//Bernice Templeman

context.xml

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

<Context path="/MusicStoreProductMaint">

<!-- maxActive: Maximum number of dB connections in pool. Make sure you

configure your mysqld max\_connections large enough to handle

all of your db connections. Set to 0 for no limit.

-->

<!-- maxIdle: Maximum number of idle dB connections to retain in pool.

Set to -1 for no limit. See also the DBCP documentation on this

and the minEvictableIdleTimeMillis configuration parameter.

-->

<!-- maxWait: Maximum time to wait for a dB connection to become available

in ms, in this example 10 seconds. An Exception is thrown if

this timeout is exceeded. Set to -1 to wait indefinitely.

-->

<!-- username and password: MySQL dB username and password for dB connections -->

<!-- driverClassName: Class name for the old mm.mysql JDBC driver is

org.gjt.mm.mysql.Driver - we recommend using Connector/J though.

Class name for the official MySQL Connector/J driver is com.mysql.jdbc.Driver.

-->

<!-- url: The JDBC connection url for connecting to your MySQL dB.

The autoReconnect=true argument to the url makes sure that the

mm.mysql JDBC Driver will automatically reconnect if mysqld closed the

connection. mysqld by default closes idle connections after 8 hours.

-->

<!-- logAbandoned, removeAbandoned, removeAbandonedTimeout: Control whether

the container will prevent connection pool leaks by automatically removing

Connection, Statement, ResultSet, etc. objects that haven't been closed

-->

<Resource auth="Container" driverClassName="com.mysql.jdbc.Driver" logAbandoned="true" maxActive="100" maxIdle="30" maxWait="10000" name="jdbc/music\_jpa" password="sesame" removeAbandoned="true" removeAbandonedTimeout="60" type="javax.sql.DataSource" url="jdbc:mysql://localhost:3306/music" username="root"/>

<Realm className="org.apache.catalina.realm.JDBCRealm" connectionName="root" connectionPassword="sesame" connectionURL="jdbc:mysql://localhost:3306/music\_jpa" debug="99" driverName="com.mysql.jdbc.Driver" roleNameCol="Rolename" userCredCol="Password" userNameCol="Username" userRoleTable="UserRole" userTable="UserPass"/>

</Context>

-------------

web.xml

--------------

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

<web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_3\_1.xsd">

<display-name>Bernice Blogs: Technology Store site</display-name>

<description>Technology </description>

<!-- define a context initialization parameter -->

<context-param>

<param-name>custServEmail</param-name>

<param-value>custserv@freshcornrecords.com</param-value>

</context-param>

<!-- servlet definitions (alphabetical by complete class name) -->

<servlet>

<servlet-name>AdminController</servlet-name>

<servlet-class>music.controllers.AdminController</servlet-class>

</servlet>

<servlet>

<servlet-name>CatalogController</servlet-name>

<servlet-class>music.controllers.CatalogController</servlet-class>

</servlet>

<servlet>

<servlet-name>OrderController</servlet-name>

<servlet-class>music.controllers.OrderController</servlet-class>

</servlet>

<servlet>

<servlet-name>ProductController</servlet-name>

<servlet-class>music.admin.ProductController</servlet-class>

</servlet>

<servlet>

<servlet-name>UserController</servlet-name>

<servlet-class>music.controllers.UserController</servlet-class>

</servlet>

<servlet>

<servlet-name>UserMaintServlet</servlet-name>

<servlet-class>music.admin.UserMaintServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>UserMaintServlet</servlet-name>

<url-pattern>/admin/userMaint</url-pattern>

</servlet-mapping>

<!-- servlet mappings -->

<servlet-mapping>

<servlet-name>ProductController</servlet-name>

<url-pattern>/admin/prodMaint/\*</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>AdminController</servlet-name>

<url-pattern>/adminController/\*</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>CatalogController</servlet-name>

<url-pattern>/catalog/product/\*</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>OrderController</servlet-name>

<url-pattern>/order/\*</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>UserController</servlet-name>

<url-pattern>/user/\*</url-pattern>

</servlet-mapping>

<!-- define a servlet context listener -->

<listener>

<listener-class>music.util.MusicStoreContextListener</listener-class>

</listener>

<!-- define the welcome files -->

<welcome-file-list>

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

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

</welcome-file-list>

<!-- Enable the custom error pages -->

<error-page>

<error-code>404</error-code>

<location>/error\_404.jsp</location>

</error-page>

<error-page>

<exception-type>java.lang.Throwable</exception-type>

<location>/error\_java.jsp</location>

</error-page>

<!-- Define two security roles -->

<security-role>

<description>customer service employees</description>

<role-name>service</role-name>

</security-role>

<security-role>

<description>programmers</description>

<role-name>programmer</role-name>

</security-role>

<security-constraint>

<!-- Restrict access to the URLs in the admin directory -->

<web-resource-collection>

<web-resource-name>Protected Area</web-resource-name>

<url-pattern>/admin/\*</url-pattern>

</web-resource-collection>

<!-- Authorize the service and programmer roles -->

<auth-constraint>

<role-name>service</role-name>

<role-name>programmer</role-name>

</auth-constraint>

<!-- Forces a secure connection - remove comments to enable -->

<user-data-constraint>

<transport-guarantee>CONFIDENTIAL</transport-guarantee>

</user-data-constraint>

</security-constraint>

<!-- Use form-based authentication to provide access -->

<login-config>

<auth-method>FORM</auth-method>

<form-login-config>

<form-login-page>/login.jsp</form-login-page>

<form-error-page>/login\_error.jsp</form-error-page>

</form-login-config>

</login-config>

<!-- Use cookies (not URL encoding) for session tracking -->

<session-config>

<tracking-mode>COOKIE</tracking-mode>

</session-config>

</web-app>

.------------------

delete\_cookies.jsp

<jsp:include page="/includes/header.jsp" />

<!-- start the middle column -->

<section>

----------------------------------

<h1>Cookies</h1>

<p>All persistent cookies for this server have been removed from this browser.</p>

</section>

<!-- end the middle column -->

<jsp:include page="/includes/footer.jsp" />

------------------------------------

error\_404.jsp

<jsp:include page="/includes/header.jsp" />

<!-- start the middle column -->

<%@ page isErrorPage="true" %>

<section>

<h1>404 Error</h1>

<p>The server was not able to find the file you requested.</p>

<p>To continue, click the Back button or select a link from this page.</p>

<h2>Details</h2>

<p>Requested URI: ${pageContext.errorData.requestURI}</p>

</section>

<!-- end the middle column -->

<jsp:include page="/includes/footer.jsp" />

----------------------------

error\_java.jsp

-------------

<jsp:include page="/includes/header.jsp" />

<!-- start the middle column -->

<%@ page isErrorPage="true" %>

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

<section>

<h1>Java Error</h1>

<p>Sorry, Java has thrown an exception.</p>

<p>To continue, click the back button or select a link from this page.</p>

<!-- NOTE: When you're done with testing, you can disable

any of the following details about the exception. -->

<h2>Details</h2>

<code>

${pageContext.errorData.servletName} threw a <br>

${pageContext.exception}<br>

<c:forEach var="line" items="${pageContext.errorData.throwable.stackTrace}">

&nbsp;&nbsp;&nbsp;&nbsp;at ${line}<br>

</c:forEach>

</code>

</section>

<!-- end the middle column -->

<jsp:include page="/includes/footer.jsp" />

----------------------------

error-sql.jsp

-----------

<jsp:include page="/includes/header.jsp" />

<!-- start the middle column -->

<%@ page isErrorPage="true" %>

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

<section>

<h1>Database error</h1>

<p>You have encountered an error with the Music Store database.

