

mAadhar Application

Source Code

Spring Boot- Backend (Admin)

AdminController.java

```
package com.admin.controller;
import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;
import com.admin.model.CardDetails;
import com.admin.model.Credentials;

import com.admin.model.UserDetails;
import com.admin.model.UserRequests;
import com.admin.service.AdminServices;
@RestController
@RequestMapping("/admin")

public class AdminController {

    @Autowired

    AdminServices service;

    @PostMapping("/signin")

    public boolean signin(@RequestBody Credentials cred) {

        Credentials c=service.read(cred.getuName());

        if(c.getuName().matches(cred.getuName()) &&
c.getPass().matches(cred.getPass())) {
            return true;}

        else {

            return false;}}

    @GetMapping("/allData")

    public List<CardDetails> viewAllData(){

        return service.getAllCards();}

    @GetMapping("/pending")

    public List<CardDetails> viewPendingCardReq(){

        return service.PendingReq();}
```

```

    @GetMapping("/approved")

    public List<CardDetails> viewApprovedCards(){

        return service.ApprovedReq(); }

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

    public boolean deleteCard(@PathVariable long id) {

        return service.deleteCard(id);    }

    @PostMapping("/add")

    public boolean addData(@RequestBody CardDetails card) {

        service.addCard(card);

        return true;}

    @PostMapping("/updateCard/{id}")

    public void updateState(@RequestBody CardDetails newData) {

        service.updateCardDetails(newData);}

    @PostMapping("/applyCard")

    public Long applyCard(@RequestBody UserDetails data) {

        CardDetails application = new CardDetails();

        application.setName(data.getName());

        application.setAddress(data.getAddress());

        application.setDob(data.getDob());

        application.setEmail(data.getEmail());

        application.setGender(data.getGender());

        application.setMobile(data.getMobileNo());

        application.setState("Pending");

        service.addCard(application);

        return service.find(data.getName());}

    @PostMapping("/submitRequest")

    public boolean submitRequest(@RequestBody UserDetails ud,String reqName)

{

    UserRequests req=new UserRequests();

    req.setName(ud.getName());

    req.setEmail(ud.getEmail());

```

```

req.setAddress(ud.getAddress());

req.setDob(ud.getDob());

req.setGender(ud.getGender());

req.setMobileNo(ud.getMobileNo());

req.setReqName(reqName);

req.setReqState("Pending");

req.setPass(ud.getPass());

return service.submitRequest(req);}

@GetMapping("/pendingReq")

public List<UserRequests> viewPending(){

return service.viewPending();

}}

```

CardDetails.java

```

package com.admin.model;
import java.sql.Date;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import org.hibernate.annotations.ColumnDefault;

@Entity
public class CardDetails {
@Id
@GeneratedValue(strategy = GenerationType.AUTO)

private long citizenId;
private String name;
private Date dob;
private String address;
private String email;
private long mobile;
private String gender;
@ColumnDefault("Pending")
private String state;
@Column(nullable = true)
private Date issueDate;
@Column(nullable = true)
private long aCardId;
public long getCitizenId() {
return citizenId;

}

public void setCitizenId(long citizenId) {

```

```
this.citizenId = citizenId;

}

public String getName() {
return name;
}

public void setName(String name) {
this.name = name;
}

public Date getDob() {
return dob;
}

public void setDob(Date dob) {
this.dob = dob;
}

public String getAddress() {
return address;
}

public void setAddress(String address) {
this.address = address;
}

public String getEmail() {
return email;
}

public void setEmail(String email) {
this.email = email;
}

public long getMobile() {
return mobile;
}

public void setMobile(long mobile) {
this.mobile = mobile;
}

public String getGender() {
return gender;
}

public void setGender(String gender) {
this.gender = gender;
}

public String getState() {
return state;
}

public void setState(String state) {
this.state = state;
}

public Date getIssueDate() {
return issueDate;
}

public void setIssueDate(Date issueDate) {
this.issueDate = issueDate;}

public long getaCardId() {
return aCardId;
}

public void setaCardId(long aCardId) {
this.aCardId = aCardId;
}
```

```

    public CardDetails(long citizenId, String name, Date dob, String address,
String email, long mobile, String gender,String state, Date issueDate, long
aCardId) {

    super();
    this.citizenId = citizenId;
    this.name = name;
    this.dob = dob;
    this.address = address;
    this.email = email;
    this.mobile = mobile;
    this.gender = gender;
    this.state = state;
    this.issueDate = issueDate;
    this.aCardId = aCardId;
    }

    public CardDetails() {
    super();
    }

    @Override

    public String toString() {
    return "CardDetails [citizenId=" + citizenId + ", name=" + name + ",
dob=" + dob + ", address=" + address + ", email=" + email + ", mobile=" +
mobile + ", gender=" + gender + ", state=" + state + ", issueDate="
+ issueDate + ", aCardId=" + aCardId + "];"

    }}

