**Sporty shoes source code**

Admin.java

packageorg.sporty\_shoes.model;

importjava.io.Serializable;

importjavax.persistence.Entity;

importjavax.persistence.Id;

importjavax.persistence.Table;

@Entity

@Table(name = "admin")

publicclassAdminimplementsSerializable {

    privatestaticfinallongserialVersionUID = 1L;

    @Id

    privateStringemail;

    privateStringpassword;

    publicAdmin() {

        super();

    }

    publicAdmin(Stringemail, Stringpassword) {

        super();

        this.email = email;

        this.password = password;

    }

    publicStringgetEmail() {

        returnemail;

    }

    publicvoidsetEmail(Stringemail) {

        this.email = email;

    }

    publicStringgetPassword() {

        returnpassword;

    }

    publicvoidsetPassword(Stringpassword) {

        this.password = password;

    }

    @Override

    publicStringtoString() {

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

    }

}

MyController.java

packageorg.sporty\_shoes.controller;

importorg.sporty\_shoes.model.Admin;

importorg.sporty\_shoes.repository.AdminRepository;

importorg.sporty\_shoes.repository.ProductRepository;

importorg.sporty\_shoes.repository.PurchaseReportRepository;

importorg.sporty\_shoes.repository.Repository;

importorg.sporty\_shoes.repository.UserRepository;

importorg.springframework.beans.factory.annotation.Autowired;

importorg.springframework.stereotype.Controller;

importorg.springframework.ui.ModelMap;

importorg.springframework.web.bind.annotation.RequestMapping;

importorg.springframework.web.bind.annotation.RequestMethod;

importorg.springframework.web.bind.annotation.RequestParam;

@Controller

publicclassMyController {

    @Autowired

    AdminRepositoryadminRepository;

    @Autowired

    ProductRepositoryproductRepository;

    @Autowired

    UserRepository  userRepository;

    @Autowired

    PurchaseReportRepositorypurchaseReportRepository;

    @Autowired

    Repositoryrepository;

    @RequestMapping("/")

    publicStringshowHome() {

        return"Home";

    }

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

    publicStringadminLoginPage() {

        return"AdminLogin";

    }

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

    publicStringforgotPassword() {

        return"ForgotPassword";

    }

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

    publicStringadminForgotPasswordPage(@RequestParam("email")Stringemail,

            @RequestParam("password")Stringpassword,ModelMapmap) {

        if(repository.forgotPassword(email,password))

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

        else

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

        return"ForgotPassword";

    }

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

    publicStringadminPage(@RequestParam(name="email")Stringemail,

            @RequestParam(name="password")Stringpassword,ModelMapmap) {

        if(adminRepository.verifyAdmin(newAdmin(email,password)))

            return"AdminPage";

        else {

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

            return"AdminLogin";

        }

    }

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

    publicStringgetAllProducts(ModelMapmap) {

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

        return"Product";

    }

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

    publicStringgetAllUsers(ModelMapmap) {

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

        return"User";

    }

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

    publicStringsearchUser(@RequestParam("email")Stringemail,ModelMapmap) {

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

        return"User";

    }

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

    publicStringgetRport(ModelMapmap) {

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

        return"PurchaseReport";

    }

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

    publicStringdeleteProduct(@RequestParam("id")intid,ModelMapmap) {

        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

packageorg.sporty\_shoes.model;

importjava.io.Serializable;

importjavax.persistence.Entity;

importjavax.persistence.GeneratedValue;

importjavax.persistence.GenerationType;

importjavax.persistence.Id;

importjavax.persistence.Table;

@Entity

@Table(name="product")

publicclassProductimplementsSerializable{

    privatestaticfinallongserialVersionUID = 1L;

    @Id

    @GeneratedValue(strategy=GenerationType.AUTO)

    privateintproductId;

    privateStringproductName;

    privateintproductPrice;

    privateStringproductDiscription;

    publicProduct() {

        super();

    }

    publicProduct(StringproductName, intproductPrice, StringproductDiscription) {

        super();

        this.productName = productName;

        this.productPrice = productPrice;

        this.productDiscription = productDiscription;

    }

    publicintgetProductId() {

        returnproductId;

    }

    publicvoidsetProductId(intproductId) {

        this.productId = productId;

    }

    publicStringgetProductName() {

        returnproductName;

    }

    publicvoidsetProductName(StringproductName) {

        this.productName = productName;

    }

    publicintgetProductPrice() {

        returnproductPrice;

    }

    publicvoidsetProductPrice(intproductPrice) {

        this.productPrice = productPrice;

    }

    publicStringgetProductDiscription() {

        returnproductDiscription;