You may need to install this database as described in appendix A.</p>

<p>To continue, click the back button or select a link from this page.</p>

<!-- NOTE: When you're done with testing, you can disable

any of the following details about the exception. -->

<h2>Details</h2>

<code>

${pageContext.errorData.servletName} threw a <br>

${pageContext.exception}<br>

<c:forEach var="line" items="${pageContext.errorData.throwable.stackTrace}">

&nbsp;&nbsp;&nbsp;&nbsp;at ${line}<br>

</c:forEach>

</code>

</section>

<!-- end the middle column -->

<jsp:include page="/includes/footer.jsp" />

------------------

index-userAdmin.jsp

---------------

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>Murach's Java Servlets and JSP</title>

<link rel="stylesheet" href="styles/main.css" type="text/css"/>

</head>

<body>

<h1>Users</h1>

<table>

<tr>

<th>First Name</th>

<th>Last Name</th>

<th colspan="3">Email</th>

</tr>

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

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

<tr>

<td>${user.firstName}</td>

<td>${user.lastName}</td>

<td>${user.email}</td>

<td><a href="userAdmin?action=display\_user&amp;email=${user.email}">Update</a></td>

<td><a href="userAdmin?action=delete\_user&amp;email=${user.email}">Delete</a></td>

</tr>

</c:forEach>

</table>

<p><a href="userAdmin">Refresh</a></p>

</body>

</html>

-------------------------

index.jsp

=---------------

<jsp:include page="/includes/header.jsp" />

<!-- start the middle column -->

<section>

<h1>Welcome to The Bernice Blogs Tech Store!</h1>

<p>Thanks for visiting. Make yourself at home. Feel free to browse through

our catalog.</p>

<p>If you find an topic that you like, we hope that you'll use this site

to order it. Most of the products we carry aren't available anywhere else!</p>

</section>

<!-- end the middle column -->

<jsp:include page="/includes/column\_right\_news.jsp" />

<jsp:include page="/includes/footer.jsp" />

---------------------------------

login.jsp

<jsp:include page="/includes/header.jsp" />

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

<!-- start the middle column -->

<section>

<h1>Login Form</h1>

<p>Please enter a username and password to continue.</p>

<form action="j\_security\_check" method="post">

<label>Username</label>

<input type="text" name="j\_username"><br>

<label>Password</label>

<input type="password" name="j\_password"><br>

<label>&nbsp;</label>

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

</form>

</section>

<!-- end the middle column -->

<jsp:include page="/includes/footer.jsp" />

------------------------------------

login\_error.jsp

<jsp:include page="/includes/header.jsp" />

<!-- start the middle column -->

<section>

<h1>Login Form - Error</h1>

<p>You did not log in successfully.</p>

<p>Please check your username and password and try again.</p>

<form action="j\_security\_check" method="get">

<label>Username</label>

<input type="text" name="j\_username"><br>

<label>Password</label>

<input type="password" name="j\_password"><br>

<label>&nbsp;</label>

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

</form>

</section>

<!-- end the middle column -->

<

<jsp:include page="/includes/footer.jsp" />

----------------------------

adduser.jsp

------------

<jsp:include page="/includes/header\_all\_absolute.jsp" />

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

<%@ taglib prefix="bt" uri="/WEB-INF/tlds/music.tld" %>

<%@page contentType="text/html" pageEncoding="windows-1252"%>

<section>

<h1>Customer</h1>

<p><bt:ifEmptyMark color="blue" field=""/> marks required fields</p>

<p><i>${message}</i></p>

<form action="userMaint" method="post">

<label class="block">Email: </label>

<input type="text" name="email" value="<c:out value="${user.email}" />" />

<bt:ifEmptyMark color="blue" field="${user.email}"/><br>

<label class="block">First Name: </label>

<input type="text" name="firstName" size="50" value="<c:out value="${user.firstName}" />" />

<bt:ifEmptyMark color="blue" field="${user.firstName}"/><br>

<label class="block">Last Name:</label>

<input type="text" name="lastName" value="<c:out value="${user.lastName}" />" />

<bt:ifEmptyMark color="blue" field="${user.lastName}"/><br>

<table style="width:10px;">

<tr>

<td>

<form action="userMaint" method="post">

<input type="hidden" name="action" value="addUserConfirmed">

<input type="submit" value="Update User">

</form>

</td>

<td>

<form action="userMaint" method="post">

<input type="hidden" name="action" value="displayUsers">

<input type="submit" value="View Users">

</form>

</td>

</tr>

</table>

</form>

</section>

<!-- end middle column -->

<jsp:include page="/includes/footer.jsp" />

-----------------------

deleteUser

-------------

<%--

Document : deleteUser

Created on : Apr 18, 2016, 1:33:57 PM

Author : Bernice

--%>

<jsp:include page="/includes/header\_all\_absolute.jsp" />

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

<%@page contentType="text/html" pageEncoding="windows-1252"%>

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

<table >

<tr>

<td><label>Email:</label></td>

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

</tr>

<tr>

<td><label>First Name:</label></td>

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

</tr>

<tr>

<td><label>Last Name:</label></td>

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

</tr>

</table>

<table style="width:10px;">

<tr>

<td>

<form action="userMaint" method="post">

<input type="hidden" name="action" value="deleteUserConfirmed">

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

</form>

</td>

<td>

<form action="userMaint" method="post">

<input type="hidden" name="action" value="displayUsers">

<input type="submit" value="No">

</form>

</td>

</tr>

</table>

<jsp:include page="/includes/footer.jsp" />

----------

index.jsp

-------------------

<jsp:include page="/includes/header\_all\_absolute.jsp" />

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

<!-- begin middle column -->

<section id="admin">

<h1>Admin Menu</h1>

<!-- these Form tags don't force a secure connection -->

<form action="<c:url value='/adminController/displayInvoices'/>" method="post">

<input type="submit" value="Process Invoices" class="left\_margin">

</form>

<form action="reports.jsp" method="post">

<input type="submit" value="Display Reports" class="left\_margin">

</form>

<form action="prodMaint/prodList" method="post">

<input type="submit" value="Product Maintenance" class="left\_margin">

</form>

<form action="<c:url value='/admin/userMaint?action=displayUsers' />" method="post">

<input type="submit" value="Maintain Users" class="left\_margin">

</form>

</section>

<!-- end middle column -->

<jsp:include page="/includes/footer.jsp" />

-------------------

invoice.jsp

-------------------

<jsp:include page="/includes/header\_all\_absolute.jsp" />

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

<!-- begin middle column -->

<section id="admin">

<h1>Your invoice</h1>

<table>

<tr>

<td><b>Date</b></td>

<td>${invoice.invoiceDate}</td>

<td></td>

</tr>

<tr>

<td><b>Ship To</b></td>

<td>${invoice.user.addressHTMLFormat}</td>

<td></td>

</tr>

<tr><td colspan="3"><hr></td></tr>

<tr>

<td><b>Qty</b></td>

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

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

</tr>

<c:forEach var="item" items="${invoice.lineItems}">

<tr>

<td>${item.quantity}</td>

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

<td>${item.totalCurrencyFormat}</td>

</tr>

</c:forEach>

<tr><td colspan="3"><hr></td></tr>

<tr>

<td><b>Total</b></td>

<td></td>

<td><p>${invoice.invoiceTotalCurrencyFormat}</td>

</tr>

</table>

<label>Payment information</label>

<span>${invoice.user.creditCardType}: ${invoice.user.creditCardNumber}

(${invoice.user.creditCardExpirationDate})</span><br>

<label>Email Address</label>

<span>${invoice.user.email}</span><br>

<form action="processInvoice" method="post">

<input type="submit" value="Process Invoice">

</form>

<form action="displayInvoices" method="post">

<input type="submit" value="View Unprocessed Invoices">

</form>

</section>

<!-- end middle column -->

<jsp:include page="/includes/footer.jsp" />

---------------------------

invoices.jsp

-------------------

<jsp:include page="/includes/header\_all\_absolute.jsp" />

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

<!-- begin middle column -->

<section id="admin">

<h1>Invoices to be processed:</h1>

<c:if test="${unprocessedInvoices == null}">

<p>There are no invoices to process.</p>

</c:if>

<c:if test="${unprocessedInvoices != null}">

<table>

<tr>

<td></td>

<td><b>Customer Name</b></td>

<td><b>Invoice Date</b></td>

</tr>

<c:forEach var="invoice" items="${unprocessedInvoices}">

<tr>

<td>

<a href="displayInvoice?invoiceNumber=${invoice.invoiceNumber}">Click to View</a>

