Author

College Name

Part 4

Table of Contents

[Overview 2](#_Toc21079971)

[Technologies 2](#_Toc21079972)

[Description 2](#_Toc21079973)

[Screenshots 2](#_Toc21079974)

[Project Structure 4](#_Toc21079975)

[Code 5](#_Toc21079976)

[JSP 5](#_Toc21079977)

[Servlets 9](#_Toc21079978)

[XMLS 23](#_Toc21079979)

Report

# Overview

The objective of the Project Maintenance Application to help users to add, update, delete and view all the products. This application undergoes 6 development stages. Following is the description of each stage –

1. Part 1 - Creation of skeletal structure of the application.
2. Part 2 - Adding Custom tags for validation.
3. Part 3 - Application development using JDBC.
4. Part 4 - Application development using EclipseLink JPA.
5. Part 5 – Introduction of SSL and Authentication.
6. Part 6 – Integration with existing Music Store Website

# Technologies

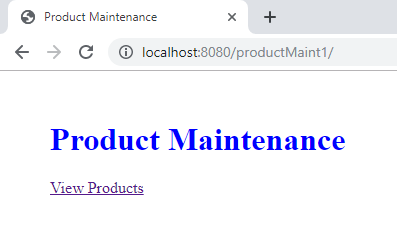
Java 1.7, Servlets, JSP, JDBC, JPA (EclipseLink), MySQL

# Description

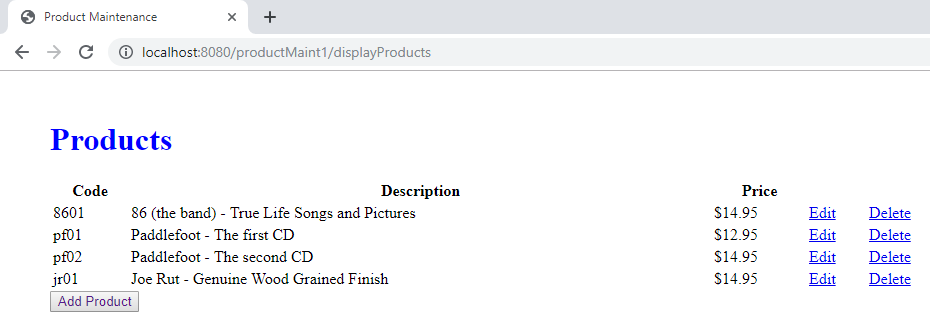
Part 4 of Product Maintenance Application introduces the entry level operation such as Add, Update, Delete and View. All the functionalities are achieved by reading the data from database tables using JPA Specification.

# Screenshots

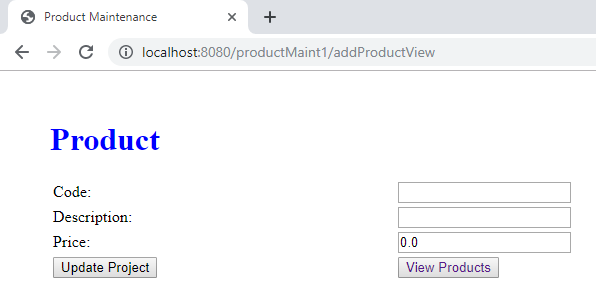
1. Welcome Page – This is the landing page when we start the project.



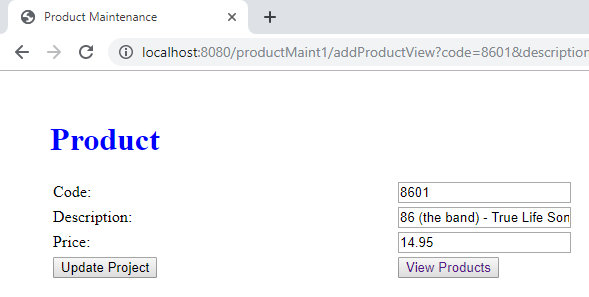
1. Display Products – After Clicking “View Products”, it will fetch all the products from database and display here.



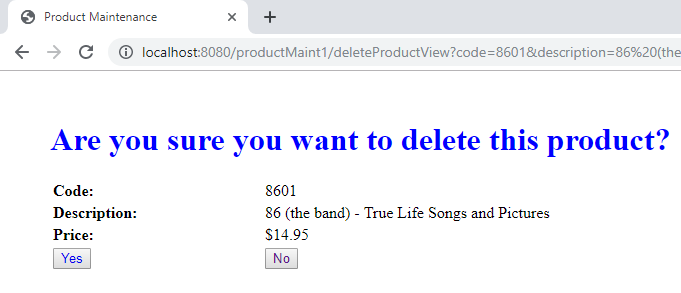
1. Add Product - After Clicking “Add Product”, it will display add product page where you can enter the details and add them to database.