```

Credentials.java

```

package com.admin.model;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
@Entity

public class Credentials {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)

    private long aId;
    @Column
    private String uName;
    @Column
    private String pass;

    public String getuName() {
    return uName;
    }

    public void setuName(String uName) {
    this.uName = uName;}
}

```

```

    public String getPass() {
        return pass;
    }

    public void setPass(String pass) {
        this.pass = pass;
    }

    public Credentials(String uName, String pass) {
        super();
        this.uName = uName;
        this.pass = pass;
    }

    public Credentials() {
        super();
    }

    @Override

    public String toString() {
        return "Credentials [uName=" + uName + ", pass=" + pass + "]";
    }
}

```

UserDetails.java

```

package com.admin.model;

import java.sql.Date;
public class UserDetails {
    private String email;
    private String pass;
    private String name;
    private Date dob;
    private String address;
    private long mobileNo;
    private String gender;

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public String getPass() {
        return pass;
    }

    public void setPass(String pass) {
        this.pass = pass;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public Date getDob() {
        return dob;
    }

    public void setDob(Date dob) {
        this.dob = dob;
    }

    public String getAddress() {
        return address;
    }

    public void setAddress(String address) {
        this.address = address;
    }
}

```

```

        public long getMobileNo() {
            return mobileNo;}
        public void setMobileNo(long mobileNo) {
            this.mobileNo = mobileNo;}
        public String getGender() {
            return gender;}
        public void setGender(String gender) {
            this.gender = gender;}
        public UserDetails( String email, String pass, String name, Date dob,
String address, long mobileNo, String gender) {
            super();
            this.email = email;
            this.pass = pass;
            this.name = name;
            this.dob = dob;
this.address = address;
this.mobileNo = mobileNo;
this.gender = gender;
        }
        public UserDetails() {
            super();
        }
        @Override
        public String toString() {
            return "UserCredentials [ email=" + email + ", pass=" + pass + ", name="
+ name + ", dob=" + dob + ", address=" + address + ", mobileNo=" + mobileNo +
", gender=" + gender + " ]";
        }
    }
}

```

UserRequests.java

```

package com.admin.model;
import java.sql.Date;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

@Entity
public class UserRequests {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)

    private long reqId;
    private String reqName;
    private String reqState;
    private String email;
    private String pass;
    private String name;
    private Date dob;
    private String address;
    private long mobileNo;
    private String gender;
    public long getReqId() {
        return reqId;
    }
}

```

```
public void setReqId(long reqId) {
    this.reqId = reqId;
}
public String getReqName() {
    return reqName;
}
public void setReqName(String reqName) {
    this.reqName = reqName;
}
public String getReqState() {
    return reqState;
}
public void setReqState(String reqState) {
    this.reqState = reqState;
}
public String getEmail() {
    return email;
}
public void setEmail(String email) {
    this.email = email;
}
public String getPass() {
    return pass;
}
public void setPass(String pass) {
    this.pass = pass;
}
public String getName() {
    return name;
}
public void setName(String name) {
    this.name = name;
}
public Date getDob() {
    return dob;
}
public void setDob(Date dob) {
    this.dob = dob;
}
public String getAddress() {
    return address;
}

public void setAddress(String address) {
    this.address = address;
}
public long getMobileNo() {
    return mobileNo;
}
public void setMobileNo(long mobileNo) {
    this.mobileNo = mobileNo;
}
public String getGender() {
    return gender;
}
public void setGender(String gender) {
    this.gender = gender;
}
```



```

public UserRequests(long reqId, String reqName, String reqState, String email,
String pass, String name, Date dob, String address, long mobileNo, String
gender) {

    super();
    this.reqId = reqId;
    this.reqName = reqName;
    this.reqState = reqState;
    this.email = email;
    this.pass = pass;
    this.name = name;
    this.dob = dob;
    this.address = address;
    this.mobileNo = mobileNo;
    this.gender = gender;
}

    public UserRequests() {
        super();
    }
@Override
    public String toString() {
        return "UserRequests [reqId=" + reqId + ", reqName=" + reqName + ",
reqState=" + reqState + ", email=" + email + ", pass=" + pass + ", name=" +
name + ", dob=" + dob + ", address=" + address + ", mobileNo=" + mobileNo + ",
gender=" + gender + "]";
    }
}

```

AdminRepository.java

```

package com.admin.repository;

import org.springframework.data.repository.CrudRepository;
import org.springframework.stereotype.Repository;
import com.admin.model.Credentials;

@Repository
public interface AdminRepository extends CrudRepository<Credentials, Long> {
    Credentials findByUserName(String userName);
}

```

CardRepository.java

```

package com.admin.repository;
import java.util.List;
import org.springframework.data.repository.CrudRepository;
import com.admin.model.CardDetails;
public interface CardRepository extends CrudRepository<CardDetails, Long> {

    List<CardDetails> findByState(String string);

    CardDetails findByName(String name);

}