    }

    publicvoidsetProductDiscription(StringproductDiscription) {

        this.productDiscription = productDiscription;

    }

    @Override

    publicStringtoString() {

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

                productId, productName, productPrice, productDiscription);

    }

}

PurchaseReport.java

packageorg.sporty\_shoes.model;

importjava.io.Serializable;

importjavax.persistence.Entity;

importjavax.persistence.GeneratedValue;

importjavax.persistence.GenerationType;

importjavax.persistence.Id;

importjavax.persistence.Table;

@Entity

@Table(name="purchase\_report")

publicclassPurchaseReportimplementsSerializable {

    privatestaticfinallongserialVersionUID = 1L;

    @Id

    @GeneratedValue(strategy=GenerationType.AUTO)

    privateintreportId;

    privateStringreportProductName;

    privateStringreportUserEmail;

    privateStringreportDate;

    privateintreportPrice;

    publicPurchaseReport() {

        super();

    }

    publicPurchaseReport(StringreportProductName, StringreportUserEmail, StringreportDate, intreportPrice) {

        super();

        this.reportProductName = reportProductName;

        this.reportUserEmail = reportUserEmail;

        this.reportDate = reportDate;

        this.reportPrice = reportPrice;

    }

    publicintgetReportId() {

        returnreportId;

    }

    publicvoidsetReportId(intreportId) {

        this.reportId = reportId;

    }

    publicStringgetReportProductName() {

        returnreportProductName;

    }

    publicvoidsetReportProductName(StringreportProductName) {

        this.reportProductName = reportProductName;

    }

    publicStringgetReportUserEmail() {

        returnreportUserEmail;

    }

    publicvoidsetReportUserEmail(StringreportUserEmail) {

        this.reportUserEmail = reportUserEmail;

    }

    publicStringgetReportDate() {

        returnreportDate;

    }

    publicvoidsetReportDate(StringreportDate) {

        this.reportDate = reportDate;

    }

    publicintgetReportPrice() {

        returnreportPrice;

    }

    publicvoidsetReportPrice(intreportPrice) {

        this.reportPrice = reportPrice;

    }

    publicstaticlonggetSerialversionuid() {

        returnserialVersionUID;

    }

    @Override

    publicStringtoString() {

        returnString.format(

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

                reportId, reportProductName, reportUserEmail, reportDate, reportPrice);

    }

}

User.java

packageorg.sporty\_shoes.model;

importjava.io.Serializable;

importjavax.persistence.Entity;

importjavax.persistence.GeneratedValue;

importjavax.persistence.GenerationType;

importjavax.persistence.Id;

importjavax.persistence.Table;

@Entity

@Table(name="user")

publicclassUserimplementsSerializable {

    privatestaticfinallongserialVersionUID = 1L;

    @Id

    @GeneratedValue(strategy=GenerationType.AUTO)

    privateintuserId;

    privateStringuserName;

    privateStringuserEmail;

    privateintuserAge;

    privateStringuserGender;

    privateStringuserAddress;

    publicUser() {

        super();

    }

    publicUser(StringuserName, StringuserEmail, intuserAge, StringuserGender, StringuserAddress) {

        super();

        this.userName = userName;

        this.userEmail = userEmail;

        this.userAge = userAge;

        this.userGender = userGender;

        this.userAddress = userAddress;

    }

    publicintgetUserId() {

        returnuserId;

    }

    publicvoidsetUserId(intuserId) {

        this.userId = userId;

    }

    publicStringgetUserName() {

        returnuserName;

    }

    publicvoidsetUserName(StringuserName) {

        this.userName = userName;

    }

    publicStringgetUserEmail() {

        returnuserEmail;

    }

    publicvoidsetUserEmail(StringuserEmail) {

        this.userEmail = userEmail;

    }

    publicintgetUserAge() {

        returnuserAge;

    }

    publicvoidsetUserAge(intuserAge) {

        this.userAge = userAge;

    }

    publicStringgetUserGender() {

        returnuserGender;

    }

    publicvoidsetUserGender(StringuserGender) {

        this.userGender = userGender;

    }

    publicStringgetUserAddress() {

        returnuserAddress;

    }

    publicvoidsetUserAddress(StringuserAddress) {

        this.userAddress = userAddress;

    }

    publicstaticlonggetSerialversionuid() {

        returnserialVersionUID;

    }

    @Override

    publicStringtoString() {

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

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

    }

}

AdminRepository.java

packageorg.sporty\_shoes.repository;

importjava.util.List;

importorg.hibernate.Query;

importorg.hibernate.Session;

importorg.hibernate.Transaction;

importorg.sporty\_shoes.model.Admin;

importorg.springframework.orm.hibernate5.HibernateTemplate;

