# 1- Hibernate\_2 class

package Connection;

import java.util.Properties;

import org.hibernate.SessionFactory;

import org.hibernate.boot.registry.StandardServiceRegistryBuilder;

import org.hibernate.cfg.Configuration;

import org.hibernate.cfg.Environment;

import org.hibernate.service.ServiceRegistry;

import Entities.Products;

public class Hibernate\_2 {

private static SessionFactory sessionFactory;

public static SessionFactory getSessionFactory(String DML) {

if (sessionFactory == null) {

try {

Configuration configuration = new Configuration();

// Hibernate settings equivalent to hibernate.cfg.xml's properties

Properties settings = new Properties();

settings.put(Environment.DRIVER, "com.mysql.jdbc.Driver");

settings.put(Environment.URL, "jdbc:mysql://localhost:3306/hibernate"); // Database schema name

settings.put(Environment.USER, "jroot"); //DB User Name

settings.put(Environment.PASS, "1234"); // DB Password

settings.put(Environment.DIALECT, "org.hibernate.dialect.MySQL5Dialect");

settings.put(Environment.SHOW\_SQL, "true");

settings.put(Environment.CURRENT\_SESSION\_CONTEXT\_CLASS, "thread");

if(DML=="create")

settings.put(Environment.HBM2DDL\_AUTO, "create");

configuration.setProperties(settings);

// The Entities

configuration.addAnnotatedClass(Products.class);

ServiceRegistry serviceRegistry = new StandardServiceRegistryBuilder()

.applySettings(configuration.getProperties()).build();

System.out.println("Hibernate Java Config serviceRegistry created");

sessionFactory = configuration.buildSessionFactory(serviceRegistry);

return sessionFactory;

} catch (Exception e) {

e.printStackTrace();

}

}

return sessionFactory;

}

}

# 2- Products Entity class

package Entities;

import javax.persistence.\*;

@Entity

public class Products {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String name;

private int price;

public Products() {

System.out.println("Product defult constructure runs..");

}

public Products(String name, int price) {

super();

this.name = name;

this.price = price;

}

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 int getPrice() {

return price;

}

public void setPrice(int price) {

this.price = price;

}

@Override

public String toString() {

return "Products [id=" + id + ", name=" + name + ", price=" + price + "]";

}

}

# 3- Product\_DAL class

package Products\_DAL;

import java.util.List;

import org.hibernate.Session;

import org.hibernate.Transaction;

import org.hibernate.query.Query;

import Connection.Hibernate\_2;

import Entities.Products;

public class Product\_DAL {

public void saveUser(Products product, String DML) {

Transaction transaction = null;

try (Session session = Hibernate\_2.getSessionFactory(DML).openSession()) {

// start a transaction

transaction = session.beginTransaction();

// save the student object

session.save(product);

// commit transaction

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

}

}

@SuppressWarnings("unchecked")

public List < Products > getAllProducts() {

Transaction transaction = null;

List < Products > listOfUser = null;

try (Session session = Hibernate\_2.getSessionFactory("create-drop").openSession()) {

// start a transaction

transaction = session.beginTransaction();

// get an user object

listOfUser = session.createQuery("from Entities.Products").getResultList();

System.out.println("The list: "+listOfUser);

// commit transaction

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

//transaction.rollback();

System.out.println("transaction closed!!");

}

e.printStackTrace();

}

return listOfUser;

}

public void deleteProduct(int id, String DML) {

Transaction transaction = null;

try (Session session = Hibernate\_2.getSessionFactory(DML).openSession()) {

// start a transaction

transaction = session.beginTransaction();

// Delete a user object

Products user = session.get(Products.class, id);

if (user != null) {

session.delete(user);

System.out.println("Product is deleted");

}

// commit transaction

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

}

}

}

# 4- productServlet Servlet class

package Servlets;

import java.io.IOException;

import java.sql.SQLException;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import java.util.List;

import Products\_DAL.Product\_DAL;

import Entities.Products;

public class productServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

private Product\_DAL pDal;

private static String name;

private static String price;

public void init() {

pDal = new Product\_DAL();

}

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

doGet(request, response);

}

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

String action = request.getServletPath();

try {

switch (action) {

case "/insert":

insertUser(request, response);

break;

case "/list":

listUser(request, response);

break;

case "/delete":

deleteSubject(request, response);

break;

}

} catch (SQLException ex) {

throw new ServletException(ex);

}

}

private void insertUser(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException {

name = request.getParameter("product\_name");

price = request.getParameter("product\_price");

System.out.println("name and price are: "+name+"-"+price);

if(name.equals("") || price.equals("")) {

request.setAttribute("TheState","");

request.setAttribute("TheState\_2","missing");

}

else {

request.setAttribute("TheState","not null");

request.setAttribute("TheState\_2","not null");

request.setAttribute("name", name);

request.setAttribute("price", price);

try {

Products newUser = new Products(name,Integer.parseInt(price));

System.out.println("4");

pDal.saveUser(newUser,"update"); // 2nd argument for the DML type (CRUD)

}catch (Exception e) {

request.setAttribute("TheState\_2","number");

}

}

try {

listUser(request, response);

} catch (Exception e) {

System.out.println("list subjects error!!");

}

}

private void listUser(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException, ServletException {

List < Products > listProduct = pDal.getAllProducts();

request.setAttribute("listProduct", listProduct);

RequestDispatcher dispatcher = request.getRequestDispatcher("product\_info.jsp");

dispatcher.forward(request, response);

}

private void deleteSubject(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException, ServletException {

System.out.println("in delete ");

int id = Integer.parseInt(request.getParameter("id"));

pDal.deleteProduct(id, "none");

try {

listUser(request, response);

