**cORE\_JAVA**

Admin.java

package org.sporty\_shoes.model;

import java.io.Serializable;

import javax.persistence.Entity;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity

@Table(name = "admin")

public class Admin implements Serializable {

    private static final long serialVersionUID = 1L;

    @Id

    private String email;

    private String password;

    public Admin() {

        super();

    }

    public Admin(String email, String password) {

        super();

        this.email = email;

        this.password = password;

    }

    public String getEmail() {

        return email;

    }

    public void setEmail(String email) {

        this.email = email;

    }

    public String getPassword() {

        return password;

    }

    public void setPassword(String password) {

        this.password = password;

    }

    @Override

    public String toString() {

        return String.format("Admin [email=%s, password=%s]", email, password);

    }

}

MyController.java

package org.sporty\_shoes.controller;

import org.sporty\_shoes.model.Admin;

import org.sporty\_shoes.repository.AdminRepository;

import org.sporty\_shoes.repository.ProductRepository;

import org.sporty\_shoes.repository.PurchaseReportRepository;

import org.sporty\_shoes.repository.Repository;

import org.sporty\_shoes.repository.UserRepository;

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

import org.springframework.stereotype.Controller;

import org.springframework.ui.ModelMap;

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

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.bind.annotation.RequestParam;

@Controller

public class MyController {

    @Autowired

    AdminRepository adminRepository;

    @Autowired

    ProductRepository productRepository;

    @Autowired

    UserRepository  userRepository;

    @Autowired

    PurchaseReportRepository purchaseReportRepository;

    @Autowired

    Repository repository;

    @RequestMapping("/")

    public String showHome() {

        return "Home";

    }

    @RequestMapping(value="adminLogin",method =RequestMethod.GET)

    public String adminLoginPage() {

        return "AdminLogin";

    }

    @RequestMapping(value="forgotPassword",method =RequestMethod.GET)

    public String forgotPassword() {

        return "ForgotPassword";

    }

    @RequestMapping(value="adminForgotPassword",method =RequestMethod.POST)

    public String adminForgotPasswordPage(@RequestParam("email")String email,

            @RequestParam("password")String password,ModelMap map) {

        if(repository.forgotPassword(email,password))

            map.addAttribute("message","Password Updated");

        else

            map.addAttribute("message","Invalid Details");

        return "ForgotPassword";

    }

    @RequestMapping(value="adminPage",method=RequestMethod.POST)

    public String adminPage(@RequestParam(name="email")String email,

            @RequestParam(name="password")String password,ModelMap map) {

        if(adminRepository.verifyAdmin(new Admin(email,password)))

            return "AdminPage";

        else {

            map.addAttribute("message", "Invalid Details");

            return "AdminLogin";

        }

    }

    @RequestMapping(value="product",method=RequestMethod.GET)

    public String getAllProducts(ModelMap map) {

        map.addAttribute("productList",productRepository.getAllProducts());

        return "Product";

    }

    @RequestMapping(value="user",method=RequestMethod.GET)

    public String getAllUsers(ModelMap map) {

        map.addAttribute("userList",userRepository.getAllUsers());

        return "User";

    }

    @RequestMapping(value="searchUser",method=RequestMethod.GET)

    public String searchUser(@RequestParam("email")String email,ModelMap map) {

        map.addAttribute("userList",userRepository.searchUser(email));

        return "User";

    }

    @RequestMapping(value="purchaseReport",method=RequestMethod.GET)

    public String getRport(ModelMap map) {

        map.addAttribute("report",purchaseReportRepository.getReport());

        return "PurchaseReport";

    }

    @RequestMapping(value="deleteProduct",method=RequestMethod.GET)

    public String deleteProduct(@RequestParam("id")int id,ModelMap map) {

        if(productRepository.deleteProduct(id))

            map.addAttribute("message","Product Deleted Successfully");

        else

            map.addAttribute("message","Try after few minutes");

        map.addAttribute("productList",productRepository.getAllProducts());

        return "Product";

    }

}

Product.java

package org.sporty\_shoes.model;

import java.io.Serializable;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity

@Table(name="product")

public class Product implements Serializable{

    private static final long serialVersionUID = 1L;

    @Id

    @GeneratedValue(strategy=GenerationType.AUTO)

    private int productId;

    private String productName;

    private int productPrice;

    private String productDiscription;

    public Product() {

        super();

    }

    public Product(String productName, int productPrice, String productDiscription) {

        super();

        this.productName = productName;

        this.productPrice = productPrice;

        this.productDiscription = productDiscription;

    }

    public int getProductId() {

        return productId;

    }

    public void setProductId(int productId) {

        this.productId = productId;

    }

    public String getProductName() {

        return productName;

    }

    public void setProductName(String productName) {

        this.productName = productName;

    }

    public int getProductPrice() {

        return productPrice;