</td>

<td>${invoice.user.firstName} ${invoice.user.lastName}</td>

<td>${invoice.invoiceDateDefaultFormat}</td>

</tr>

</c:forEach>

</table>

</c:if>

<form action="<c:url value='/admin'/>" method="post">

<input type=submit value="Go Back to Menu">

</form>

</section>

<!-- end middle column -->

<jsp:include page="/includes/footer.jsp" />

---------

parameters.jsp

------------

<jsp:include page="/includes/header\_all\_absolute.jsp" />

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

<!-- begin middle column -->

<section>

<h1>${param.reportTitle}</h1>

<p>Please enter dates using this format: yyyy-mm-dd.</p>

<form action="<c:url value='/adminController/displayReport'/>" method="post">

<!-- Note the use of hidden fields to pass parameters -->

<input type="hidden" name="reportName" value="${param.reportName}">

<input type="hidden" name="reportTitle" value="${param.reportTitle}">

<label>Start Date</label>

<input type="text" size="10" maxlength=10

name="startDate" value="${currentYear}-01-01" required><br>

<label>End Date</label>

<input type="text" size="10" maxlength=10

name="endDate" value="${currentYear}-12-31" required><br>

<label>&nbsp;</label>

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

</form>

</section>

<!-- end middle column -->

<jsp:include page="/includes/footer.jsp" />

=----------------------

prodAdd.jsp

---------------

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

<%@ taglib prefix="bt" uri="/WEB-INF/tlds/music.tld" %>

<jsp:include page="/includes/header\_all\_absolute.jsp" />

<section>

<h1>Product</h1>

<p><bt:ifEmptyMark color="blue" field=""/> marks required fields</p>

<p><i>${message}</i></p>

<form action="addConfirmed" method="post">

<label for="prodCode">Code:</label>

<input type="text" name="prodCode" value="<c:out value="${product.code}" />" />

<bt:ifEmptyMark field="${product.code}"/><br />

<label for="prodDescription">Description:</label>

<input type="text" name="prodDescription" size="50" value="<c:out value="${product.description}" />" />

<bt:ifEmptyMark field="${product.description}"/><br />

<c:choose>

<c:when test="${product.price=='0.0'}">

<label for="prodPrice">Price:</label>

<input type="text" name="prodPrice" value="<c:out value="${product.price}" />" />

<bt:ifEmptyMark field="${product.price}"/><br /><br />

</c:when>

<c:otherwise>

<label for="prodPrice">Price:</label>

<input type="text" name="prodPrice" value="<c:out value="${product.price}" />" />

<bt:ifEmptyMark color="blue" field="${product.price}"/><br>

</c:otherwise>

</c:choose>

<br/>

<table>

<tr>

<td>

<form action="addConfirmed" method="post">

<input type="submit" value="Update Product">

</form>

</td>

<td>

<form action="prodList" method="post">

<input type="submit" value="View Product">

</form>

</td>

</tr>

</table>

</section>

<jsp:include page="/includes/footer.jsp" />

--------------------------------

prodDelete.jsp

------------------------

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

<jsp:include page="/includes/header\_all\_absolute.jsp" />

<section>

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

<table class="noBorder noPad">

<tr>

<td><label>Code:</label></td>

<td><c:out value="${product.code}" /></td>

</tr>

<tr>

<td><label>Description:</label></td>

<td><c:out value="${product.description}" /></td>

</tr>

<tr>

<td><label>Price:</label></td>

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

</tr>

</table>

<table>

<tr>

<td>

<form action="deleteConfirmed" method="post">

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

</form>

</td>

<td>

<form action="prodList" method="post">

<input type="submit" value="No">

</form>

</td>

</tr>

</table>

</section>

<jsp:include page="/includes/footer.jsp" />

-------------------------------------

prodlist.jsp

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

<jsp:include page="/includes/header\_all\_absolute.jsp" />

<section>

<h1>Products</h1>

<table class="border">

<tr>

<th>Code</th>

<th>Description</th>

<th class="right">Price</th>

<th>&nbsp;</th>

<th>&nbsp;</th>

</tr>

<c:choose>

<c:when test="${products != '[]'}">

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

<tr>

<td><c:out value="${product.code}" /></td>

<td><c:out value="${product.description}" /></td>

<td class="right"><c:out value="${product.priceCurrencyFormat}" /></td>

<td><a href="addProduct?productCode=${product.code}">Edit</a></td>

<td><a href="deleteProduct?productCode=${product.code}">Delete</a></td>

</tr>

</c:forEach>

</c:when>

<c:otherwise>

<tr>

<td colspan="5">There are no products found.</td>

</tr>

</c:otherwise>

</c:choose>

</table><br />

<form action="addProduct" method="post">

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

</form>

</section>

<jsp:include page="/includes/footer.jsp" />

------------------------

reports.jsp

\_\_----------------

<jsp:include page="/includes/header\_all\_absolute.jsp" />

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

<!-- begin middle column -->

<section id="admin">

<h1>Report list</h1>

<p>This page contains a list of the reports that are

available from this site.</p>

<form action="<c:url value='/adminController/displayReport'/>" method="post">

<input type="hidden" name="reportName" value="userEmail">

<input type="hidden" name="reportTitle" value="The User Email report">

<input type="submit" value="User Email Report" class="left\_margin">

</form>

<form action="parameters.jsp" method="post">

<input type="hidden" name="reportName" value="downloadDetail">

<input type="hidden" name="reportTitle" value="The Downloads report">

<input type="submit" value="Downloads report" class="left\_margin">

</form>

</section>

<!-- end middle column -->

<jsp:include page="/includes/footer.jsp" />

------------------------

users.jsp

<%--

Document : users

Created on : Apr 18, 2016, 9:39:51 AM

Author : Bernice

--%>

<jsp:include page="/includes/header\_all\_absolute.jsp" />

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

<h1>Customers</h1>

<table >

<tr>

<th>Email</th>

<th>First Name</th>

<th class="right">Last Name</th>

<th>&nbsp;</th>

<th>&nbsp;</th>

</tr>

<c:choose>

<c:when test="${users != null}">

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

<tr>

<td><c:out value='${user.email}'/></td>

<td><c:out value='${user.firstName}'/></td>

<td class="right">${user.lastName}</td>

<td><form action="userMaint" method="post">

<input type="hidden" name="email" value="<c:out value='${user.email}'/>">

<a href="userMaint?action=addUser&email=${user.email}">Edit</a>

</form></td>

<td><a href="userMaint?action=deleteUser&email=${user.email}">Delete</a></td>

</tr>

</c:forEach>

</c:when>

<c:otherwise>

<tr>

<td colspan="5">There are no users found.</td>

</tr>

</c:otherwise>

</c:choose>

</table>

&nbsp;