@SuppressWarnings("deprecation")

publicclassAdminRepository {

    privateHibernateTemplatehibernateTemplate;

    publicHibernateTemplategetHibernateTemplate() {

        returnhibernateTemplate;

    }

    publicvoidsetHibernateTemplate(HibernateTemplatehibernateTemplate) {

        this.hibernateTemplate = hibernateTemplate;

    }

    publicbooleanverifyAdmin(Admina) {

        Sessionsession=hibernateTemplate.getSessionFactory().openSession();

        Transactiontransaction=session.beginTransaction();

        Stringhql = "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();

        returnresults.size()>0;

    }

}

ProductRepository.java

packageorg.sporty\_shoes.repository;

importjava.util.List;

importorg.hibernate.Session;

importorg.hibernate.Transaction;

importorg.sporty\_shoes.model.Product;

importorg.springframework.orm.hibernate5.HibernateTemplate;

publicclassProductRepository {

    HibernateTemplatehibernateTemplate;

    publicProductRepository() {

        super();

    }

    publicProductRepository(HibernateTemplatehibernateTemplate) {

        super();

        this.hibernateTemplate = hibernateTemplate;

    }

    publicHibernateTemplategetHibernateTemplate() {

        returnhibernateTemplate;

    }

    publicvoidsetHibernateTemplate(HibernateTemplatehibernateTemplate) {

        this.hibernateTemplate = hibernateTemplate;

    }

    publicList<Product>getAllProducts(){

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

        returnproductList;

    }

    publicBooleandeleteProduct(intid) {

        try {

        Sessionsession=hibernateTemplate.getSessionFactory().openSession();

        Transactiontransaction=session.beginTransaction();

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

        session.delete(p);

        transaction.commit();

        session.close();

        }

        catch(Exceptione) {

            returnfalse;

        }

        returntrue;

    }

}

PurchaseReportRepository.java

packageorg.sporty\_shoes.repository;

importjava.util.List;

importorg.sporty\_shoes.model.PurchaseReport;

importorg.springframework.orm.hibernate5.HibernateTemplate;

publicclassPurchaseReportRepository {

    HibernateTemplatehibernateTemplate;

    publicHibernateTemplategetHibernateTemplate() {

        returnhibernateTemplate;

    }

    publicvoidsetHibernateTemplate(HibernateTemplatehibernateTemplate) {

        this.hibernateTemplate = hibernateTemplate;

    }

    @SuppressWarnings("deprecation")

    publicList<PurchaseReport>getReport(){

        @SuppressWarnings("unchecked")

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

        returnreport;

    }

}

Repository.java

packageorg.sporty\_shoes.repository;

importorg.hibernate.Query;

importorg.hibernate.Session;

importorg.hibernate.Transaction;

importorg.sporty\_shoes.model.Admin;

importorg.springframework.orm.hibernate5.HibernateTemplate;

@SuppressWarnings("deprecation")

publicclassRepository {

    privateHibernateTemplatehibernateTemplate;

    publicHibernateTemplategetHibernateTemplate() {

        returnhibernateTemplate;

    }

    publicvoidsetHibernateTemplate(HibernateTemplatehibernateTemplate) {

        this.hibernateTemplate = hibernateTemplate;

    }

    publicbooleanforgotPassword(Stringemail,Stringpassword) {

        Sessionsession=hibernateTemplate.getSessionFactory().openSession();

        Transactiontx=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);

        intstatus=q.executeUpdate();

        System.out.println(status);

        tx.commit();

        returnstatus>0?true:false;

    }

}

UserRepository.java

packageorg.sporty\_shoes.repository;

importjava.util.List;

importorg.hibernate.Query;

importorg.hibernate.Session;

importorg.hibernate.Transaction;

importorg.sporty\_shoes.model.User;

importorg.springframework.orm.hibernate5.HibernateTemplate;

@SuppressWarnings("deprecation")

publicclassUserRepository {

    HibernateTemplatehibernateTemplate;

    publicUserRepository() {

        super();

    }

    publicUserRepository(HibernateTemplatehibernateTemplate) {

        super();

        this.hibernateTemplate = hibernateTemplate;

    }

    publicHibernateTemplategetHibernateTemplate() {

        returnhibernateTemplate;

    }

    publicvoidsetHibernateTemplate(HibernateTemplatehibernateTemplate) {

        this.hibernateTemplate = hibernateTemplate;

    }

    publicList<User>getAllUsers(){

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

        returnuserList;

    }

    publicList<User>searchUser(Stringemail){

        Sessionsession=hibernateTemplate.getSessionFactory().openSession();

        Transactiontransaction=session.beginTransaction();

        Stringhql = "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();

        returnuser;

    }

}