```

RequestRepository.java

```
package com.admin.repository;
import java.util.List;
import org.springframework.data.repository.CrudRepository;
import com.admin.model.UserRequests;
public interface RequestRepository extends CrudRepository<UserRequests, Long>
{
    List<UserRequests> findByReqState(String string);
}
```

AdminServices.java

```
package com.admin.service;

import java.util.List;
import java.util.Optional;
import org.hibernate.engine.internal.Nullability;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import com.admin.model.CardDetails;
import com.admin.model.Credentials;
import com.admin.model.UserRequests;
import com.admin.repository.AdminRepository;
import com.admin.repository.CardRepository;
import com.admin.repository.RequestRepository;
@Service
public class AdminServices {

    @Autowired
    AdminRepository
    Repo;

    @Autowired
    CardRepository
    cardRepo;

    @Autowired
    RequestRepository
    reqRepo;

    public void create(Credentials
        cred) {Repo.save(cred);
    }

    public List<Credentials> readAll(){
        return (List<Credentials>) Repo.findAll();
    }

    public Credentials read(String uName) {
        return Repo.findByName(uName);
    }

    public List<CardDetails> getAllCards(){
        return (List<CardDetails>) cardRepo.findAll();
    }
}
```

```

        public void updateCardDetails(CardDetails
            newCardData) {Optional<CardDetails>
oldData=cardRepo.findById(newCardData.getCitizenI
            d());cardRepo.save(newCardData);
        }

        public boolean deleteCard(long
            citizenId) {
            cardRepo.deleteById(citizenId);
            return true;
        }

        public boolean addCard(CardDetails
            data) {cardRepo.save(data);
            return true;
        }

        public List<CardDetails> PendingReq(){
            return cardRepo.findByState("Pending");
        }

        public List<CardDetails> ApprovedReq(){
            return cardRepo.findByState("Approved");
        }

        public long find(String name) {
            return cardRepo.findByName(name).getCitizenId();
        }

        public Iterable<UserRequests> allUserRequests() {
            return reqRepo.findAll();
        }

        public boolean submitRequest(UserRequests
            req) {reqRepo.save(req);
            return true;
        }

        public List<UserRequests> viewPending(){
            return reqRepo.findByReqState("Pending");
        }
    }

```

MAadharAdminApplication.java

```

package com.admin;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;
@SpringBootApplication
public class MAadharAdminApplication {

    public static void main(String[] args) {

        SpringApplication.run(MAadharAdminApplication.class, args);

    }
}

```

application.properties:

```
spring.jpa.database-platform=org.hibernate.dialect.MySQL5InnoDBDialect
spring.jpa.hibernate.ddl-auto=update
spring.datasource.driverClassName=com.mysql.cj.jdbc.Driver
spring.datasource.url=jdbc:mysql://localhost:3306/aadhar
spring.datasource.username=root
spring.datasource.password=root
```

pom.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.7.4</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>com.admin</groupId>
  <artifactId>mAadhar-admin</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>war</packaging>
  <name>mAadhar-admin</name>
  <description>mAadharApp's Admin service</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-jdbc</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-tomcat</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>

```

Spring Boot- Backend (User)

UserController.java:

```

package com.user.controller;
import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
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.RequestParam;
import org.springframework.web.bind.annotation.RestController;

import com.user.model.UserCredentials;
import com.user.services.UserServices;
@RestController
@RequestMapping("/AadharApp")
public class UserController {
    @Autowired UserServices services;
    @PostMapping("/login")
    public boolean login(@RequestParam String email, @RequestParam String pass) {
        return services.login(email, pass);
    }

    @PostMapping("/register")

    public boolean register(@RequestBody UserCredentials uCred) {
        return services.register(uCred);
    }

    @PostMapping("/applyCard")

    public long applyCard(@RequestBody UserCredentials userCred) {
        return services.applyAadhar(userCred);
    }

    @PostMapping("/sendRequest")

```

```

public boolean sendRequest(@RequestBody UserCredentials userCred, @RequestParam
String reqName) {
return services.sendRequest(userCred, reqName);
}
}

```

UserCredentials.java:

```

package com.user.model;
import java.sql.Date;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
@Entity
public class UserCredentials {
@Id
@GeneratedValue(strategy = GenerationType.AUTO)
private long uId;
private String email;
private String pass;
private String name;
private Date dob;
private String address;
private long mobileNo;
private String gender;
public long getuId() {
return uId;
}
public void setuId(long uId) {
this.uId = uId;
}
public String getEmail() {
return email;
}
public void setEmail(String email) {
this.email = email;
}
public String getPass() {
return pass;
}
public void setPass(String pass) {
this.pass = pass;
}
public String getName() {
return name;
}
public void setName(String name) {
this.name = name;
}
public Date getDob() {
return dob;
}
public void setDob(Date dob) {
this.dob = dob;
}
public String getAddress() {
return address;
}
public void setAddress(String address) {

```