<form action="userMaint" method="post">

<input type="hidden" name="action" value="addUser" />

<input type="submit" value="Add User" />

</form>

<!-- end middle column -->

<jsp:include page="/includes/footer.jsp" />

--------------------------

cart.jsp

<jsp:include page="/includes/header.jsp" />

<!-- begin middle column -->

<section class="cart">

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

<h1>Your cart</h1>

<c:choose>

<c:when test="${emptyCart != null}">

<p>Your cart is empty.</p>

</c:when>

<c:otherwise>

<table>

<tr>

<th>Qty</th>

<th>Description</th>

<th>Price</th>

<th>Amount</th>

<th>&nbsp;</th>

</tr>

<c:forEach var="item" items="${cart.items}">

<tr class="cart\_row">

<td>

<form action="<c:url value='/order/updateItem'/>" method="post">

<input type="hidden" name="productCode"

value="<c:out value='${item.product.code}'/>">

<input type=text name="quantity"

value="<c:out value='${item.quantity}'/>" id="quantity">

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

</form>

</td>

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

<td>${item.product.priceCurrencyFormat}</td>

<td>${item.totalCurrencyFormat}</td>

<td>

<form action="<c:url value='/order/removeItem'/>" method="post">

<input type="hidden" name="productCode"

value="<c:out value='${item.product.code}'/>">

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

</form>

</td>

</tr>

</c:forEach>

<tr>

<td colspan="2">

<p><b>To change the quantity for an item</b>, enter the new quantity

and click on the Update button.</p>

<p><b>To remove an item</b>, click on the Remove button.</p>

</td>

<td colspan="3">&nbsp;</td>

</tr>

</table>

</c:otherwise>

</c:choose>

<form action="<c:url value='/catalog'/>" method="get" id="float\_left">

<input type="submit" value="Continue Shopping">

</form>

<c:if test="${emptyCart == null}">

<!-- Connection is NOT SECURE. For testing only. -->

<form action="<c:url value='/order/checkUser'/>" method="post">

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

</form>

<!-- Connection is SECURE. Before you can use it, you need to configure

a secure connection on your system as described in chapter 15, comment

out the previous form, and remove the comments from the following form. -->

<!--

<form action="${absolutePathSecure}/order/checkUser" method="post">

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

</form>

-->

</c:if>

</section>

<!-- end middle column -->

<jsp:include page="/includes/footer.jsp" />

-----------------------

complete.jsp

----------------

<jsp:include page="/includes/header\_all\_absolute.jsp" />

<%-- Use the following left column instead after you configure a

secure connection as described in chapter 15.

<jsp:include page="/includes/column\_left\_all\_absolute.jsp" />

--%>

<!-- begin middle column -->

<section>

<h1>Thank you, ${user.firstName}</h1>

<!-- store email address as a global variable and use EL to display it -->

<p>Your order has been submitted. We'll begin processing your

order right away. If you have any questions about your order,

please feel free to contact us at

<a href="mailto:${custServEmail}">${custServEmail}</a></p>

</section>

<!-- end middle column -->

<jsp:include page="/includes/column\_right\_news.jsp" />

<jsp:include page="/includes/footer.jsp" />

------------------------

credit\_card.jsp

<jsp:include page="/includes/header\_all\_absolute.jsp" />

<%-- Use the following left column instead after you configure a

secure connection as described in chapter 15.

<jsp:include page="/includes/column\_left\_all\_absolute.jsp" />

--%>

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

<!-- begin middle column -->

<section class="cart">

<h1>Enter your credit card information</h1>

<form action="<c:url value='/order/completeOrder' />" method="post">

<table>

<tr>

<td><b>Credit card type</b></td>

<td>

<select name="creditCardType" size="1">

<option value="Visa">Visa</option>

<option value="Mastercard">Mastercard</option>

<option value="AmEx">American Express</option>

</select>

</td>

</tr>

<tr>

<td><b>Card number</b></td>

<td><input type="text" size="20" name="creditCardNumber"

maxlength="25" required></td>

</tr>

<tr>

<td><b>Expiration date (mm/yyyy)</b></td>

<td><select name="creditCardExpirationMonth">

<option value="01">01</option>

<option value="02">02</option>

<option value="03">03</option>

<option value="04">04</option>

<option value="05">05</option>

<option value="06">06</option>

<option value="07">07</option>

<option value="08">08</option>

<option value="09">09</option>

<option value="10">10</option>

<option value="11">11</option>

<option value="12">12</option>

</select>

/

<select name="creditCardExpirationYear">

<c:forEach var="year" items="${creditCardYears}">

<option value="${year}">${year}</option>

</c:forEach>

</select>

</td>

</tr>

<tr>

<td></td>

<td><input type="submit" value="Submit Order"></td>

</tr>

</table>

</form>

</section>

<!-- end middle column -->

<jsp:include page="/includes/footer.jsp" />

--------------------------

invoice.jsp

---------------------

<jsp:include page="/includes/header\_all\_absolute.jsp" />

<%-- Use the following left column instead after you configure a

secure connection as described in chapter 15.

<jsp:include page="/includes/column\_left\_all\_absolute.jsp" />

--%>

<!-- begin middle column -->

<section class="cart">

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

<h1>Your invoice</h1>

<table>

<tr>

<th>Date</th>

<td>${invoice.invoiceDateDefaultFormat}</td>

<td></td>

</tr>

<tr>

<th class="top">Ship To</th>

<td>${user.addressHTMLFormat}</td>

<td></td>

</tr>

<tr>

<td colspan="3"><hr></td>

</tr>

<tr>

<th>Qty</th>

<th>Description</th>

<th>Price</th>

</tr>

<c:forEach var="item" items="${invoice.lineItems}">

<tr>

<td>${item.quantity}</td>

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

<td>${item.totalCurrencyFormat}</td>

</tr>

</c:forEach>

<tr>

<th>Total:</th>

<td></td>

<td>${invoice.invoiceTotalCurrencyFormat}</td>

</tr>

</table>

<form action="<c:url value='/order/displayUser' />" method="post" id="float\_left">

<input type="submit" value="Edit Address">

</form>

<form action="<c:url value='/order/displayCreditCard' />" method="post">

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

</form>

</section>

<!-- end middle column -->

<jsp:include page="/includes/footer.jsp" />

----------------------

user.jsp

----------------

<jsp:include page="/includes/header\_all\_absolute.jsp" />

<%-- Use the following left column instead after you configure a

secure connection as described in chapter 15.

<jsp:include page="/includes/column\_left\_all\_absolute.jsp" />

--%>

<!-- begin middle column -->

<section class="cart">

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

<h1>Enter your name and contact information</h1>

<form action="<c:url value='/order/processUser' />" method=post>

<p id="required">Required <span class="required">\*</span></p>

<label>First Name</label>

<input type="text" name="firstName" maxlength=20

value="${user.firstName}" required>

<p class="required">\*</p><br>

<label>Last Name</label>

<input type="text" name="lastName" value="${user.lastName}" required>

<p class="required">\*</p><br>

<label>Email Address</label>

<input type="email" name="email" value="${user.email}" required>

<p class="required">\*</p><br>

<label>Company</label>

<input type="text" name="companyName" value="${user.companyName}">

<p class="required">&nbsp;</p><br>

<label>Address1</label>

<input type="text" name="address1" value="${user.address1}" required>

<p class="required">\*</p><br>

<label>Address2</label>

<input type="text" name="address2" value="${user.address2}">

<p class="required">&nbsp;</p><br>

<label>City</label>

<input type="text" name="city" value="${user.city}" required>

<p class="required">\*</p><br>

<label>State</label>

<input type="text" name="state" value="${user.state}" required>

<p class="required">\*</p><br>

<label>Zip Code</label>

<input type="text" name="zip" value="${user.zip}" required>

<p class="required">\*</p><br>

<label>Country</label>

<input type="text" name="country" value="${user.country}" required>

<p class="required">\*</p><br>

<label>&nbsp;</label>

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

</form>

</section>

<!-- end middle column -->

<jsp:include page="/includes/footer.jsp" />

-------------------

8601 index.jsp

<jsp:include page="/includes/header.jsp" />

<!-- start the middle column -->

<section>

<jsp:include page="/includes/product\_table.jsp" />

<p class="clear\_both">

The debut album from 86 (the band), True Life Songs and Pictures rocks and

twangs in equal measure. Filled with banjo, one-string bass, fiddle, and

3-part harmonies, this semi-rock, semi-country, semi-bluegrass album covers

a lot of ground. "How to Get There" is a rambling epic that unleashes a

rapid-fire lyrical barrage while "Don't Close Your Eyes" and "Morning Sun"

are more melancholy and bittersweet.

