**CONTROLLER:**

package com.example.demo.controller;

import java.util.List;

import java.util.Optional;

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

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.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.model.User;

import com.example.demo.service.ApiService;

@CrossOrigin(origins="http://localhost:3000/")

@RestController

@RequestMapping("/user")

public class ApiController

{

@Autowired

ApiService service;

@PostMapping("")

public boolean create(@RequestBody User user)

{

return service.addUser(user);

}

@GetMapping("")

public List<User>read()

{

return service.getUser();

}

@GetMapping("/{emailid}")

public Optional<User>readById(@PathVariable String emailid)

{

return service.getUserById(emailid);

}

@PutMapping("/{emailid}")

public User update(@RequestBody User user)

{

return service.updateUser(user);

}

@DeleteMapping("/{emailid}")

public boolean delete(@PathVariable String emailid)

{

return service.deleteUserById(emailid);

}

}

**MODEL:**

package com.example.demo.model;

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

@Entity

public class User

{

@Id

@GeneratedValue(strategy=GenerationType.AUTO)

private String userid;

private String firstname;

private String lastname;

private String emailid;

private String password;

private String confirmpassword;

public String getUserid() {

return userid;

}

public void setUserid(String userid) {

this.userid = userid;

}

public String getFirstname() {

return firstname;

}

public void setFirstname(String firstname) {

this.firstname = firstname;

}

public String getLastname() {

return lastname;

}

public void setLastname(String lastname) {

this.lastname = lastname;

}

public String getEmailid() {

return emailid;

}

public void setEmailid(String emailid) {

this.emailid = emailid;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

public String getConfirmpassword() {

return confirmpassword;

}

public void setConfirmpassword(String confirmpassword) {

this.confirmpassword = confirmpassword;

}

@Override

public String toString() {

return "User [userid=" + userid + ", firstname=" + firstname + ", lastname=" + lastname + ", emailid=" + emailid

+ ", password=" + password + ", confirmpassword=" + confirmpassword + "]";

}

}

**REPOSITORY:**

package com.example.demo.repository;

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

import org.springframework.stereotype.Repository;

import com.example.demo.model.User;

@Repository

public interface UserRepo extends JpaRepository<User, String>

{

}

**SERVICE:**

package com.example.demo.service;

import java.util.List;

import java.util.Optional;

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

import org.springframework.stereotype.Service;

import com.example.demo.model.User;

import com.example.demo.repository.UserRepo;

@Service

public class ApiService

{

@Autowired

UserRepo repository;

public boolean addUser(User user)

{

repository.save(user);

return true;

}

public List<User>getUser()

{

return repository.findAll();

}

public Optional<User>getUserById(String emailid)

{

return repository.findById(emailid);

}

public User updateUser(User user)

{

repository.save(user);

return user;

}

public boolean deleteUserById(String emailid)

{

repository.deleteById(emailid);

return true;

}

}

**APPLICATION PROPERTIES:**

spring.datasource.url=jdbc:mysql://localhost:3306/userdb

spring.datasource.username=root

spring.datasource.password=\*\*\*\*

spring.datasource.dbcp2.driver-class-name=com.mysql.cj.jdbc.Driver

spring.jpa.database-platform=org.hibernate.dialect.MySQLDialect

spring.jpa.generate-ddl=true

spring.jpa.hibernate.ddl-auto=update