

M-AADHAR APPLICATION

SOURCE CODE :

BACKEND:

BackendApplicatoin.java

```
package com;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.boot.autoconfigure.domain.EntityScan;

@SpringBootApplication(scanBasePackages = "com.m_aadhar")
@EntityScan(basePackages = "com.m_aadhar.bean")
public class BackendApplication {

    public static void main(String[] args) {

        SpringApplication.run(BackendApplication.class, args);
        System.out.println("server running on port number 8090.....");
    }

}
```

Login.java :

```
package com.m_aadhar.bean;

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

@Entity

```
public class 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 + "]\n";
    }
}

```

LoginController.java :

```
package com.m_aadhar.controller;
```

```
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.MediaType;
import org.springframework.web.bind.annotation.CrossOrigin;
import org.springframework.web.bind.annotation.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.m_aadhar.bean.Login;
import com.m_aadhar.service.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.m_aadhar.controller;

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.m_aadhar.bean.Login;
import com.m_aadhar.service.LoginService;

```

@RestController

@CrossOrigin("*")

public class OperationController {

 @Autowired

 LoginService ls;

 @PostMapping("/apply")

 public String apply(@RequestBody Login login) {

 return ls.storeUser(login);

 }

 @PatchMapping("/update")

 public String update(@RequestBody Login login) {

 return ls.update(login);

 }

 @GetMapping("/getAll")

 public List<Login> getAllUser(){

 return ls.getAll();

 }

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

 public List<Login> getOneUser(@PathVariable("id") int id){

 return ls.findByid(id);

 }

 @DeleteMapping("delete/{emailid}")

 public String deleteUser(@PathVariable("emailid") String emailid){

 return ls.delete(emailid);

```
}  
  
}
```

LoginRepository.java :

```
package com.m_aadhar.repository;  
  
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.m_aadhar.bean.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);  
  
}
```

LoginService.java :

```
package com.m_aadhar.service;  
  
import java.util.List;  
import java.util.Optional;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
import com.m_aadhar.bean.Login;  
import com.m_aadhar.repository.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 {
        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";
    }
}

```

Application.properties :

```
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.url=jdbc:mysql://localhost:3306/db1
spring.datasource.username=root
spring.datasource.password=Kumar@123
spring.jpa.hibernate.ddl-auto=update
spring.jpa.properties.hibernate.show_sql=true
spring.jpa.properties.hibernate.format_sql=true
server.port = 8090

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
https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.6.12</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>com.example</groupId>
  <artifactId>Backend</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>Backend</name>
  <description>Demo project for Spring Boot</description>
  <properties>
    <java.version>1.8</java.version>
  </properties>
  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-data-jpa</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
```

```

        <artifactId>spring-boot-devtools</artifactId>
        <scope>runtime</scope>
        <optional>true</optional>
    </dependency>
    <dependency>
        <groupId>mysql</groupId>
        <artifactId>mysql-connector-java</artifactId>
        <scope>runtime</scope>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
    </dependency>
    <dependency>
<groupId>javax.persistence</groupId>
<artifactId>javax.persistence-api</artifactId>
</dependency>
</dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>
</project>

```