</p>

</section>

<!-- end the middle column -->

<jsp:include page="/includes/column\_right\_buttons.jsp" />

<jsp:include page="/includes/footer.jsp" />

-------------------------

sound.jsp

--------------<jsp:include page="/includes/header.jsp" />

<!-- start the middle column -->

<section>

<jsp:include page="/includes/product\_table.jsp" />

<p class="clear\_both">To listen to a track, click on the song's link.</p>

<p>1. How to Get There</p>

<p>2. <a href="/musicStore/sound/8601/star.mp3">You Are a Star</a></p>

<p>3. <a href="/musicStore/sound/8601/no\_difference.mp3">Don't Make No Difference</a></p>

<p>4. Unidentified Fiddling Object</p>

<p>5. Beat Up Old Car</p>

<p>6. Wildflowers</p>

<p>7. What You Whistle When You Walk Home</p>

<p>8. Three Sheets to the Wind</p>

<p>9. Singin' Drunk</p>

<p>10. Don't Close Your Eyes</p>

<p>11. Morning Sun</p>

</section>

<!-- end the middle column -->

<jsp:include page="/includes/column\_right\_buttons.jsp" />

<jsp:include page="/includes/footer.jsp" />

------------------------

<jsp:include page="/includes/header.jsp" />

<!-- start the middle column -->

<section>

<jsp:include page="/includes/product\_table.jsp" />

<p class="clear\_both">

The debut album from Joe Rut rambles from Byrds-esque folk pop of "Filter" to

the country sounds of "Find My Way Marie" to psychedelic Brit-pop tunes like

"A Place In All This." This well-crafted album is unique and cohesive, revealing

its many layers on repeated listens.

</p>

</section>

<!-- end the middle column -->

<jsp:include page="/includes/column\_right\_buttons.jsp" />

<jsp:include page="/includes/footer.jsp" />

--------------------------------------

<jsp:include page="/includes/header.jsp" />

<!-- start the middle column -->

<section>

<jsp:include page="/includes/product\_table.jsp" />

<p class="clear\_both">To listen to a track, click on the song's link.</p>

<p>1. <a href="/musicStore/sound/jr01/filter.mp3">Filter</a></p>

<p>2. Find My Way Marie</p>

<p>3. Hole</p>

<p>4. 1400 Years</p>

<p>5. <a href="/musicStore/sound/jr01/so\_long.mp3">So Long Lazy Ray</a></p>

<p>6. A Tuna Is a Damn Big Fish</p>

<p>7. El Dorado</p>

<p>8. Dream of You</p>

<p>9. This Sea Is Full of Monsters</p>

<p>10. A Place in All This</p>

<p>11. GTTSWMD</p>

<p>12. AM Land</p>

<p>13. Whole Month of Sundays</p>

<p>14. Penny From a Poor Man</p>

</section>

<!-- end the middle column -->

<jsp:include page="/includes/column\_right\_buttons.jsp" />

<jsp:include page="/includes/footer.jsp" />

------------------------------

<jsp:include page="/includes/header.jsp" />

<!-- start the middle column -->

<section>

<jsp:include page="/includes/product\_table.jsp" />

<p class="clear\_both">

This 68-minute opus from San Francisco's Paddlefoot doesn't pull any

punches. Songs like "64 Corvair, Part 2", "Whiskey Before Breakfast",

and "The Murphy-Ryan Polkas" mix traditional American and Irish fiddle

tunes with indie rock. The result is somewhere between The Pogues, Camper

Van Beethoven, and Uncle Tupelo. The sincerity and quirkiness of other songs

like "When I Was Nine" and "Tiny House" are more reminiscent of Jonathan Richman.

</p>

</section>

<!-- end the middle column -->

<jsp:include page="/includes/column\_right\_buttons.jsp" />

<jsp:include page="/includes/footer.jsp" />

-------------------------

<jsp:include page="/includes/header.jsp" />

<!-- start the middle column -->

<section>

<jsp:include page="/includes/product\_table.jsp" />

<p class="clear\_both">To listen to a track, click on the song's link.</p>

<p>1. Pete and Jimmy</p>

<p>2. <a href="/musicStore/sound/pf01/whiskey.mp3">Whiskey Before Breakfast</a></p>

<p>3. Fishing Rod</p>

<p>4. The Murphy-Ryan Polkas</p>

<p>5. John Henry's Blues</p>

<p>6. <a href="/musicStore/sound/pf01/corvair.mp3">64 Corvair, Part 2</a></p>

<p>7. Racoons Like Moonshine, Too</p>

<p>8. Shelf Life</p>

<p>9. Hey Chris</p>

<p>10. Rock and Roll Scene</p>

<p>11. Tiny House</p>

<p>12. Strum Along Slow</p>

<p>13. She's Moving Back Home</p>

<p>14. When I Was Nine</p>

<p>15. Kangaroo's Paw</p>

<p>16. Sound of the Fiddle</p>

<p>17. Amatxi Smiled</p>

<p>18. Yeah</p>

<p>19. Dashboard Waltz</p>

</section>

<!-- end the middle column -->

<jsp:include page="/includes/column\_right\_buttons.jsp" />

<jsp:include page="/includes/footer.jsp" />

--------------------------------

<jsp:include page="/includes/header.jsp" />

<!-- start the middle column -->

<!-- If necessary, this page could be generated from the database. -->

<section>

<h1>The Bernice Blogs Tech Store Catalog</h1>

<h2>86 (the band)</h2>

<p><a href="product/8601">True Life Songs and Pictures</a></p>

<h2 class="top\_margin">Paddlefoot</h2>

<p><a href="product/pf01">Paddlefoot (the first album)</a></p>

<p><a href="product/pf02">Paddlefoot (the second album)</a></p>

<h2 class="top\_margin">Joe Rut</h2>

<p><a href="product/jr01">Genuine Wood Grained Finish</a></p>

</section>

<!-- end the middle column -->

<jsp:include page="/includes/column\_right\_news.jsp" flush="true" />

<jsp:include page="/includes/footer.jsp" />

----------------------

<jsp:include page="/includes/header.jsp" />

<!-- start the middle column -->

<section>

<h1>Download registration</h1>

<p>Before you can download and listen to these sound files,

you must register with us by entering your name and email

address below.</p>

<!-- Import the core JSTL library -->

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

<!-- Use the JSTL url tag to encode the URL -->

<form action="<c:url value='/catalog/product/${product.code}/listen/register'/>" method="post">

<label>Email</label>

<input type="email" name="email" required><br>

<label>First Name</label>

<input type="text" name="firstName" required><br>

<label>Last Name</label>

<input type="text" name="lastName" required><br>

<label>&nbsp;</label>

<input type="submit" value="Register" id="submit">

</form>

</section>

<!-- end the middle column -->

<jsp:include page="/includes/column\_right\_buttons.jsp" />

<jsp:include page="/includes/footer.jsp" />

----------

custermer service

------------------

<jsp:include page="/includes/header.jsp" />

<!-- start the middle column -->

<section>

<h1>Customer Service</h1>

<p>

We want to back our music albums up with the best possible service.

Whether you have a question about what albums we offer, need a printed

catalog, or need help with an order, just let us know. We'll do everything

we can to make it easy and enjoyable for you to do business with us.