    }

    public void setProductPrice(int productPrice) {

        this.productPrice = productPrice;

    }

    public String getProductDiscription() {

        return productDiscription;

    }

    public void setProductDiscription(String productDiscription) {

        this.productDiscription = productDiscription;

    }

    @Override

    public String toString() {

        return String.format("Product [productId=%s, productName=%s, productprice=%s, productDiscription=%s]",

                productId, productName, productPrice, productDiscription);

    }

}

PurchaseReport.java

package org.sporty\_shoes.model;

import java.io.Serializable;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity

@Table(name="purchase\_report")

public class PurchaseReport implements Serializable {

    private static final long serialVersionUID = 1L;

    @Id

    @GeneratedValue(strategy=GenerationType.AUTO)

    private int reportId;

    private String reportProductName;

    private String reportUserEmail;

    private String reportDate;

    private int reportPrice;

    public PurchaseReport() {

        super();

    }

    public PurchaseReport(String reportProductName, String reportUserEmail, String reportDate, int reportPrice) {

        super();

        this.reportProductName = reportProductName;

        this.reportUserEmail = reportUserEmail;

        this.reportDate = reportDate;

        this.reportPrice = reportPrice;

    }

    public int getReportId() {

        return reportId;

    }

    public void setReportId(int reportId) {

        this.reportId = reportId;

    }

    public String getReportProductName() {

        return reportProductName;

    }

    public void setReportProductName(String reportProductName) {

        this.reportProductName = reportProductName;

    }

    public String getReportUserEmail() {

        return reportUserEmail;

    }

    public void setReportUserEmail(String reportUserEmail) {

        this.reportUserEmail = reportUserEmail;

    }

    public String getReportDate() {

        return reportDate;

    }

    public void setReportDate(String reportDate) {

        this.reportDate = reportDate;

    }

    public int getReportPrice() {

        return reportPrice;

    }

    public void setReportPrice(int reportPrice) {

        this.reportPrice = reportPrice;

    }

    public static long getSerialversionuid() {

        return serialVersionUID;

    }

    @Override

    public String toString() {

        return String.format(

                "PurchaseReport [reportId=%s, reportProductName=%s, reportUserEmail=%s, reportDate=%s, reportPrice=%s]",

                reportId, reportProductName, reportUserEmail, reportDate, reportPrice);

    }

}

User.java

package org.sporty\_shoes.model;

import java.io.Serializable;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity

@Table(name="user")

public class User implements Serializable {

    private static final long serialVersionUID = 1L;

    @Id

    @GeneratedValue(strategy=GenerationType.AUTO)

    private int userId;

    private String userName;

    private String userEmail;

    private int userAge;

    private String userGender;

    private String userAddress;

    public User() {

        super();

    }

    public User(String userName, String userEmail, int userAge, String userGender, String userAddress) {

        super();

        this.userName = userName;

        this.userEmail = userEmail;

        this.userAge = userAge;

        this.userGender = userGender;

        this.userAddress = userAddress;

    }

    public int getUserId() {

        return userId;

    }

    public void setUserId(int userId) {

        this.userId = userId;

    }

    public String getUserName() {

        return userName;

    }

    public void setUserName(String userName) {

        this.userName = userName;

    }

    public String getUserEmail() {

        return userEmail;

    }

    public void setUserEmail(String userEmail) {

        this.userEmail = userEmail;

    }

    public int getUserAge() {

        return userAge;

    }

    public void setUserAge(int userAge) {

        this.userAge = userAge;

    }

    public String getUserGender() {

        return userGender;

    }

    public void setUserGender(String userGender) {

        this.userGender = userGender;

    }

    public String getUserAddress() {

        return userAddress;

    }

    public void setUserAddress(String userAddress) {

        this.userAddress = userAddress;

    }

    public static long getSerialversionuid() {

        return serialVersionUID;

    }

    @Override

    public String toString() {

        return String.format("User [userId=%s, userName=%s, userEmail=%s, userAge=%s, userGender=%s, userAddress=%s]",

                userId, userName, userEmail, userAge, userGender, userAddress);

    }

}

AdminRepository.java

package org.sporty\_shoes.repository;

import java.util.List;

import org.hibernate.Query;

import org.hibernate.Session;

import org.hibernate.Transaction;

import org.sporty\_shoes.model.Admin;

import org.springframework.orm.hibernate5.HibernateTemplate;

@SuppressWarnings("deprecation")

public class AdminRepository {

    private HibernateTemplate hibernateTemplate;

    public HibernateTemplate getHibernateTemplate() {

        return hibernateTemplate;

    }

    public void setHibernateTemplate(HibernateTemplate hibernateTemplate) {

        this.hibernateTemplate = hibernateTemplate;

    }

    public boolean verifyAdmin(Admin a) {

        Session session=hibernateTemplate.getSessionFactory().openSession();

        Transaction transaction=session.beginTransaction();

