);
```

```
    public Optional<LoginController> findById(Long id);
```

```
public void deleteById(Long id);
```

```
public void save(LoginController citizen);
```

```
}
```

## **//LoginServices**

```
package com.ecommerce.services;
```

```
import java.util.List;
```

```
import java.util.Optional;
```

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

```
import org.springframework.stereotype.Service;
```

```
import com.ecommerce.entity.Login;
```

```
import com.ecommerce.repositories.LoginRepository;
```

@Service

public class LoginService {

    @Autowired

    LoginRepository lr;

    public String signIn(Login login) {

        Optional<Login> result = lr.findById(login.getEmailid());

        if(result.isPresent()) {

            Login ll = result.get();

            if(ll.getPassword().equals(login.getPassword())) {

                if(login.getTypeOfUser().equals(ll.getTypeOfUser()) &&  
login.getTypeOfUser().equals("admin")) {

                    return "Admin sucessfully

login";

        }else

        if(login.getTypeOfUser().equals(ll.getTypeOfUser()) &&  
login.getTypeOfUser().equals("user")){

            return "User successfully

login";

        }else {

```

        return "Invalid details";
    }
    }else {
        return "Invalid password";
    }
}
}else {
    return "Invalid emailId";
}
}

```

```

public String signUp(Login login) {
    Optional<Login> result = lr.findById(login.getEmailid());
    if(result.isPresent()) {
        return "Email Id already exists";
    }else {
        // Login ll = result.get();
        if(login.getTypeOfUser().equals("admin")) {
            return "You can't create admin account";
        }else {
            lr.save(login);
        }
    }
}

```

```
        return "Account created successfully";
    }

}

}
```

```
public List<Login> getAll(){
    return lr.findAll();
}
```

```
public List<Login> findById(int id) {
    return lr.userById(id);
}
```

```
public String delete(String emailid) {
    Optional<Login> result = lr.findById(emailid);
    if(result.isPresent()) {
        Login l = result.get();
        lr.delete(l);
        return "User deleted successfully";
    }else {
```



```
        return "User with this id not present";
    }
}

public String update(Login login) {
    Optional<Login> result = lr.findById(login.getEmailid());
    if(result.isPresent()) {
        Login l = result.get();
        l.setAddress(login.getAddress());
        l.setDob(login.getDob());
        l.setPhone(login.getPhone());
        return "User updated successfully";
    }else {
        return "User with this id not present";
    }
}
```

```
public String storeUser(Login login) {
    lr.save(login);
    return "User applied for aadhar card";
}
```

```

    public String note() {

        return "this is an application";

    }

}

```

## //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>3.1.0</version>
        <relativePath /> <!-- lookup parent from repository -->
    </parent>
    <groupId>com.ecommerce</groupId>
    <artifactId>spring-boot-mvc-demo</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>spring-boot-mvc-demo</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-web</artifactId>

```

```

</dependency>

<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>

<dependency>
    <groupId>com.mysql</groupId>
    <artifactId>mysql-connector-j</artifactId>
    <scope>runtime</scope>
</dependency>
<!--
    https://mvnrepository.com/artifact/jakarta.servlet.jsp.jstl/jakarta.s
    ervlet.jsp.jstl-api -->
<dependency>
    <groupId>jakarta.servlet.jsp.jstl</groupId>
    <artifactId>jakarta.servlet.jsp.jstl-api</artifactId>
    <version>3.0.0</version>
</dependency>

<!-- JSP support in Spring -->
<dependency>
    <groupId>org.apache.tomcat.embed</groupId>
    <artifactId>tomcat-embed-jasper</artifactId>
    <scope>provided</scope>
</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>

```

**// MySql database aadharapp**

show databases;

use Aadharapp;

CREATE TABLE citizens (

mobileNo VARCHAR(15) PRIMARY KEY,

name VARCHAR(100) NOT NULL,

dob DATE NOT NULL,

address VARCHAR(200) NOT NULL,

emailId VARCHAR(100) NOT NULL,

gender ENUM('Male', 'Female', 'Other') NOT NULL

);

INSERT INTO citizens (mobileNo, name, dob, address, emailId,  
gender)

VALUES

('1234567890', 'Chandan Kumar ', '2001-05-11', 'Patna Bihar, Jamui',  
'ChandanKumar@gmail.com', 'Male'),

('7976694711', 'Uttam Patel', '2011-08-23', '2/5 Heerabagh Flats',  
'uttampatel0811@gmail.com', 'Male');

SHOW TABLES;

Select \* from citizens;

use Aadharapp;

CREATE TABLE issueAadhar (

```
citizenId INT NOT NULL ,  
passportId VARCHAR(20) NOT NULL,  
issueDate DATE NOT NULL);  
  
INSERT INTO issueAadhar (citizenId, passportId, issueDate)  
VALUES  
  
(1001,'null', '2020-04-24'),  
  
(1002, 'null', '2020-04-25');  
  
SHOW TABLES;  
  
select * from issueaadhar;
```

