```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<h2>Product Management System</h2>
<a href="addProduct.jsp">Add Product</a> |
<a href="ProductController">Display Product</a> |
<a href="deleteProduct.jsp">Delete Product</a> |
</body>
</html>
```

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<h2>Add Product</h2>
<form action="ProductController" method="post">
<input type="hidden" name="operation" value="add">
<label>PName</label>
<input type="text" name="pname"/><br/>
<label>Price</label>
<input type="number" name="price"/><br/>
<label>URL</label>
<input type="url" name="url"/><br/>
<input type="submit" value="Store Product"/>
<input type="reset" value="reset"/>
</form>
<br/>
<a href="index.jsp">Back</a>
</body>
</html>
```

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<h2>Delete Product</h2>
<form action="ProductController" method="post">
<input type="hidden" name="operation" value="delete">
<label>PId</label>
```

```
<input type="number" name="pid"/><br/>
<input type="submit" value="Delete Product"/>
<input type="reset" value="reset"/>
</form>
<br/>
<a href="index.jsp">Back</a>
</body>
</html>
```

```
<%@page import="java.util.Iterator"%>
<%@page import="com.pms.entity.Product"%>
<%@page import="java.util.List"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<table border="1">
<tr>
<th>PId</th>
<th>PName</th>
<th>Price</th>
<th>URL</th>
</tr>
<%
Object obj = session.getAttribute("products");
List<Product> listOfProduct = (List<Product>)obj;
Iterator<Product> li = listOfProduct.iterator();
while(li.hasNext()){
Product p = li.next();
%>
<tr>
<td><%=p.getPid()%> </td>
<td><%=p.getPname()%> </td>
<td><%=p.getPrice()%> </td>
<td><img src=<%=p.getUrl()%> width="85px" height="85px"/> </td>
</tr>
<%
}
%>
</table>
<br/>
<a href="index.jsp">Back</a>
</body>
</html>
```

```
package com.pms.entity;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

@Entity
public class Product {

@Id
@GeneratedValue(strategy = GenerationType.IDENTITY)
private int pid;
private String pname;
private float price;
private String url;

public Product() {
super();
// TODO Auto-generated constructor stub
}
public Product(int pid, String pname, float price, String url) {
super();
this.pid = pid;
this.pname = pname;
this.price = price;
this.url = url;
}
public int getPid() {
return pid;
}
public void setPid(int pid) {
this.pid = pid;
}
public String getPname() {
return pname;
}
public void setPname(String pname) {
this.pname = pname;
}
public float getPrice() {
return price;
}
public void setPrice(float price) {
this.price = price;
}
public String getUrl() {
return url;
}
public void setUrl(String url) {
this.url = url;
}
}
```

```
package com.pms.resource;

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

public class SessionFactoryResource {
```

```
static SessionFactory sf;

// This code execute only once.
static {
Configuration con = new Configuration();
con.configure("hibernate.cfg.xml");
sf = con.buildSessionFactory();
}

public static SessionFactory getSessionFactory() {
try {
return sf;
}catch(Exception e) {
System.err.println("Session Factory error "+e);
return null;
}
}
}
package com.pms.dao;

import java.util.List;

import javax.persistence.TypedQuery;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import com.pms.entity.Product;
import com.pms.resource.SessionFactoryResource;

public class ProductDao {

// alter table product modify column url varchar(10000);
// alter table product modify column url blob;

public int storeProduct(Product product) {
try {
SessionFactory sf = SessionFactoryResource.getSessionFactory();
Session session = sf.openSession();
Transaction tran = session.getTransaction();
tran.begin();
session.save(product);
tran.commit();
return 1;
} catch (Exception e) {
System.err.println(e);
return 0;
}
}

public List<Product> findAllProduct(){
SessionFactory sf=SessionFactoryResource.getSessionFactory();
Session session =sf.openSession();
TypedQuery tq= session.createQuery("from Product");
```

```java
List<Product> listOfProduct = tq.getResultList();
return listOfProduct;
}

public int deleteProduct(int pid) {

SessionFactory sf = SessionFactoryResource.getSessionFactory();
Session session = sf.openSession();
Transaction tran = session.getTransaction();
Product p = session.find(Product.class, pid);

if(p==null) {
return 0;
}else {
tran.begin();
session.delete(p);
tran.commit();
return 1;
}
}
}
```

```java
package com.pms.service;

import java.util.List;

import com.pms.dao.ProductDao;
import com.pms.entity.Product;

public class ProductService {

ProductDao pd=new ProductDao();

public String storeProduct(Product product) {
if(product.getPrice()<1000) {
return "Price must be > 1000";
}else if(pd.storeProduct(product)>0) {
return "Product details stored successfully";
}else {
return "Product didn't store";
}


}


public List<Product> findAllProduct(){
//here we can use condition also
return pd.findAllProduct();
}

public String deleteProduct(int pid) {
if(pd.deleteProduct(pid)>0) {
return "Product deleted successfully";
}else {
return "Product not present";
}
}
} package com.pms.controller;

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

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 javax.servlet.http.HttpSession;

import com.pms.entity.Product;
import com.pms.service.ProductService;

/**
* Servlet implementation class ProductController
*/
@WebServlet("/ProductController")
public class ProductController extends HttpServlet {
private static final long serialVersionUID = 1L;
/**
* @see HttpServlet#HttpServlet()
*/
public ProductController() {
super();
// TODO Auto-generated constructor stub
}


/**
 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
response)
 */
protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
//    PrintWriter pw = response.getWriter();

ProductService ps = new ProductService();
List<Product> listOfProduct = ps.findAllProduct();

HttpSession hs = request.getSession();
hs.setAttribute("products", listOfProduct);

response.sendRedirect("displayProduct.jsp");
response.setContentType("text/html");
}

/**
 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
response)
 */
```

```
protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
PrintWriter pw = response.getWriter();                      // object
created.

//created service class object.
ProductService ps=new ProductService();
// receive the value from form

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

if(operation.equals("add")) {
String pname = request.getParameter("pname");
float price = Float.parseFloat(request.getParameter("price"));
String url = request.getParameter("url");

//convert those values to objects
Product p=new Product();
p.setPname(pname);
p.setPrice(price);
p.setUrl(url);


String result = ps.storeProduct(p);
pw.print(result);
RequestDispatcher rd = request.getRequestDispatcher("addProduct.jsp");
rd.include(request, response);

}else if(operation.equals("delete")) {
int pid = Integer.parseInt(request.getParameter("pid"));
String result = ps.deleteProduct(pid);
pw.println(result);
RequestDispatcher rd = request.getRequestDispatcher("deleteProduct.jsp");
rd.include(request, response);
}



response.setContentType("text/html");
}


}
```

```
<?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="hibernate.connection.driver_class">com.mysql.cj.jdbc.Driver</property
>
<property name="hibernate.connection.password">Un$ammu#2982</property>
```

```
<property
name="hibernate.connection.url">jdbc:mysql://localhost:3306/mydb_phase2_app
</property>
<property name="hibernate.connection.username">root</property>
<property
name="hibernate.dialect">org.hibernate.dialect.MySQL8Dialect</property>
<property name="hibernate.hbm2ddl.auto">update</property>
<mapping class="com.pms.entity.Product"/>
</session-factory>
</hibernate-configuration>
```