```

    this.address = address;
}
public long getMobileNo() {
    return mobileNo;
}
public void setMobileNo(long mobileNo) {
    this.mobileNo = mobileNo;
}
public String getGender() {
    return gender;
}
public void setGender(String gender) {
    this.gender = gender;
}
public UserCredentials(long uId, String email, String pass, String name, Date
dob, String address, long mobileNo,
String gender) {
    super(); this.uId = uId;
    this.email = email; this.pass = pass; this.name = name; this.dob = dob;
    this.address = address; this.mobileNo = mobileNo; this.gender = gender;
}
public UserCredentials() {
    super();
}
@Override
public String toString() {
    return "UserCredentials [uId=" + uId + ", email=" + email + ", pass=" + pass +
        ", name=" + name + ", dob=" + dob
        + ", address=" + address + ", mobileNo=" + mobileNo + ", gender=" +
        gender + "]";
}
}

```

UserRepository.java:

```

package com.user.repository;

import org.springframework.data.repository.CrudRepository;
import org.springframework.stereotype.Repository;
import com.user.model.UserCredentials;
@Repository
public interface UserRepository extends CrudRepository<UserCredentials, Long>{
    UserCredentials findByEmail(String email);
}

```

UserServices.java:

```

package com.user.services;
import java.net.URL;
import java.util.Arrays;
import java.util.HashMap;
import java.util.Map;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpEntity;
import org.springframework.http.HttpHeaders;

```

```

import org.springframework.http.HttpMethod;
import org.springframework.http.MediaType;
import org.springframework.stereotype.Service;
import org.springframework.web.client.RestTemplate;
import com.user.model.UserCredentials;
import com.user.repository.UserRepository; @Service
public class UserServices {
    @Autowired UserRepository Repo;
    @Autowired
    RestTemplate restTemplate;
    public boolean login(String email, String pass) {
        UserCredentials uCred= Repo.findByEmail(email);
        if(uCred==null)
        {
            return false;
        }
        if(uCred.getEmail().matches(email)&&uCred.getPass().matches(pass))
        {

            return true;

        }
        else

            return false;
        }
    public boolean register(UserCredentials uCred) {
        Repo.save(uCred);
        return true;
    }
    public Iterable<UserCredentials> getAll(){
        return Repo.findAll();
    }

    public Long applyAadhar(UserCredentials userCred) {
        String url="http://localhost:8080/admin/applyCard";
        HttpHeaders headers = new HttpHeaders();
        headers.setAccept(Arrays.asList(MediaType.APPLICATION_JSON));
        HttpEntity<UserCredentials> entity = new
        HttpEntity<UserCredentials>(userCred,headers);

        return restTemplate.exchange(
            url, HttpMethod.POST, entity, Long.class).getBody();

    }

    public boolean sendRequest(UserCredentials userCred,String reqName) {
        String url="http://localhost:8080/admin/submitRequest";
        HttpHeaders headers = new HttpHeaders();
        headers.setAccept(Arrays.asList(MediaType.APPLICATION_JSON));
        HttpEntity<UserCredentials> entity = new
        HttpEntity<UserCredentials>(userCred,headers);
        Map<String, String> map = new HashMap<>();
        map.put("reqName", reqName);
        return restTemplate.exchange(url, HttpMethod.POST, entity, Boolean.class,
        map).getBody();
        //    return restTemplate.exchange(
        //        url, HttpMethod.POST, entity, Boolean.class).getBody();

    }

}

```


application.properties:

```
spring.jpa.database-platform=org.hibernate.dialect.MySQL5InnoDBDialect
spring.jpa.hibernate.ddl-auto=update
spring.datasource.driverClassName=com.mysql.cj.jdbc.Driver
spring.datasource.url=jdbc:mysql://localhost:3306/aadhar
spring.datasource.username=root
spring.datasource.password=root
server.port=6789
```

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/
4.0.0http://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.admin</groupId>
  <artifactId>mAadhar-user</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>war</packaging>
  <name>mAadhar-user</name>
  <description>mAadharApp's User service</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-jdbc</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-tomcat</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>
```

Angular- Frontend

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">

  <script src="https://kit.fontawesome.com/d8bfec6881.js"
  crossorigin="anonymous"></script>