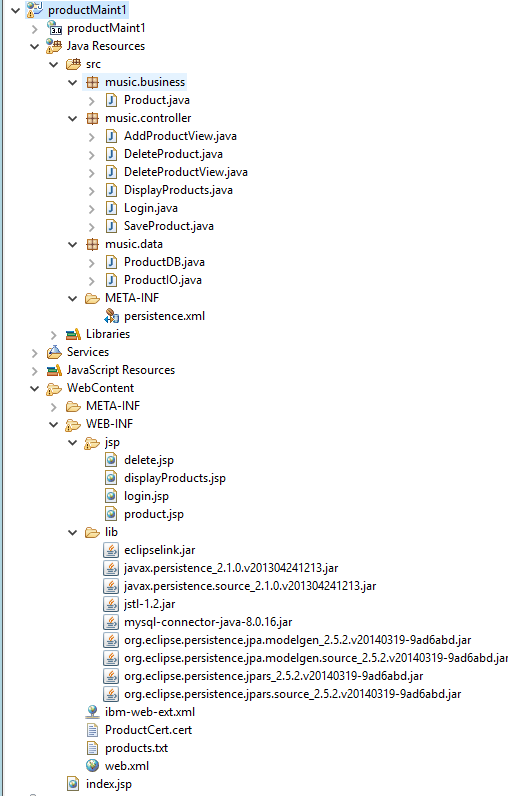
1. Update Product - After Clicking “Edit”, it will redirect to update product page with corresponding product data where you can update the details and add them to database.



1. Delete Product - After Clicking “Delete”, it will display delete confirmation page where you can choose to delete particular product from the database or go to display products page.



# Project Structure



# Code

## JSP

1. **Index.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

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

<title>Product Maintenance</title>

<style>

**body**{

margin: *50px*;

}

**h1**{

color: *blue*;

}

</style>

</head>

<body>

<h1>Product Maintenance</h1>

<a href=*"displayProducts"*>View Products</a>

</body>

</html>

1. **product.jsp**

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

<!DOCTYPE html>

<html>

<head>

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

<title>Product Maintenance</title>

<style>

body{

margin: 50px;

}

h1{

color: blue;

}

a{

text-decoration: none

}

</style>

</head>

<body>

<h1>Product</h1>

<form action="saveProduct">

<table style="width: 50%; collapse; background-color: white">

<tr>

<td><label for="code">Code: </label></td>

<td><input type="text" name="code" id="code" value = "${product.code}"/></td>

</tr>

<tr>

<td><label for="description">Description: </label></td>

<td><input type="text" name="description" id="description" value = "${product.description}" /></td>

</tr>

<tr>

<td><label for="price">Price: </label></td>

<td><input type="text" name="price" id="price" value = "${product.price}"/></td>

</tr>

<tr>

<td><input type="submit" value="Update Project" /></td>

<td><button type="button">

<a href="displayProducts">View Products</a>

</button></td>

</tr>

</table>

</form>

</body>

</html>

1. **displayProducts.jsp**

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

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

<!DOCTYPE html>

<html>

<head>

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

<title>Product Maintenance</title>

<style>

body{

margin: 50px;

}

h1{

color: blue;

}

</style>

</head>

<body>

<h1>Products</h1>

<div class="table">

<table style="width: 50%; collapse; background-color: white">

<tr>

<th>Code</th>

<th>Description</th>

<th>Price</th>

<th></th>

<th></th>

</tr>

<c:forEach items="${productList}" var="product">

<tr>

<td height='9px'>${product.code}</td>

<td>${product.description}</td>

<td>$${product.price}</td>

<td><a href="addProductView?code=${product.code}&description=${product.description}&price=${product.price}">Edit</a></td>

<td><a href="deleteProductView?code=${product.code}&description=${product.description}&price=${product.price}">Delete</a></td>

</tr>

</c:forEach>

</table>

<button type="button"><a style="text-decoration: none" href="addProductView">Add Product</a></button>

</div>

</body>

</html>

1. **delete.jsp**

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

<!DOCTYPE html>

<html>

<head>

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

<title>Product Maintenance</title>

<style>

body{

margin: 50px;

}

h1{

color: blue;

}

a{

text-decoration: none

}

</style>

</head>

<body>

<h1>Are you sure you want to delete this product?</h1>

<table style="width: 50%; collapse; background-color: white">

<tr>

<td><b>Code: </b></td>

<td>${product.code}</td>

</tr>

<tr>

<td><b>Description: </b></td>

<td>${product.description}</td>

</tr>

<tr>

<td><b>Price: </b></td>

<td>$${product.price}</td>

</tr>

<tr>

<td><button>

<a href="deleteProduct?code=${product.code}&description=${product.description}&price=${product.price}">Yes</a>