## //Frontend-code

```
npm install -g @angular/cli    //installing Angular
```

```
ng new aadhar-card-application  //creating new app
```

```
ng generate component name     //for creating new component
```

```
ng generate component registration
```

```
ng generate component login
```

```
ng generate component admin-login
```

```
ng generate component approve-request
```

```
ng generate component verify-duplicate-request
```

```
ng generate component display-issued-cards
```

```
ng generate component delete-deceased-citizen
```

```
ng generate component dashboard
```

```
ng generate component apply-for-aadhar
```

```
ng generate component view-aadhar-number
```

```
ng serve    // for running Angular project
```

```
// registration.component.html
```

```
<h2>Registration</h2>
```

```
<link rel="stylesheet" href="registration.component.css">
```

```
<div *ngIf="!registrationSuccess">
```

```
  <form (ngSubmit)="register()" #registrationForm="ngForm">
```

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

```
    <input type="email" id="email" name="email"  
    [(ngModel)]="userData.email" required>
```

```
<label for="password">Password:</label>

<input type="password" id="password" name="password"
[(ngModel)]="userData.password" required>

<button type="submit"
[disabled]="registrationForm.invalid">Register</button>

</form>

</div>

<div *ngIf="registrationSuccess">

  <p class="success-message">Registration successful! You will be
  redirected to the login page.</p>

</div>


```

```
// registration.component.ts
```

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

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



```
@Component({
  selector: 'app-registration',
  templateUrl: './registration.component.html',
  styleUrls: ['./registration.component.css']
})

export class RegistrationComponent {
  userData: any = {
    email: '',
    password: ''
  };

  registrationSuccess = false;

  constructor(private router: Router) {}

  register(): void {
    // Assuming you have a backend service to handle user
    registration.

    // Replace the following code with actual API call to register the
    user.

    // For this example, we'll simulate a successful registration after
    2 seconds.
```

```
setTimeout(() => {  
    this.registrationSuccess = true;  
  
    // Redirect to login page after successful registration  
    setTimeout(() => {  
        this.router.navigate(['/login']);  
    }, 2000);  
}, 2000);  
}  
}
```

```
// Login.component.html  
  
<h2>User Login</h2>  
  
<div *ngIf="!loginSuccess">  
    <form (ngSubmit)="login()" #loginForm="ngForm">  
        <label for="email">Email:</label>  
        <input type="email" id="email" name="email"  
            [(ngModel)]="userData.email" required>  
  
        <label for="password">Password:</label>
```

```
<input type="password" id="password" name="password"
[(ngModel)]="userData.password" required>
```

```
<button type="submit"
[disabled]="loginForm.invalid">Login</button>
```

```
</form>
```

```
</div>
```

```
<div *ngIf="loginSuccess">
```

```
<div>
```

```
<p class="success-message">User login successful! You are now
logged in.</p>
```

```
<p><a routerLink="/apply-for-aadhar">Click here</a> to apply
for a new Aadhar card.</p>
```

```
<p><a routerLink="/view-aadhar-number">Click here</a> to
view your assigned Aadhar number.</p>
```

```
</div>
```

```
</div>
```

```

```

```
//Login.component.ts

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

@Component({
  selector: 'app-login',
  templateUrl: './login.component.html',
  styleUrls: ['./login.component.css']
})
export class LoginComponent {
  userData: any = {
    email: '',
    password: ''
  };
  loginSuccess: boolean = false;

  constructor(private router: Router) {}

  login(): void {
    // Assuming you have a login service that validates user
    credentials.
  }
}
```

```
// Check user credentials and set loginSuccess to true upon  
successful login.
```

```
// For this example, let's assume successful login always.
```

```
this.loginSuccess = true;
```

```
// Redirect to the dashboard after successful login.
```

```
this.router.navigateByUrl('/dashboard');
```

```
}
```

```
}
```

```
//admin-login.component.html
```

```
<h2>Admin Login</h2>
```

```
<link rel="stylesheet" href="admin-login.component.css">
```

```
<div *ngIf="!loginSuccess">
```

```
  <form (ngSubmit)="login()" #loginForm="ngForm">
```

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

```
    <input type="email" id="email" name="email"  
    [(ngModel)]="adminData.email" required>
```

```
    <label for="password">Password:</label>
```

```
<input type="password" id="password" name="password"
[(ngModel)]="adminData.password" required>
```

```
<button type="submit"
[disabled]="loginForm.invalid">Login</button>
```

```
</form>
```

```
</div>
```

```
<div *ngIf="loginSuccess">
```

```
<p class="success-message">Admin login successful! You are now
logged in as an admin.</p>
```

```
<ul>
```

```
<li><a routerLink="/approve-aadhar-request">Click here </a>for
Approve new Aadhaar Card request</li>
```

```
<li><a routerLink="/verify-duplicate-aadhar">Click here </a>for
Verify request for duplicate Aadhaar</li>
```

```
<li><a routerLink="/display-issued-aadhar">Click here</a> for
Display all issued Aadhaar Card</li>
```

```
<li><a routerLink="/delete-aadhar-details">Click here</a> for
Delete Aadhaar card details for dead citizen</li>
```

```
</ul>
```

```
</div>
```

```