</head>

<body>

  <app-root></app-root>

</body>

</html>
```

app.component.html

```
<nav class="navbar navbar-expand-lg bg-dark navbar-dark">

  <a class="navbar-brand " href="#">

  </a>

  <h3 class="text-white">Aadhar App</h3>

  <div class="collapse navbar-collapse" id="navbarSupportedContent">

    <ul class="navbar-nav" style="margin-left: 65vw">
```

```

<li
  class="nav-item"
  routerLinkActive="active"
  [routerLinkActiveOptions]="{ exact: true }"
>
  <a class="nav-link" routerLink="/">Home</a>
</li>
<li role="presentation" class="nav-item" routerLinkActive="active">
  <a class="nav-link" routerLink="signup">SignUp</a>
</li>
<li role="presentation" class="nav-item" routerLinkActive="active">
  <a class="nav-link" routerLink="adminsignin">Admin</a>
</li>
</ul>
</div>
</nav>
<router-outlet></router-outlet>

```

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_app';
}
```

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 { SigninComponent } from './signin/signin.component';
import { RegistrationComponent } from
 './registration/registration.component'; import {
DashboardComponent } from './dashboard/dashboard.component';
import { UpdateaadharComponent } from
 './updateaadhar/updateaadhar.component'; import {
AdminloginComponent } from './adminlogin/adminlogin.component';
import { AdmindashboardComponent } from
 './admindashboard/admindashboard.component';

const appRoutes: Routes = [
  {path: '', component:SigninComponent},
  {path: 'signup', component:RegistrationComponent},
  {path: 'dashboard', component:DashboardComponent},
  {path: 'updateaadhar', component:UpdateaadharComponent},
  {path: 'adminsignin', component:AdminloginComponent},
  {path: 'admindashboard', component:AdmindashboardComponent},
];
```

```
@NgModule({
  declarations: [
    AppComponent,
    SigninComponent,
    RegistrationComponent
    , DashboardComponent,
    UpdateaadharComponent,
    AdminloginComponent,
    AdmindashboardComponent
  ],
  imports: [
    BrowserModule
    , FormsModule,
    RouterModule.forRoot(appRoutes),
    ReactiveFormsModule,
    HttpClientModule,
  ],
  providers: [],
  bootstrap: [AppComponent]
})

export class AppModule { }
```

admindashboard.component.html

```
<div id="layoutSidenav">
  <div id="layoutSidenav_content">
    <main>
      <div class="container-fluid px-4">
        <div><p class="h2 mt-4 mb-4">Dashboard</p></div>
        <div class="row">
          <div class="col-xl-3 col-md-6">
            <div class="card bg-primary text-white mb-4">
              <div class="card-body">New Card Requests</div>
              <div class="card-footer d-flex align-items-center">
                <a class="small text-white stretched-link" href="#">View
Details</a><span class="badge badge-light ms-2">0</span>
                <div class="small text-white"><i class="fas fa-angle-
right"></i></div>
              </div>
            </div>
          </div>
          <div class="col-xl-3 col-md-6">
            <div class="card bg-warning text-white mb-4">
              <div class="card-body">Correction Requests</div>
              <div class="card-footer d-flex align-items-center">
                <a class="small text-white stretched-link" href="#">View
Details</a><span class="badge badge-danger ms-2">1</span>
                <div class="small text-white"><i class="fas fa-angle-
right"></i></div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </main>
  </div>
</div>
```

```

    </div>

</div>

<div class="col-xl-3 col-md-6">

    <div class="card bg-success text-white mb-4">

        <div class="card-body">Approved Requests</div>

        <div class="card-footer d-flex align-items-center">

            <a class="small text-white stretched-link" href="#">View
Details</a><span class="badge badge-danger ms-2">3</span>

            <div class="small text-white"><i class="fa fa-angle-
right"></i></div>

        </div>

    </div>

</div>

<div class="col-xl-3 col-md-6">

    <div class="card bg-danger text-white mb-4">

        <div class="card-body">Delete Requests</div>

        <div class="card-footer d-flex align-items-center">

            <a class="small text-white stretched-link" href="#">View
Details</a><span class="badge badge-light ms-2">0</span>

            <div class="small text-white"><i class="fas fa-angle-
right"></i></div>

        </div>

    </div>

</div>

</div>

<div class="card mb-4">

    <div class="card-header">

```



```
<i class="fas fa-table me-
1"></i> Issued citizen Data
</div>
<div class="card-body">
  <table id="datatablesSimple" class="table table-hover">
    <thead>
      <tr>
        <th>Name</th>
        <th>Gender</th>
        <th>Address</th>
        <th>email</th>
        <th>Contact</th>
        <th>Issue date</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>Gopalakrishnan</td>
        <td>Male</td>
        <td>Madurai</td>
        <td>gk@gmail.com</td>
        <td>9952658175</td>
        <td>13/10/2022</td>
      </tr>
      <tr>
```

```
        <td>krishna</td>

        <td>Male</td>

        <td>Melur</td>

        <td>gk@test.com</td>

        <td>8220338647</td>

        <td>01/08/2021</td>
    </tr>

    <tr>

        <td>Geetha</td>

        <td>Female</td>

        <td>Chennai</td>

        <td>test@test.com</td>

        <td>1234567890</td>

        <td>13/10/2022</td>
    </tr>
</tbody>
</table>
</div>
</div>
</div>
</main>
<footer class="py-4 bg-light mt-
    auto">
<div class="container-fluid px-4">