} catch (Exception e) {

System.out.println("list subjects error!!");

}

}

}

# 5- product\_info.jsp HTML Page

<%@page import="java.util.ArrayList"%>

<%@page import="java.util.List"%>

<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" isELIgnored="false"%>

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Insert title here</title>

<style type="text/css">

body{

background-color: #540B0E;

}

h1{

text-align: center; color: #EDDDD4;

font-family: Segoe UI;

font-size: 40px;

}

hr.new5 {

margin-top: -30px;

margin-left: 20%;

margin-right: 20%;

border: 1px solid #EDDDD4;

border-radius: 10px;

}

form{

text-align: center;

}

#span1{

display: inline-block;

text-align: left;

color: #EDDDD4;

font-family: Segoe UI;

font-size: 20px;

padding: 2px 10px;

padding-bottom: 20px;

}

input[type=text],input[type=password]{

width: 200px;

padding: 3px 10px;

margin: 8px 0;

box-sizing: border-box;

font-family: Segoe UI;

font-size: 15px;

}

input[type=button], input[type=submit], input[type=reset] {

background-color: #197278;

border: none;

color: #EDDDD4;

padding: 3px 30px;

text-decoration: none;

margin: 4px 2px;

cursor: pointer;

font-family: Segoe UI;

font-size: 20px;

}

table{

text-align: center; color: #EDDDD4;

font-family: Segoe UI;

font-size: 20px;

border: 1px solid #EDDDD4;

border-radius: 10px;

}

</style>

</head>

<body>

<h1>Product information</h1>

<hr class="new5">

<form action="insert" class="form">

<span id="span1">

Product Name

<%

String x="";

String y="";

String state=(String)request.getAttribute("TheState");

String state\_2=(String)request.getAttribute("TheState\_2");

if(state==null){

}

else if( state.equals("")){

}

else{

x=(String)request.getAttribute("name");

y=(String)request.getAttribute("price");

}

%>

<input type="text" placeholder="Product Name" name="product\_name" value="<%=x%>" ><br/>

Product Price

<input type="text" placeholder="Product Price" name="product\_price" value="<%=y%>" ><br/>

<%

if(state==null){

}

else if( state\_2.equals("number")){

%>Price take only numbers!! <%

}

else if( state\_2.equals("missing")){

%>Missing fields !!<%

}

%>

</span>

<br/>

<input type="submit" value="Save">

</form>

<div align="center">

<table border="1" cellpadding="10">

<caption><h2>Product information</h2></caption>

<tr>

<th>ID</th>

<th>Name</th>

<th>Price</th>

</tr>

<c:forEach var="user" items="${listProduct}">

<tr>

<td><c:out value="${user.id}" /></td>

<td><c:out value="${user.name}" /></td>

<td><c:out value="${user.price}" /></td>

<td>

&nbsp;&nbsp;&nbsp;&nbsp;

<a href="delete?id=<c:out value='${user.id}' />">Delete</a>

</td>

</tr>

</c:forEach>

<%

if(request.getAttribute("start") == "none" ){

%> <div style="text-align: center; color: #EDDDD4;

font-family: Segoe UI;

font-size: 20px;"> No Recourd Found!! </div> <%

}

%>

</table>

</div>

</body>

</html>

# 6- web.xml File

<!DOCTYPE web-app PUBLIC

"-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"

"http://java.sun.com/dtd/web-app\_2\_3.dtd" >

<web-app>

<display-name>Archetype Created Web Application</display-name>

<servlet>

<description></description>

<servlet-name>product\_Servlet</servlet-name>

<display-name>productServlet</display-name>

<description></description>

<servlet-class>Servlets.productServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>product\_Servlet</servlet-name>

<url-pattern>/insert</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>product\_Servlet</servlet-name>

<url-pattern>/list</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>product\_Servlet</servlet-name>

<url-pattern>/delete</url-pattern>

</servlet-mapping>

</web-app>

# 7- pom.xml File

<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 http://maven.apache.org/maven-v4\_0\_0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>test1</groupId>

<artifactId>Practice\_3-20\_Add\_Products</artifactId>

<packaging>war</packaging>

<version>0.0.1-SNAPSHOT</version>

<name>Practice\_3-20\_Add\_Products Maven Webapp</name>

<url>http://maven.apache.org</url>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

<!-- https://mvnrepository.com/artifact/javax.servlet.jsp/javax.servlet.jsp-api -->

<dependency>

<groupId>javax.servlet.jsp</groupId>

<artifactId>javax.servlet.jsp-api</artifactId>

<version>2.3.1</version>

<scope>provided</scope>

</dependency>

<!-- https://mvnrepository.com/artifact/javax.servlet/jstl -->

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>jstl</artifactId>

<version>1.2</version>

</dependency>

<!-- https://mvnrepository.com/artifact/javax.servlet/servlet-api -->

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>servlet-api</artifactId>

<version>2.5</version>

<scope>provided</scope>

</dependency>

<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>8.0.26</version>

</dependency>

<!-- https://mvnrepository.com/artifact/javax.servlet/jstl -->

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>jstl</artifactId>

<version>1.2</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-core -->

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>5.5.5.Final</version>

</dependency>

<!-- https://mvnrepository.com/artifact/commons-logging/commons-logging -->

<dependency>

<groupId>commons-logging</groupId>

<artifactId>commons-logging</artifactId>

<version>1.2</version>

</dependency>

<!-- https://mvnrepository.com/artifact/javax.servlet/javax.servlet-api -->

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>javax.servlet-api</artifactId>

<version>4.0.1</version>

<scope>provided</scope>

</dependency>

</dependencies>

<build>

<finalName>Practice\_3-20\_Add\_Products</finalName>

</build>

</project>