```

```
//admin-login.component.ts
```

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

```
@Component({  
  selector: 'app-admin-login',  
  templateUrl: './admin-login.component.html',  
  styleUrls: ['./admin-login.component.css']  
})
```

```
export class AdminLoginComponent {
```

```
  adminData: any = {
```

```
    email: '',
```

```
    password: ''
```

```
};
```

```
loginSuccess = false;
```

```
login(): void {
```

**// Assuming you have a backend service to handle admin authentication.**

**// Replace the following code with actual API call to authenticate the admin.**

**// For this example, we'll simulate a successful admin login after 2 seconds.**

```
setTimeout(() => {  
  this.loginSuccess = true;  
}, 2000);  
}  

```

**//app.component.html**

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

```
@Component({  
  selector: 'app-root',  
  templateUrl: './app.component.html',  
  styleUrls: ['./app.component.css']  
})
```



```
  })  
  
  export class AppComponent {  
    title = 'aadhar-card-application';  
  }
```

**//app.component.ts**

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

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

**//app.module.ts**

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

```
import { BrowserModule } from '@angular/platform-browser';  
  
import { FormsModule } from '@angular/forms'; // Import  
FormsModule
```

```
import { AppRoutingModuleModule } from './app-routing.module';
```

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

```
import { RegistrationComponent } from  
'./registration/registration.component';
```

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

```
import { AdminLoginComponent } from './admin-login/admin-  
login.component';
```

```
import { ApproveRequestComponent } from './approve-  
request/approve-request.component';
```

```
import { VerifyDuplicateRequestComponent } from './verify-  
duplicate-request/verify-duplicate-request.component';
```

```
import { DisplayIssuedCardsComponent } from './display-issued-  
cards/display-issued-cards.component';
```

```
import { DeleteDeceasedCitizenComponent } from './delete-  
deceased-citizen/delete-deceased-citizen.component';
```

```
import { ApplyForAadharComponent } from './apply-for-  
aadhar/apply-for-aadhar.component';
```

```
import { ViewAadharNumberComponent } from './view-aadhar-  
number/view-aadhar-number.component';
```

```
import { DashboardComponent } from  
'./dashboard/dashboard.component';
```

```
@NgModule({  
  declarations: [  
    AppComponent,  
    RegistrationComponent,  
    LoginComponent,  
    AdminLoginComponent,  
    ApproveRequestComponent,  
    VerifyDuplicateRequestComponent,  
    DisplayIssuedCardsComponent,  
    DeleteDeceasedCitizenComponent,  
    ApplyForAadharComponent,  
    ViewAadharNumberComponent,  
    DashboardComponent  
  ],  
  imports: [  
    BrowserModule,  
    AppRoutingModule,  
    FormsModule // Add FormsModule to the imports array  
  ],
```

```
providers: [],  
  
bootstrap: [AppComponent]  
}))  
  
export class AppModule { }
```

**//app-routing.module.ts**

```
import { NgModule } from '@angular/core';  
  
import { RouterModule, Routes } from '@angular/router';  
  
import { RegistrationComponent } from  
'./registration/registration.component';  
  
import { LoginComponent } from './login/login.component';  
  
import { AdminLoginComponent } from './admin-login/admin-  
login.component';  
  
import { ApproveRequestComponent } from './approve-  
request/approve-request.component';  
  
import { VerifyDuplicateRequestComponent } from './verify-  
duplicate-request/verify-duplicate-request.component';  
  
import { DisplayIssuedCardsComponent } from './display-issued-  
cards/display-issued-cards.component';  
  
import { DeleteDeceasedCitizenComponent } from './delete-  
deceased-citizen/delete-deceased-citizen.component';  
  
import { DashboardComponent } from  
'./dashboard/dashboard.component';
```

```
import { ApplyForAadharComponent } from './apply-for-aadhar/apply-for-aadhar.component';
```

```
import { ViewAadharNumberComponent } from './view-aadhar-number/view-aadhar-number.component';
```

```
const routes: Routes = [
```

```
  { path: '', redirectTo: '/register', pathMatch: 'full' }, // Redirect to /register when accessing the home page
```

```
  { path: 'register', component: RegistrationComponent },
```

```
  { path: 'login', component: LoginComponent },
```

```
  { path: 'admin-login', component: AdminLoginComponent },
```

```
  { path: 'approve-request', component: ApproveRequestComponent },
```

```
  { path: 'verify-duplicate-request', component: VerifyDuplicateRequestComponent },
```

```
  { path: 'display-issued-cards', component: DisplayIssuedCardsComponent },
```

```
  { path: 'delete-deceased-citizen', component: DeleteDeceasedCitizenComponent },
```

```
  {
```

```
    path: 'dashboard',
```

```
    component: DashboardComponent,
```

```
    children: [
```

```
    { path: 'apply-for-aadhar', component:
ApplyForAadharComponent },

    { path: 'view-aadhar-number', component:
ViewAadharNumberComponent },

],

},

];
```

```
@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```