    <div class="d-flex align-items-center justify-content-between small">

        <div class="text-muted">Copyright &copy; Varniraj Service PVT. LTD
        2022</div>
```

```

        <div>

            <a href="#">Privacy
            Policy</a> &middot;

            <a href="#">Terms & amp; Conditions</a>

        </div>

    </div>

</div>

</footer>

</div>

</div>

```

admin-data.ts

```

export interface
    AdminData { aId:
        number;
        uName:
        string;
        pass:
        string;
    }

```

adminlogin.component.html

```

<div class="container">

    <!-- Form -->

    <div class="row">

        <div class="col-6">

            <form class="form-example card mt-5 pl-5 pt-5 pb-5" (ngSubmit)="onSubmit()">

                <h1>Admin Login</h1>

                <div class="form-group">

```

```
<label for="phone_num" class="text-light">Username:</label>
<input
  type="text"
  class="form-control"
  uName" id="uName"
  placeholder="Username"
  name="uName"
  style="width: 20vw"
/>
</div>
<div class="form-group">
  <label for="password" class="text-light">Password:</label>
  <input
    type="password"
    [(ngModel)]="admin.pass"
    class="form-control"
    password" id="pass"
    placeholder="Password"
    name="pass"
    style="width: 20vw"
  />
</div>
<div>
  <button type="submit" class="btn btn-primary btn-
    customized"> Login
```

```

        </button>

    </div>

    <!-- End input fields -->

</form>

</div>

<div class="col-6">

    

</div>

<!-- Form end -->

</div>

</div>

```

adminlogin.component.ts

```

import { Component, OnInit } from '@angular/core';
import { FormBuilder, FormControl, FormGroup, Validators } from
'@angular/forms'; import { Admin } from '../services/admin';
import { AdminServiceService } from '../services/admin-
service.service'; import { Router } from
'@angular/router';

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

```

```

}))

export class AdminloginComponent implements OnInit {

    signupForm: FormGroup;

    admin:Admin=new
    Admin();

    constructor(private adminServ:AdminServiceService, private router:
Router,private fb: FormBuilder) { }

    ngOnInit(): void {
        this.signupForm =
        this.fb.group({ uName:
        'admin',
        pass: 'Qwerty@098!'
        });
    }

    onSubmit() {

        if (this.signupForm.value.uName=="admin

        " &&
this.signupForm.value.pass=="Qwerty@098!") {
            this.router.navigate(['/admindashboard']);
        }
        else{
            alert("Invalid credentials");
        }
    }
}

```

dashboard.component.html

```
<div class="container">

<h2>Your Card</h2>

<div>

  <div class="card" style="width: 638px;" >

    <div class="card-header">

      <i class="fa-solid fa-id-
        card"></i> Preview

    </div>

    <div class="card-body"></div>

  </div>

  <button class="btn btn-primary ml-5"
    (click)="gotoUpdateAadhar()"> Update aadhar details

  </button>

</div>

</div>

<div id="layoutSidenav">

  <div id="layoutSidenav_content">

    <main>

      <div class="container-fluid px-4">

        <div><p class="h2 mt-4 mb-4">User Dashboard</p></div>

        <div class="row">

          <div class="col-xl-3 col-md-6">
```

```
<div class="card bg-primary text-white mb-4">

  <div class="card-body">New Card Requests</div>

  <div class="card-footer d-flex align-items-center">

    <a class="small text-white stretched-link" href="#">Apply</a>

    <div class="small text-white"><i class="fas fa-angle-
      right"></i></div>

  </div>

</div>

</div>

<div class="col-xl-3 col-md-6">

  <div class="card bg-warning text-white mb-4">

    <div class="card-body">Correction Request</div>

    <div class="card-footer d-flex align-items-center">

      <a class="small text-white stretched-link" href="#">Send</a>

      <div class="small text-white"><i class="fas fa-angle-
        right"></i></div>

    </div>

  </div>

</div>

<div class="col-xl-3 col-md-6">

  <div class="card bg-danger text-white mb-4">

    <div class="card-body">Delete Request</div>

    <div class="card-footer d-flex align-items-center">

      <a class="small text-white stretched-link" href="#">Send</a>

      <div class="small text-white"><i class="fas fa-angle-
        right"></i></div>

    </div>

  </div>

</div>
```



```

        </div>

    </div>

</div>
</main>

<footer class="py-4 bg-light mt-
        auto">

<div class="container-fluid px-4">

    <div class="d-flex align-items-center justify-content-between small">

        <div class="text-muted">Copyright &copy; Varniraj Service PVT. LTD
        2022</div>

        <div>

            <a href="#">Privacy
            Policy</a> &middot;

