**Adding a New Product in the Database**

As a part of developing an e-commerce web application, you have to create a database table for storing product information. A form is needed to add new products. The form submission is validated, and a new record is created in the product table.

**Solution:**

// index.html

<!DOCTYPE html>

<html>

<head>

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

<title>Insert title here</title>

</head>

<body>

<h1>Welcome to Hibernate CRUD application</h1>

<a href=*"read-product"*>Read Product</a>

<br />

<a href=*"add-product"*>Add Product</a>

</body>

</html>

//add-product.html

<!DOCTYPE html>

<html>

<head>

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

<title>Insert title here</title>

</head>

<body>

<h1> Add Product</h1>

<form method=*"post"* action=*"add-product"*>

<fieldset>

<legend>Add Product form</legend>

Product Name: <input type=*"text"* name=*"name"* id=*"name"*

/><br/><br/>

Product Price: <input type=*"text"* name=*"price"* id =

*"price"* /> <br/><br/>

<input type=*"submit"* value=*"Add Product"*>

</fieldset>

</form>

</body>

</html>

//hibernate.cfg.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE hibernate-configuration SYSTEM ["http://www.hibernate.org/dtd/hibernate](http://www.hibernate.org/dtd/hibernate-configuration-5.3.dtd)-[configuration-5.3.dtd"](http://www.hibernate.org/dtd/hibernate-configuration-5.3.dtd)>

<hibernate-configuration>

<session-factory>

<property name=*"hibernate.connection.driver\_class"*>com.mysql.jdbc.Driver</property>

<property name=*"hibernate.connection.url"*>jdbc:mysql://localhost:3306/Karunakar</prop erty>

<property name=*"hibernate.connection.username"*>root</property>

<property name=*"hibernate.connection.password"*></property>

<!-- Optional properties -->

<property name=*"hibernate.show\_sql"*>true</property>

<property name=*"hibernate.format\_sql"*>true</property>

<property name=*"hbm2ddl.auto"*>update</property>

</session-factory>

</hibernate-configuration>

//HibernateUtil.java

package com.karunakar.training;

import org.hibernate.SessionFactory; import org.hibernate.cfg.Configuration;

import com.entity.EProduct;

public class HibernateUtil {

public static SessionFactory buildSessionFactory() {

Configuration cfg = new Configuration(); cfg.configure("hibernate.cfg.xml"); cfg.addAnnotatedClass(EProduct.class);

SessionFactory sf = cfg.buildSessionFactory();

return sf;

}

}

// ReadProductServlet.java

package com.karunakar.training;

import java.io.IOException; import java.io.PrintWriter; import java.util.List;

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 org.hibernate.Session;

import org.hibernate.SessionFactory;

import com.entity.EProduct; import com.util.HibernateUtil;

@WebServlet("/read-product")

public class ReadProductServlet extends HttpServlet { private static final long serialVersionUID = 1L;

public ReadProductServlet() {

}

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

PrintWriter out = response.getWriter();

SessionFactory sf = HibernateUtil.buildSessionFactory();

Session session = sf.openSession();

List<EProduct> products = session.createQuery(" from EProduct").list();

out.print("<h1> Product List :- </h1>");

out.print("<style> table,td,th { border:2px solid red; padding: 10px; }</style>" ); out.print("<table>");

out.print("<tr>");

out.print("<th> Product Id </th>"); out.print("<th> Product Name </th>");

out.print("<th> Product Price </th>"); out.print("</tr>");

for(EProduct p : products) {

out.print("<tr>"); out.print("<td>"+p.getId()+"</td>"); out.print("<td>"+p.getName()+"</td>"); out.print("<td>"+p.getPrice()+"</td>"); out.print("</tr>");

}

out.print("</table>");

session.close();

}

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

doGet(request, response);

}

}

// AddProductServlet.java

package com.karunakar.training;

import java.io.IOException; import java.io.PrintWriter;

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 org.hibernate.Session;

import org.hibernate.SessionFactory; import org.hibernate.Transaction;

import com.entity.EProduct; import com.util.HibernateUtil;

@WebServlet("/add-product")

public class AddProductServlet extends HttpServlet { private static final long serialVersionUID = 1L;

public AddProductServlet() {

}

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException{

request.getRequestDispatcher("add-product.html").include(request, response);

}

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

PrintWriter out = response.getWriter();

String productName = request.getParameter("name"); String productPrice = request.getParameter("price");

//Step1: Gets session Factory

SessionFactory sf = HibernateUtil.buildSessionFactory();

//Step 2: Gets Session object Session session = sf.openSession();

//Step3: Gets Tx object and begin transaction Transaction tx = session.beginTransaction();

// Step 4: Create and populate entity object EProduct product = new EProduct(); product.setName(productName);

product.setPrice(Double.parseDouble(productPrice));

//Step 5: <h3 style='color:green'> ProductNischithNischith is created successfully !

</h3>Save record in DB

session.save(product);

tx.commit();

out.print("<h3> Product is created successfully ! <h3>"); session.close();

}

}

// EProduct.java

package com.karunakar.training;

import javax.persistence.Column; import javax.persistence.Entity;

import javax.persistence.GeneratedValue; import javax.persistence.Id;

import javax.persistence.Table;

@Entity @Table(name="eproduct\_27012022") public class EProduct {

@Id @GeneratedValue

@Column(name="eproduct\_id")

private int id;

@Column(name="eproduct\_name") private String name;

@Column(name="eproduct\_price") private Double 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 Double getPrice() {

return price;

}

public void setPrice(Double price) {

this.price = price;

}

}