</button></td>

<td><button>

<a href="displayProducts">No</a>

</button></td>

</tr>

</table>

</body>

</html>

## Servlets

1. **Product.java**

package music.business;

import java.io.Serializable;

import java.text.NumberFormat;

import javax.persistence.Entity;

import javax.persistence.Id;

@Entity

public class Product implements Serializable {

/\*\*

\*

\*/

private static final long serialVersionUID = 1L;

private Long productId;

@Id

private String code;

private String description;

private double price;

public Product() {

}

public Long getId() {

return productId;

}

public void setId(Long productId) {

this.productId = productId;

}

public void setCode(String code) {

this.code = code;

}

public String getCode() {

return code;

}

public void setDescription(String description) {

this.description = description;

}

public String getDescription() {

return description;

}

public String getArtistName() {

String artistName = description.substring(0, description.indexOf(" - "));

return artistName;

}

public String getAlbumName() {

String albumName = description.substring(description.indexOf(" - ") + 3);

return albumName;

}

public void setPrice(double price) {

this.price = price;

}

public double getPrice() {

return price;

}

public String getPriceCurrencyFormat() {

NumberFormat currency = NumberFormat.getCurrencyInstance();

return currency.format(price);

}

public String getImageURL() {

String imageURL = "/musicStore/images/" + code + "\_cover.jpg";

return imageURL;

}

public String getProductType() {

return "Audio CD";

}

}

1. **ProductDB.java**

package music.data;

import java.util.ArrayList;

import java.util.List;

import javax.persistence.EntityManager;

import javax.persistence.EntityManagerFactory;

import javax.persistence.Persistence;

import javax.persistence.Query;

import music.business.Product;

public class ProductDB {

private static List<Product> products = null;

private static final String classNAME = "product";

private static EntityManagerFactory entityManagerFactory = Persistence.createEntityManagerFactory(classNAME);

private ProductDB() {

}

@SuppressWarnings("unchecked")

public static List<Product> selectProducts() {

products = new ArrayList<Product>();

EntityManager entityManager = entityManagerFactory.createEntityManager();

try {

Query query = entityManager.createQuery("select p from Product p");

products = query.getResultList();

entityManager.close();

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return products;

}

public static Product selectProduct(String productCode) {

products = selectProducts();

for (Product p : products) {

if (productCode != null && productCode.equalsIgnoreCase(p.getCode())) {

return p;

}

}

return null;

}

public static boolean exists(String productCode) {

if (productCode != null) {

EntityManager entityManager = entityManagerFactory.createEntityManager();

entityManager.getTransaction().begin();

Product p = entityManager.find(Product.class, productCode);

entityManager.getTransaction().commit();

if (p != null && productCode.equalsIgnoreCase(p.getCode())) {

return true;

} else

return false;

}

return false;

}

public static void insertProduct(Product product) {

try {

EntityManager entityManager = entityManagerFactory.createEntityManager();

entityManager.getTransaction().begin();

entityManager.persist(product);

entityManager.getTransaction().commit();

entityManager.close();

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

public static void updateProduct(Product product) {

try {

EntityManager entityManager = entityManagerFactory.createEntityManager();

entityManager.getTransaction().begin();

Product p = entityManager.find(Product.class, product.getCode());

p.setDescription(product.getDescription());

p.setPrice(product.getPrice());

entityManager.getTransaction().commit();

entityManager.close();

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

public static void deleteProduct(Product product) {

try {

EntityManager entityManager = entityManagerFactory.createEntityManager();

entityManager.getTransaction().begin();

Product p = entityManager.find(Product.class, product.getCode());

entityManager.remove(p);

entityManager.getTransaction().commit();

entityManager.close();

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

1. **SaveProduct.java**

package music.controller;

import java.io.IOException;

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 music.business.Product;

import music.data.ProductDB;

/\*\*

\* Servlet implementation class SaveProduct

\*/

@WebServlet("/saveProduct")

public class SaveProduct extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public SaveProduct() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

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

// TODO Auto-generated method stub

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

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

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

Product p = new Product();

p.setCode(code);

p.setDescription(description);

p.setPrice(Double.valueOf(price));

//Saving new product

if(ProductDB.exists(code))

ProductDB.updateProduct(p);

else

ProductDB.insertProduct(p);

//Displaying products

List<Product> productList = ProductDB.selectProducts();

request.setAttribute("productList", productList);

RequestDispatcher rd = request.getRequestDispatcher("/WEB-INF/jsp/displayProducts.jsp");

rd.forward(request, response);

}

/\*\*

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

\*/

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

// TODO Auto-generated method stub

doGet(request, response);

}

}

1. **DisplayProducts.java**

package music.controller;

import java.io.IOException;

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 music.business.Product;

import music.data.ProductDB;