            <a href="#">Terms &amp; Conditions</a>

        </div>

    </div>

</div>

</div>

</div>
</div>

```

dashboard.component.ts

```

import { Component, OnInit } from
'@angular/core'; import { Router } from
'@angular/router'; @Component({
    selector: 'app-dashboard',
    templateUrl:
    './dashboard.component.html',
    styleUrls:
    ['./dashboard.component.css']

```

```

})

export class DashboardComponent implements
  OnInit { constructor(private router: Router)
  { }

  ngOnInit(): void {

  }

  gotoUpdateAadhar() {
    this.router.navigate(['/updateaadhar']);
  }
}

```

registration.component.html

```

<div class="container">

  <!-- Form -->

<div class="row">

<div class="col-6">

  <section class="h-100 bg-dark">

    <div class="container h-100">

      <div class="row d-flex justify-content-center align-items-center h-100">

        <div class="col">

          <div class="card card-registration">

            <div class="row g-0">

              <div class="">

                <div class="card-body p-md-5 text-black">

                  <h3 class="mb-5 text-uppercase">Citizen Registration form</h3>

                <div class="row">

```

```

<div class="col-md-6 mb-4">

  <div class="form-outline">

    <input type="text" id="form3Example1m" class="form-control form-
    control-lg"

/>

    <label class="form-label" for="form3Example1m">First name</label>

  </div>

</div>

<div class="col-md-6 mb-4">

  <div class="form-outline">

    <input type="text" id="form3Example1n" class="form-control form-
    control-lg"

/>

    <label class="form-label" for="form3Example1n">Last name</label>

  </div>

</div>

</div>

<div class="form-outline mb-4">

  <input type="text" id="form3Example8" class="form-control form-
  control-lg" />

  <label class="form-label" for="form3Example8">Address</label>

</div>

<div class="d-md-flex justify-content-start align-items-center mb-4
py-2">

  <h6 class="mb-0 me-4">Gender: </h6>

  <div class="form-check form-check-inline mb-0 me-4">

```

```
        <input class="form-check-input" type="radio"
name="inlineRadioOptions" id="femaleGender"

        value="option1" />

        <label class="form-check-label" for="femaleGender">Female</label>

</div>
```

```
<div class="form-check form-check-inline mb-0 me-4">

    <input class="form-check-input" type="radio"
name="inlineRadioOptions" id="maleGender"

    value="option2" />

    <label class="form-check-label" for="maleGender">Male</label>

</div>
```

```
<div class="form-check form-check-inline mb-0">

    <input class="form-check-input" type="radio"
name="inlineRadioOptions" id="otherGender"

    value="option3" />

    <label class="form-check-label" for="otherGender">Other</label>

</div>
```

```
</div>
```

```
<div class="row">

    <div class="col-md-6 mb-4">
```

```
        <select class="select">
```

```
<option value="1">State</option>
<option value="2">Option 1</option>
<option value="3">Option 2</option>
<option value="4">Option 3</option>
</select>
```

```
</div>
```

```
<div class="col-md-6 mb-4">
```

```
<select class="select">
  <option value="1">City</option>
  <option value="2">Option 1</option>
  <option value="3">Option 2</option>
  <option value="4">Option 3</option>
</select>
```

```
</div>
```

```
</div>
```

```
<div class="form-outline mb-4">
```

```
<input type="date" id="form3Example9" class="form-control form-
control-lg" />
```

```
<label class="form-label" for="form3Example9">DOB</label>
```

```
</div>
```

```
<div class="form-outline mb-4">
```

```

/>      <input type="number" id="form3Example90" class="form-control form-
control-lg"

      <label class="form-label" for="form3Example90">Contact No.</label>
</div>

<div class="form-outline mb-4">

  <input type="text" id="form3Example97" class="form-control form-
control-lg" />

  <label class="form-label" for="form3Example97">Email ID</label>
</div>

lg"    <div class="form-outline mb-4">
/>      <input type="password" id="form3Example97" class="form-control form-
control-

      <label class="form-label" for="password">Password</label>
</div>

<div class="d-flex justify-content-end pt-3">

  <button type="button" class="btn btn-light btn-lg">Reset all</button>

  <button type="button" class="btn btn-primary btn-lg ms-
2">Register</button>

</div>

</div>
</div>
</div>
</div>
</div>

```

```

    </div>

</div>

</section>

</div>

<div class="col-6">

    

    </div>

<!-- Form end -->

</div>

</div>

```

registration.component.ts

```

import { Component, OnInit } from '@angular/core';
import { FormControl, FormGroup, Validators } from
'@angular/forms'; import { Router } from
'@angular/router';
import { kMaxLength } from
'buffer'; @Component({
    selector: 'app-registration',
    templateUrl:
    './registration.component.html',
    styleUrls:
    ['./registration.component.css']
})
export class RegistrationComponent implements
    OnInit { signupForm: FormGroup;
    constructor(private router:
    Router) { } ngOnInit(): void {

```

