# **Index.html file**

<!DOCTYPE html>

<html>

<head>

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

<title>Add Product</title>

</head>

<body>

<a href=*"list"*>Product-Details</a><br><br>

<form action=*"save"* method=*"post"*>

Name: &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

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

<input type=*"text"* name=*"pname"*><br><br>

Price: &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

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

<input type=*"text"* name=*"pprice"*><br><br>

Hard Disk Details: <input type=*"text"* name=*"hdd"*><br><br>

CPU: &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

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

<input type=*"text"* name=*"cpu"*><br><br>

RAM: &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

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

<input type=*"text"* name=*"ram"*><br><br>

<input type=*"submit"* value=*"Save data using Component"*>

</form>

</body>

</html>

# **Product Parts Class (ProductParts.java)**

**package** com;

**public** **class** ProductParts {

**private** String hdd;

**private** String cpu;

**private** String ram;

**public** ProductParts() {

}

**public** ProductParts(String hdd, String cpu, String ram) {

**this**.hdd = hdd;

**this**.cpu = cpu;

**this**.ram = ram;

}

**public** String getHdd() {

**return** hdd;

}

**public** **void** setHdd(String hdd) {

**this**.hdd = hdd;

}

**public** String getCpu() {

**return** cpu;

}

**public** **void** setCpu(String cpu) {

**this**.cpu = cpu;

}

**public** String getRam() {

**return** ram;

}

**public** **void** setRam(String ram) {

**this**.ram = ram;

}

}

# **eProduct Class (EProduct.java)**

package com;

import java.math.BigDecimal;

import java.util.Date;

public class EProduct {

private long ID;

private String name;

private BigDecimal price;

private Date dateAdded;

private ProductParts parts;

public EProduct() {

}

public EProduct(String name, BigDecimal price, ProductParts parts) {

this.name = name;

this.price = price;

this.parts = parts;

}

public long getID() {

return ID;

}

public void setID(long iD) {

ID = iD;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public BigDecimal getPrice() {

return price;

}

public void setPrice(BigDecimal price) {

this.price = price;

}

public Date getDateAdded() {

return dateAdded;

}

public void setDateAdded(Date dateAdded) {

this.dateAdded = dateAdded;

}

public ProductParts getParts() {

return parts;

}

public void setParts(ProductParts parts) {

this.parts = parts;

}

}

# **Hibernate Utility Class (HibernateUtil.java)**

package com;

import org.hibernate.SessionFactory;

import org.hibernate.boot.Metadata;

import org.hibernate.boot.MetadataSources;

import org.hibernate.boot.registry.StandardServiceRegistry;

import org.hibernate.boot.registry.StandardServiceRegistryBuilder;

public class HibernateUtil {

private static SessionFactory sessionFactory;

static {

try {

StandardServiceRegistry registry= new StandardServiceRegistryBuilder().

configure("hibernate.cfg.xml").build();

Metadata metadata= new MetadataSources(registry).getMetadataBuilder().build();

sessionFactory= metadata.getSessionFactoryBuilder().build();

} catch (Exception e) {

e.printStackTrace();

}

}

public static SessionFactory getSessionFactory() {

return sessionFactory;

}

}

# **List Data Servlet (ListData.java)**

package com;

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;

@WebServlet("/list")

public class ListData extends HttpServlet {

private static final long serialVersionUID = 1L;

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

response.setContentType("text/html");

SessionFactory factory = HibernateUtil.getSessionFactory();

Session session = factory.openSession();

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

PrintWriter out = response.getWriter();

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

for(EProduct p: list) {

out.print(p.getID()+": "+p.getName()+" "+p.getPrice());

out.print("<br>Product Parts: <br>");

ProductParts parts = p.getParts();

out.print("HDD "+parts.getHdd()+"<br>");

out.print("CPU "+parts.getCpu()+"<br>");

out.print("RAM "+parts.getRam()+"<br>");

out.print("<hr>");

}

}

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

// TODO Auto-generated method stub

doGet(request, response);

}

}

# **Save Servlet (SaveServlet.java)**

package com;

import java.io.IOException;

import java.math.BigDecimal;

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;

@WebServlet("/save")

public class SaveServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

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

String name = request.getParameter("pname");

String price = request.getParameter("pprice");

String hdd = request.getParameter("hdd");

String cpu = request.getParameter("cpu");

String ram = request.getParameter("ram");

ProductParts parts = new ProductParts(hdd, cpu, ram);

EProduct product = new EProduct(name, new BigDecimal(price), parts);

response.setContentType("text/html");

SessionFactory factory= HibernateUtil.getSessionFactory();

Session session= factory.openSession();

Transaction t = session.beginTransaction();

session.save(product);

t.commit();

response.sendRedirect("list");

}

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

// TODO Auto-generated method stub

doGet(request, response);

}

}

# **Hibernate Configuration File (hibernate.cfg.xml)**

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

<!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

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

<property name=*"connection.url"*>jdbc:mysql://localhost:3306/component</property>

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

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

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

<property name=*"dialect"*>org.hibernate.dialect.MySQL8Dialect</property>

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

<mapping resource=*"eproduct.hbm.xml"* />

</session-factory>

</hibernate-configuration>

# **Hibernate Mapping File (eproduct.hbm.xml)**

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

<!DOCTYPE hibernate-mapping PUBLIC

"-//Hibernate/Hibernate Mapping DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">

<hibernate-mapping package=*"com"*>

<class name=*"EProduct"* table=*"eproduct"*>

<id name=*"ID"* type=*"long"* column=*"ID"*>

<generator class=*"identity"*></generator>

</id>

<property name=*"name"* type=*"string"* column=*"NAME"*></property>

<property name=*"price"* type=*"big\_decimal"* column=*"PRICE"*></property>

<property name=*"dateAdded"* type=*"timestamp"* generated=*"always"* column=*"DATE\_ADDED"*></property>

<component name=*"parts"* class=*"com.ProductParts"*>

<property name=*"hdd"* column=*"parts\_hdd"* type=*"string"*></property>

<property name=*"cpu"* column=*"parts\_cpu"* type=*"string"*></property>

<property name=*"ram"* column=*"parts\_ram"* type=*"string"*></property>

</component>

</class>

</hibernate-mapping>