</p>

<p>

If you have questions or comments, please contact us

in whatever way is most convenient for you. We look forward to hearing from you!

</p>

<h2>Phone:</h2>

<p>1-800-221-5528</p>

<h2>FAX:</h2>

<p>1-559-440-0963</p>

<h2>Email:</h2>

<p><a href="mailto:custserv@berniceblogs.com">custserv@berniceblogs.com</a></p>

</section>

<!-- end the middle column -->

<jsp:include page="/includes/column\_right\_news.jsp" />

<jsp:include page="/includes/footer.jsp" />

---------------------------

email

-------------------

<jsp:include page="/includes/header.jsp" />

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

<section>

<h1>Join our email list</h1>

<p>If you do, we'll send you announcements about new releases and special offers.</p>

<p><i>${message}</i></p>

<form action="<c:url value='/user/subscribeToEmail'/>" method="post">

<label>Email</label>

<input type="email" name="email" required><br>

<label>First Name</label>

<input type="text" name="firstName" required><br>

<label>Last Name</label>

<input type="text" name="lastName" required><br>

<label>&nbsp;</label>

<input type="submit" value="Join Now" id="submit">

</form>

</section>

<jsp:include page="/includes/column\_right\_news.jsp" />

<jsp:include page="/includes/footer.jsp" />

--------------------

jsp:include page="/includes/header.jsp" />

<section>

<h1>Thanks for joining our email list</h1>

<p>Here is the information that you entered:</p>

<label class="no\_pad\_top">Email</label>

<span>${user.email}</span><br>

<label class="no\_pad\_top">First Name</label>

<span>${user.firstName}</span><br>

<label class="no\_pad\_top">Last Name</label>

<span>${user.lastName}</span><br>

</section>

<jsp:include page="/includes/column\_right\_news.jsp" />

<jsp:include page="/includes/footer.jsp" />

------------------------------------

includes

----------------

<jsp:include page="/includes/header.jsp" />

<section>

<h1>Thanks for joining our email list</h1>

<p>Here is the information that you entered:</p>

<label class="no\_pad\_top">Email</label>

<span>${user.email}</span><br>

<label class="no\_pad\_top">First Name</label>

<span>${user.firstName}</span><br>

<label class="no\_pad\_top">Last Name</label>

<span>${user.lastName}</span><br>

</section>

<jsp:include page="/includes/column\_right\_news.jsp" />

<jsp:include page="/includes/footer.jsp" />

-------------------

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

<aside id="sidebarA">

<nav>

<ul>

<li><a href="<c:url value='/' />">

Home</a></li>

<li><a href="<c:url value='/catalog' />">

Browse Catalog</a></li>

<li><a href="<c:url value='/email' />">

Join Email List</a></li>

<li><a href="<c:url value='/customer\_service' />">

Customer Service</a></li>

</ul>

</nav>

</aside>

-----------------------------

<aside id="sidebarA">

<nav id="nav\_bar">

<ul>

<li><a href="http://localhost:8084/MusicStoreProductMaint/">

Home</a></li>

<li><a href="http://localhost:8084/MusicStoreProductMaint/catalog">

Browse Catalog</a></li>

<li><a href="http://localhost:8084/MusicStoreProductMaint/email">

Join Email List</a></li>

<li><a href="http://localhost:8084/MusicStoreProductMaint/customer\_service">

Customer Service</a></li>

<li><a href="<c:url value='http://localhost:8084/MusicStoreProductMaint/order/showCart'/>">

Show Cart</a></li>

</ul>

</nav>

</aside>

-------------------------------------

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

<aside id="sidebarA">

<nav>

<ul>

<li><a href="<c:url value='/' />">

Home</a></li>

<li><a class="current" href="<c:url value='/catalog' />">

Browse Catalog</a></li>

<li><a href="<c:url value='/email' />">

Join Email List</a></li>

<li><a href="<c:url value='/customer\_service' />">

Customer Service</a></li>

</ul>

</nav>

</aside>

-------------------------------------

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

<aside id="sidebarA">

<nav>

<ul>

<li><a href="<c:url value='/' />">

Home</a></li>

<li><a href="<c:url value='/catalog' />">

Browse Catalog</a></li>

<li><a href="<c:url value='/email' />">

Join Email List</a></li>

<li><a class="current" href="<c:url value='/customer\_service' />">

Customer Service</a></li>

</ul>

</nav>

</aside>

-------------------------------------

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

<aside id="sidebarA">

<nav>

<ul>

<li><a href="<c:url value='/' />">

Home</a></li>

<li><a href="<c:url value='/catalog' />">

Browse Catalog</a></li>

<li><a class="current" href="<c:url value='/email' />">

Join Email List</a></li>

<li><a href="<c:url value='/customer\_service' />">

Customer Service</a></li>

</ul>

</nav>

</aside>

-------------------------------------

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

<aside id="sidebarA">

<nav>

<ul>

<li><a class="current" href="<c:url value='/' />">

Home</a></li>

<li><a href="<c:url value='/catalog' />">

Browse Catalog</a></li>

<li><a href="<c:url value='/email' />">

Join Email List</a></li>

<li><a href="<c:url value='/customer\_service' />">

Customer Service</a></li>

</ul>

</nav>

</aside>

-------------------------------------

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

<aside id="sidebarB">

<form method="post" action="<c:url value='/order/addItem'/>">

<input type="hidden" name="productCode" value="${product.code}">

<input type="image" src="<c:url value='/images/addtocart.gif'/>"

width="113" alt="Add to Cart">

</form>

<a href="<c:url value='/catalog/product/${product.code}/listen' />">

<img src="<c:url value='/images/listen.gif'/>"

width="113" alt="Listen to Samples" class="top\_margin">

</a>

</aside>

-------------------------------------

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

<aside id="sidebarB">

<h1>Bernice Blogs</h1>

<img src="<c:url value='/images/Bernice.jpg' />" width="80"

alt="True Life ">

<h2>

Bernice Templeman - True Life Software Development Lessons

</h2>

<p class="news\_item">A refreshing mix of Java, Operating Systems, and Web Programming

that will have you earning money and growing your wealth.</p>

</aside>

-------------------------------------

<%@page contentType="text/html" pageEncoding="utf-8"%>

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

<!doctype html>

<html>

<head>

<meta charset="utf-8">

<title>Bernice Blogs</title>

<link rel="shortcut icon" href="<c:url value='/images/favicon.ico'/>">

<link rel="stylesheet" href="<c:url value='/styles/main.css'/> ">

<script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script>

<style>

ul {

list-style-type: none;

margin: 0;

padding: 0;

overflow: hidden;

background-color: #333;

}

li {

float: left;

}

li a {

display: block;

color: white;

text-align: center;

padding: 14px 16px;

text-decoration: none;

}

li a:hover {

background-color: #111;

}

</style>

</head>

<body>

<footer>

<nav id="nav\_bar2">

<ul>

<li style="float:right;"><a href="<c:url value='/admin'/>">

Admin</a></li>

<li style="float:right;"><a href="<c:url value='/user/deleteCookies'/>">

Delete Cookies</a></li>

</ul>

</nav>

<p>&copy; Copyright ${currentYear} Bernice Templeman

All rights reserved.</p>

</footer>

</body>

</html>

-------------------------------------

%@page contentType="text/html" pageEncoding="utf-8"%>

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

<!doctype html>

<html>

<head>

<meta charset="utf-8">

<title>Bernice Blogs</title>

<link rel="shortcut icon" href="<c:url value='/images/favicon.ico'/>">

<link rel="stylesheet" href="<c:url value='/styles/main.css'/> ">

<script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script>

<style>

ul {

list-style-type: none;

margin: 0;

padding: 0;

overflow: hidden;

background-color: #333;

}

li {

float: left;

}

li a {

display: block;

color: white;

text-align: center;

padding: 14px 16px;

text-decoration: none;