```

    this.signupForm = new FormGroup({
      phone_num: new FormControl(null,
        [Validators.required, Validators.pattern("^(\\+91-
?)|0)?[0-9]{10}$")]),
      password: new FormControl(null, [Validators.required,]),
    });
  }

  onSubmit() {
    console.log(this.signupForm.valu
e); this.router.navigate(['/']);
  }
}

```

signin.component.html

```

<div class="container">
  <!-- Form -->
  <div class="row">
    <div class="col-6">
      <form class="form-example card mt-5 pl-5 pt-5 pb-5" [formGroup]="signupForm"
(ngSubmit)="onSubmit()">
        <h1>Login</h1>

        <div class="form-group">
          <label for="phone_num" class="text-light">MobileNumber:</label>
          <input
            type="tex
            t"
            class="form-control
            phone_num" id="phone_num"

```



```
        placeholder="Mobile
        Number" name="phone_num"
        style="width: 20vw"
    />
</div>
<div class="form-group">
    <label for="password" class="text-light">Password:</label>
    <input
        type="password"
        class="form-control
        password" id="password"
        placeholder="Password"
        name="password"
        style="width: 20vw"
    />
</div>
<div>
    <button type="submit" class="btn btn-primary btn-
        customized"> Login
    </button>
</div>
<!-- End input fields -->
</form>
</div>
<div class="col-6">
```

```

    

  </div>

  <!-- Form end -->

</div>

</div>

```

signin.component.ts

```

import { Component, OnInit } from '@angular/core';
import { FormControl, FormGroup, Validators } from
 '@angular/forms'; import { Router } from
 '@angular/router';

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

export class SigninComponent implements
  OnInit { signupForm: FormGroup;

  constructor(private router:
  Router) { } ngOnInit(): void {

    this.signupForm = new FormGroup({

      phone_num: new FormControl(null,
        [Validators.required,Validators.pattern("^(\\+91-
?)|0)?[0-9]{10}$")),

      password: new FormControl(null, [Validators.required]),

    });
  }
}

```

```
onSubmit() {  
  console.log(this.signupForm.valu  
e);  
  this.router.navigate(['/dashboar  
d']);  
}}
```

updateaadhar.component.html

```
<div>  
  <!-- Form -->  
  
  <form class="form-example">  
    <h2>Update aadhar page</h2>  
  
    <div class="form-group">  
      <label for="address" class="text-light">Address:</label>  
      <input  
        type="tex  
t"  
        class="form-control  
address" id="address"  
placeholder="City"  
name="address"  
style="width: 20vw"  
      />  
    </div>  
  
    <div class="form-group">  
      <label for="phonenum" class="text-light">MobileNumber:</label>  
      <input
```

```
        type="text"
        class="form-
        control"
        id="phonenum"
        placeholder="MobileNumbe
        r" name="phonenum"
        style="width: 20vw"
    />
</div>
<div class="form-group">
    <label for="dob" class="text-light">Date of Birth:</label>
    <input
        type="tex
        t"
        class="form-control"
        id="dob"
        placeholder="Date of
        Birth" name="dob"
        style="width: 20vw"
    />
</div>
<div>
    <button type="submit" class="btn btn-primary btn-
        customized"> Send Request
    </button>
</div>
<!-- End input fields -->
```

```
</form>

<!-- Form end -->

</div>
```

admin-service.service.ts

```
import { Injectable } from '@angular/core';
import { HttpClient } from
'@angular/common/http' import { Observable
} from 'rxjs';
import { Admin } from
'./admin'; @Injectable({
  providedIn: 'root'
})
export class AdminServiceService {
  private baseUrl =
"http://localhost:8080/admin/signin";
  constructor(private httpClient: HttpClient) {
  }
  signin(cred: Admin): Observable<Object>{
    return this.httpClient.post(`${this.baseUrl}`, cred);
  }
}
```

admin.ts

```
export class
  Admin { aId:
    number; uName:
    string; pass:
    string;
}
```

Lists of URL

UserAPI URLs:

<http://localhost:6789/AadharApp/login>
<http://localhost:6789/AadharApp/register>
<http://localhost:6789/AadharApp/applyCard>
<http://localhost:6789/AadharApp/sendRequest>

AdminAPI URLs:

<http://localhost:8080/admin/signin>
<http://localhost:8080/admin/allData>
<http://localhost:8080/admin/pending>
<http://localhost:8080/admin/approved>
<http://localhost:8080/admin/delete/{id}>
<http://localhost:8080/admin/add>
<http://localhost:8080/admin/updateCard/{id}>
<http://localhost:8080/admin/applyCard>
<http://localhost:8080/admin/submitRequest>
<http://localhost:8080/admin/pendingReq>