/\*\*

\* Servlet implementation class DisplayProducts

\*/

@WebServlet("/displayProducts")

public class DisplayProducts extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public DisplayProducts() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse

\* response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

// TODO Auto-generated method stub

/\*String filePath = request.getServletContext().getRealPath("/WEB-INF/products.txt");

ProductIO.init(filePath);\*/

List<Product> productList = ProductDB.selectProducts();

request.setAttribute("productList", productList);

RequestDispatcher rd = request.getRequestDispatcher("/WEB-INF/jsp/displayProducts.jsp");

rd.forward(request, response);

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse

\* response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

// TODO Auto-generated method stub

doGet(request, response);

}

}

1. **DeleteProductView.java**

package music.controller;

import java.io.IOException;

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 music.business.Product;

/\*\*

\* Servlet implementation class DeleteProductView

\*/

@WebServlet("/deleteProductView")

public class DeleteProductView extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public DeleteProductView() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

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

// TODO Auto-generated method stub

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

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

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

Product p = new Product();

p.setCode(code);

p.setDescription(description);

p.setPrice(Double.valueOf(price));

request.setAttribute("product", p);

RequestDispatcher rd = request.getRequestDispatcher("/WEB-INF/jsp/delete.jsp");

rd.forward(request, response);

}

/\*\*

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

\*/

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

// TODO Auto-generated method stub

doGet(request, response);

}

}

1. **DeleteProduct.java**

package music.controller;

import java.io.IOException;

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 music.business.Product;

import music.data.ProductDB;

/\*\*

\* Servlet implementation class DeleteProduct

\*/

@WebServlet("/deleteProduct")

public class DeleteProduct extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public DeleteProduct() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

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

// TODO Auto-generated method stub

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

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

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

Product p = new Product();

p.setCode(code);

p.setDescription(description);

p.setPrice(Double.valueOf(price));

if(ProductDB.exists(code))

ProductDB.deleteProduct(p);

//Displaying products

List<Product> productList = ProductDB.selectProducts();

request.setAttribute("productList", productList);

RequestDispatcher rd = request.getRequestDispatcher("/WEB-INF/jsp/displayProducts.jsp");

rd.forward(request, response);

}

/\*\*

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

\*/

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

// TODO Auto-generated method stub

doGet(request, response);

}

}

1. **AddProductView.java**

package music.controller;

import java.io.IOException;

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 music.business.Product;

/\*\*

\* Servlet implementation class AddProductView

\*/

@WebServlet("/addProductView")

public class AddProductView extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public AddProductView() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

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

// TODO Auto-generated method stub

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

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

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

Product p = new Product();

if(code == null)

p.setCode("");

else

p.setCode(code);

if(description == null)

p.setDescription("");

else

p.setDescription(description);

if(price == null)

p.setPrice(0);

else

p.setPrice(Double.valueOf(price));

request.setAttribute("product", p);

RequestDispatcher rd = request.getRequestDispatcher("/WEB-INF/jsp/product.jsp");

rd.forward(request, response);

}

/\*\*

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

\*/

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

// TODO Auto-generated method stub

doGet(request, response);

}

}

## XMLS

1. **Web.xml**

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

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xmlns=*"http://java.sun.com/xml/ns/javaee"*

xsi:schemaLocation=*"http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd"*

id=*"WebApp\_ID"* version=*"3.0"*>

<display-name>productMaint1</display-name>

<welcome-file-list>

<welcome-file>index.html</welcome-file>

<welcome-file>index.htm</welcome-file>

<welcome-file>index.jsp</welcome-file>

<welcome-file>default.html</welcome-file>

<welcome-file>default.htm</welcome-file>

<welcome-file>default.jsp</welcome-file>

</welcome-file-list>

</web-app>

1. **Persistence.xml**

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

<persistence version="2.1"

xmlns="http://xmlns.jcp.org/xml/ns/persistence" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence http://xmlns.jcp.org/xml/ns/persistence/persistence\_2\_1.xsd">

<persistence-unit name="product" transaction-type="RESOURCE\_LOCAL">

<provider>org.eclipse.persistence.jpa.PersistenceProvider</provider>

<class>music.business.Product</class>

<properties>

<property name="javax.persistence.jdbc.driver" value="com.mysql.cj.jdbc.Driver" />

<property name="javax.persistence.jdbc.url" value="jdbc:mysql://localhost:3306/mydb" />

<property name="javax.persistence.jdbc.user" value="root" />

<property name="javax.persistence.jdbc.password" value="root" />

<!-- EclipseLink should create the database schema automatically -->

<!-- <property name="eclipselink.ddl-generation" value="create-tables"

/> -->

<property name="eclipselink.ddl-generation.output-mode"

value="database" />

</properties>

</persistence-unit>

</persistence>