}

li a:hover {

background-color: #111;

}

</style>

</head>

<body>

<header>

<h1>The Bernice Blogs' Tech Store</h1>

</header>

<nav id="nav\_bar">

<ul>

<li><a href="<c:url value='/' />">

Home</a></li>

<li><a href="<c:url value='/catalog' />">

Browse Catalog</a></li>

<li><a href="<c:url value='/email' />">

Join Email List</a></li>

<li><a href="<c:url value='/customer\_service' />">

Customer Service</a></li>

<li><a href="<c:url value='/order/showCart'/>">

Show Cart</a></li>

</ul>

</nav>

-------------------------------------

<%--

Document : header\_all\_absolute

Created on : Apr 18, 2016, 11:25:04 PM

Author : Bernice

--%>

<%@page contentType="text/html" pageEncoding="utf-8"%>

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

<!doctype html>

<html>

<head>

<meta charset="utf-8">

<title>BerniceBlogs.com </title>

<link rel="stylesheet" href="<c:url value='/styles/main.css'/> ">

<script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script>

</head>

<body>

<header>

<h1>Bernice Blogs</h1>

<h2>About Technology</h2>

</header>

<style>

ul {

list-style-type: none;

margin: 0;

padding: 0;

overflow: hidden;

background-color: #333;

}

li {

float: left;

}

li a {

display: block;

color: white;

text-align: center;

padding: 14px 16px;

text-decoration: none;

}

li a:hover {

background-color: #111;

}

</style>

<nav id="nav\_bar">

<ul>

<li><a href="http://localhost:8084/MusicStoreProductMaint/">

Home</a></li>

<li><a href="http://localhost:8084/MusicStoreProductMaint/catalog">

Browse Catalog</a></li>

<li><a href="http://localhost:8084/MusicStoreProductMaint/email">

Join Email List</a></li>

<li><a href="http://localhost:8084/MusicStoreProductMaint/customer\_service">

Customer Service</a></li>

<li><a href="<c:url value='http://localhost:8084/MusicStoreProductMaint/order/showCart'/>">

Show Cart</a></li>

</ul>

</nav>

-------------------------------------

<img src="${product.imageURL}" width="175" height="175" alt="Album Image">

<h2>${product.artistName}</h2>

<h2>${product.albumName}</h2>

<p>${product.productType}</p>

<p>${product.priceCurrencyFormat}</p>

-------------------------------------

css

-------------------------------------

/\* type selectors \*/

article, aside, figure, footer, header, nav, section {

display: block;

}

html {

background-image: -moz-linear-gradient(top, white 0%, #facd8a 100%);

}

body {

font-family: Verdana, Arial, Helvetica, sans-serif;

font-size: 87.5%;

width: 964px;

background-color: white;

margin: 15px auto;

padding: 15px 25px;

border: 1px solid black;

}

section, aside, h1, h2, p, ul {

margin: 0;

padding: 0;

}

section, aside {

margin-top: 1.5em;

margin-bottom: 1em;

}

h1 { font-size: 175%; }

h2 { font-size: 125%; }

a { font-weight: bold; }

a:link { color: green; }

a:visited { color: green; }

a.current { color: #ef9c00; }

a:hover, a:focus { color: blue; }

ul {

margin-left: 1.5em;

margin-bottom: 1.5em;

}

li { padding-bottom: 0.25em; }

p { padding-bottom: 0.5em; }

em { font-weight: bold; }

/\* the styles for the header \*/

header {

padding-bottom: 2em;

border-bottom: 2px solid #ef9c00;

}

header h1 {

color: #ef9c00;

margin-left: 120px;

margin-bottom: 0.25em;

}

header h2 {

color: green;

font-style: italic;

margin-left: 120px;

}

header img {

float: left;

}

/\* the styles for the top navigation bar \*/

#nav\_bar {

clear: left;

margin: 0;

padding: 0;

border-bottom: 2px solid #ef9c00;

text-align: right;

}

#nav\_bar ul {

margin-bottom: .5em;

list-style: none;

padding-top: .5em;

}

#nav\_bar li {

display: inline;

padding: 0 1.5em;

border-left: 2px solid black;

}

/\* the styles for the side navigation bar \*/

#sidebarA ul {

margin-left: 0;

list-style: none;

}

#sidebarA li {

margin-bottom: .5em;

}

/\* the styles for the section \*/

section {

width: 520px;

padding-right: 20px;

padding-left: 20px;

float: left;

}

section h1 {

margin-bottom: 0.35em;

}

section h2 {

margin-bottom: 0.35em;

}

.top\_margin {

margin-top: 1em;

}

/\* product \*/

section img {

float: left;

margin-right: 1em;

margin-bottom: 1em;

}

.clear\_both {

clear: both;

}

/\* email list \*/

section label {

float: left;

width: 8em;

margin-bottom: 0.5em;

font-weight: bold;

padding-top: 0.25em;

}

section input[type="text"], input[type="email"], input[type="password"] {

width: 12em;

margin-left: 0.5em;

margin-bottom: 0.5em;

}

section input[type="submit"] {

margin-left: 0.5em;

}

br {

clear: both;

}

#submit {

margin-left: 0.5em;

}

table {

border: none;

}

th {

vertical-align: bottom;

text-align: left;

}

td {

vertical-align: top;

}

th, td {

padding: .3em;

}

.top {

vertical-align: top;

}

/\* the style for the cart pages \*/

.cart {

width: 700px;

}

.cart label {

float: left;

width: 10em;

font-weight: bold;

padding-top: 0.25em;

}

.cart input[type="text"], input[type="email"], input[type="password"] {

float: left;

width: 12em;

margin-left: 0.5em;

margin-right: 0.2em;

}

.cart input[type="button"] {

margin-left: 0.5em;

}

.cart select {

margin-left: 0.5em;

margin-right: 0.2em;

}

.cart\_row input[type="text"] {

width: 2em;

margin-left: 0em;

}

#required {

margin-left: 8.5em;

margin-bottom: .5em;

}

.required {

color: red;

margin-bottom: 0;

}

#quantity\_text {

width: 2em;

}

#float\_left {

float: left;

margin-right: 1em;

}

/\* the style for the admin pages \*/

#admin {

width: 700px;

}

#admin form {

margin-top: .5em;

}

#admin label {

width: 14em;

margin-left: .5em;

padding-top: 0.25em;

}

#admin .left\_margin {

margin-left: 0em;

}

/\* the styles for the sidebars \*/

aside h1 {

font-size: 125%;

padding-bottom: .5em;

}

aside h2 {

font-size: 100%;

color: green;

padding-bottom: .5em;

}

aside p {

margin-bottom: .5em;

}

aside img {

margin-bottom: 1em;

}

#sidebarA {

width: 180px;

float: left;

height: 400px;

border-right: 2px solid #ef9c00;

}

#sidebarB {

width: 180px;

float: right;

height: 400px;

padding-left: 20px;

border-left: 2px solid #ef9c00;

}

/\* the styles for the footer \*/

footer {

clear: both;

margin-top: 1em;

border-top: 2px solid #ef9c00;

padding-top: 0.7em;

}

footer p {

font-size: 80%;

text-align: right;

}

/\* The styles for the classes \*/

.pad\_top {

padding-top: 0.25em;

}

.no\_pad\_top {

padding-top: 0;

}

.margin\_left {

margin-left: 0.5em;

}

.no\_underline {

text-decoration:none;

}

-------------------------------------

/\* type selectors \*/

article, aside, figure, footer, header, nav, section {

display: block;

}

html {

}

body {

font-family: Verdana, Arial, Helvetica, sans-serif;

font-size: 87.5%;

width: 964px;

background-color: white;

margin: 15px auto;

padding: 15px 25px;

border: 1px solid black;