        String hql = "from Admin where email=:email and password=:password";

        Query<Admin> query = session.createQuery(hql,Admin.class);

        query.setParameter("email", a.getEmail());

        query.setParameter("password", a.getPassword());

        List<Admin> results = query.getResultList();

        transaction.commit();

        session.close();

        return results.size()>0;

    }

}

ProductRepository.java

package org.sporty\_shoes.repository;

import java.util.List;

import org.hibernate.Session;

import org.hibernate.Transaction;

import org.sporty\_shoes.model.Product;

import org.springframework.orm.hibernate5.HibernateTemplate;

public class ProductRepository {

    HibernateTemplate hibernateTemplate;

    public ProductRepository() {

        super();

    }

    public ProductRepository(HibernateTemplate hibernateTemplate) {

        super();

        this.hibernateTemplate = hibernateTemplate;

    }

    public HibernateTemplate getHibernateTemplate() {

        return hibernateTemplate;

    }

    public void setHibernateTemplate(HibernateTemplate hibernateTemplate) {

        this.hibernateTemplate = hibernateTemplate;

    }

    public List<Product> getAllProducts(){

        List<Product> productList=hibernateTemplate.loadAll(Product.class);

        return productList;

    }

    public Boolean deleteProduct(int id) {

        try {

        Session session=hibernateTemplate.getSessionFactory().openSession();

        Transaction transaction=session.beginTransaction();

        Product p=session.get(Product.class, id);

        session.delete(p);

        transaction.commit();

        session.close();

        }

        catch(Exception e) {

            return false;

        }

        return true;

    }

}

PurchaseReportRepository.java

package org.sporty\_shoes.repository;

import java.util.List;

import org.sporty\_shoes.model.PurchaseReport;

import org.springframework.orm.hibernate5.HibernateTemplate;

public class PurchaseReportRepository {

    HibernateTemplate hibernateTemplate;

    public HibernateTemplate getHibernateTemplate() {

        return hibernateTemplate;

    }

    public void setHibernateTemplate(HibernateTemplate hibernateTemplate) {

        this.hibernateTemplate = hibernateTemplate;

    }

    @SuppressWarnings("deprecation")

    public List<PurchaseReport> getReport(){

        @SuppressWarnings("unchecked")

        List<PurchaseReport> report=(List<PurchaseReport>) hibernateTemplate.find("from PurchaseReport order by reportProductName,reportDate");

        return report;

    }

}

Repository.java

package org.sporty\_shoes.repository;

import org.hibernate.Query;

import org.hibernate.Session;

import org.hibernate.Transaction;

import org.sporty\_shoes.model.Admin;

import org.springframework.orm.hibernate5.HibernateTemplate;

@SuppressWarnings("deprecation")

public class Repository {

    private HibernateTemplate hibernateTemplate;

    public HibernateTemplate getHibernateTemplate() {

        return hibernateTemplate;

    }

    public void setHibernateTemplate(HibernateTemplate hibernateTemplate) {

        this.hibernateTemplate = hibernateTemplate;

    }

    public boolean forgotPassword(String email,String password) {

        Session session=hibernateTemplate.getSessionFactory().openSession();

        Transaction tx=session.beginTransaction();

        @SuppressWarnings("unchecked")

        Query<Admin> q=session.createQuery("update Admin set password=:p where email=:e");

        q.setParameter("e",email);

        q.setParameter("p",password);

        int status=q.executeUpdate();

        System.out.println(status);

        tx.commit();

        return status>0?true:false;

    }

}

UserRepository.java

package org.sporty\_shoes.repository;

import java.util.List;

import org.hibernate.Query;

import org.hibernate.Session;

import org.hibernate.Transaction;

import org.sporty\_shoes.model.User;

import org.springframework.orm.hibernate5.HibernateTemplate;

@SuppressWarnings("deprecation")

public class UserRepository {

    HibernateTemplate hibernateTemplate;

    public UserRepository() {

        super();

    }

    public UserRepository(HibernateTemplate hibernateTemplate) {

        super();

        this.hibernateTemplate = hibernateTemplate;

    }

    public HibernateTemplate getHibernateTemplate() {

        return hibernateTemplate;

    }

    public void setHibernateTemplate(HibernateTemplate hibernateTemplate) {

        this.hibernateTemplate = hibernateTemplate;

    }

    public List<User> getAllUsers(){

        List<User> userList=hibernateTemplate.loadAll(User.class);

        return userList;

    }

    public List<User> searchUser(String email){

        Session session=hibernateTemplate.getSessionFactory().openSession();

        Transaction transaction=session.beginTransaction();

        String hql = "from User where userEmail=:email";

        Query<User> query = session.createQuery(hql,User.class);

        query.setParameter("email", email);

        List<User> user = query.getResultList();

        transaction.commit();

        session.close();

        return user;

    }

}