}

section, aside, h1, h2, p, ul {

margin: 0;

padding: 0;

}

section, aside {

margin-top: 1.5em;

margin-bottom: 1em;

}

h1 { font-size: 175%; }

h2 { font-size: 125%; }

a { font-weight: bold; }

a:link { color: #146EB4 ; }

a:visited { color: #146EB4 ; }

a.current { color: #FF9900; }

a:hover, a:focus { color: #FF9900; }

ul {

margin-left: 1.5em;

margin-bottom: 1.5em;

}

li {

padding-bottom: 0.25em;

}

p { padding-bottom: 0.5em; }

em { font-weight: bold; }

/\* the styles for the header \*/

header {

padding-bottom: 2em;

border-bottom: 2px solid black;

}

header h1 {

color: black;

margin-left: 120px;

margin-bottom: 0.25em;

}

header h2 {

color: gray;

font-style: italic;

margin-left: 120px;

}

header img {

float: left;

}

/\* the styles for the top navigation bar \*/

#nav\_bar {

clear: left;

margin: 0;

padding: 0;

border-bottom: 2px solid black;

text-align: right;

}

#nav\_bar ul {

margin-bottom: .5em;

list-style: none;

padding-top: .5em;

}

#nav\_bar ul li:hover a{

color: #FF9900 ; }

#nav\_bar ul li a, visited {

color: white;

}

#nav\_bar ul li:visited a{

color: white ; }

#nav\_bar ul li:current a{

color: white; }

#nav\_bar ul li:hover,focus a{

color: white; }

#nav\_bar li {

display: inline;

padding: 0 1.5em;

border-left: 2px solid black;

}

#nav\_bar2 {

clear: left;

margin: 0;

padding: 0;

border-bottom: 2px solid black;

text-align: right;

}

#nav\_bar2 ul {

margin-bottom: .5em;

list-style: none;

padding-top: .5em;

}

#nav\_bar2 ul li:hover a{

color: #FF9900 ; }

#nav\_bar2 ul li a, visited {

color: white;

}

#nav\_bar2 ul li:visited a{

color: white ; }

#nav\_bar2 ul li:current a{

color: white; }

#nav\_bar2 ul li:hover,focus a{

color: white; }

#nav\_bar2 li {

display: inline;

padding: 0 1.5em;

border-right: 2px solid black;

text-align: right;

}

/\* the styles for the side navigation bar \*/

#sidebarA ul {

margin-left: 0;

list-style: none;

}

#sidebarA li {

margin-bottom: .5em;

}

/\* the styles for the section \*/

section {

width: 520px;

padding-right: 20px;

padding-left: 20px;

float: left;

}

section h1 {

margin-bottom: 0.35em;

}

section h2 {

margin-bottom: 0.35em;

}

.top\_margin {

margin-top: 1em;

}

/\* product \*/

section img {

float: left;

margin-right: 1em;

margin-bottom: 1em;

}

.clear\_both {

clear: both;

}

/\* email list \*/

section label {

float: left;

width: 8em;

margin-bottom: 0.5em;

font-weight: bold;

padding-top: 0.25em;

}

section input[type="text"], input[type="email"], input[type="password"] {

width: 12em;

margin-left: 0.5em;

margin-bottom: 0.5em;

}

section input[type="submit"] {

margin-left: 0.5em;

}

br {

clear: both;

}

#submit {

margin-left: 0.5em;

}

table {

border: none;

}

th {

vertical-align: bottom;

text-align: left;

}

td {

vertical-align: top;

}

th, td {

padding: .3em;

}

.top {

vertical-align: top;

}

/\* the style for the cart pages \*/

.cart {

width: 700px;

}

.cart label {

float: left;

width: 10em;

font-weight: bold;

padding-top: 0.25em;

}

.cart input[type="text"], input[type="email"], input[type="password"] {

float: left;

width: 12em;

margin-left: 0.5em;

margin-right: 0.2em;

}

.cart input[type="button"] {

margin-left: 0.5em;

}

.cart select {

margin-left: 0.5em;

margin-right: 0.2em;

}

.cart\_row input[type="text"] {

width: 2em;

margin-left: 0em;

}

#required {

margin-left: 8.5em;

margin-bottom: .5em;

}

.required {

color: red;

margin-bottom: 0;

}

#quantity\_text {

width: 2em;

}

#float\_left {

float: left;

margin-right: 1em;

}

/\* the style for the admin pages \*/

#admin {

width: 700px;

}

#admin form {

margin-top: .5em;

}

#admin label {

width: 14em;

margin-left: .5em;

padding-top: 0.25em;

}

#admin .left\_margin {

margin-left: 0em;

}

/\* the styles for the sidebars \*/

aside h1 {

font-size: 125%;

padding-bottom: .5em;

}

aside h2 {

font-size: 100%;

color: green;

padding-bottom: .5em;

}

aside p {

margin-bottom: .5em;

}

aside img {

margin-bottom: 1em;

}

#sidebarA {

width: 180px;

float: left;

height: 400px;

border-right: 2px solid black;

}

#sidebarB {

width: 180px;

float: right;

height: 400px;

padding-left: 20px;

border-left: 2px solid black;

}

/\* the styles for the footer \*/

footer {

clear: both;

margin-top: 1em;

border-top: 2px solid black;

padding-top: 0.7em;

}

footer p {

font-size: 80%;

text-align: right;

}

/\* The styles for the classes \*/

.pad\_top {

padding-top: 0.25em;

}

.no\_pad\_top {

padding-top: 0;

}

.margin\_left {

margin-left: 0.5em;

}

.no\_underline {

text-decoration:none;

}

-------------------------------------

//ProductCOntroller.java

/\*

Bernice Templeman

CIS 5740/ Spring 2016

Music Store Product Maintenance

\*/

package music.admin;

import java.io.IOException;

import java.io.PrintWriter;

import java.util.regex.Pattern;

import javax.servlet.ServletContext;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import music.business.Product;

import music.data.ProductDB;

public class ProductController extends HttpServlet

{

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException

{

HttpSession session = request.getSession();

session.setAttribute("error", null);

String requestURI = request.getRequestURI();

String url = "";

if (requestURI.endsWith("/addProduct"))

{

// set a product into the session if indicated, otherwise clear it

if (request.getParameter("productCode") != null)

{

session.setAttribute("product", ProductDB.selectByCode(request.getParameter("productCode")));

}

else

{

session.setAttribute("product", null);

}

url = "/admin/prodAdd.jsp";

}

else if (requestURI.endsWith("/addConfirmed"))

{

// create temp product

Product product = ProductDB.selectByCode(request.getParameter("prodCode"));

Boolean isInsert = false;

if (product == null)

{

isInsert = !isInsert;

product = new Product();

}

// get parameters from the request

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

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

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

product.setCode(prodCode);

product.setDescription(prodDescription);

double prodPriceD = 0.0;

String message = "";

boolean match = false;

if(!prodPrice.isEmpty())

{

//check if price is valid double value

String doublePattern = "([0-9].\*)\\.([0-9].\*)";

match = Pattern.matches(doublePattern, (request.getParameter("prodPrice")));

if (match)

{

try

{

prodPriceD = Double.valueOf(request.getParameter("prodPrice"));

}

catch(Exception e)

{

product.setPrice(prodPriceD);

session.setAttribute("product", product);

match = false;

message = "You must enter a valid price for the product. Example: 1.00";

request.setAttribute("message", message);

url = "/admin/prodAdd.jsp";

}

}

}// prod price is not empty

else //prod price is empty

{

// store the Product object in the session

match = false;

message = "You must enter a valid price for the product. Example: 1.00";

request.setAttribute("message", message);

url = "/admin/prodAdd.jsp";

}

product.setPrice(prodPriceD);

session.setAttribute("product", product);

if (match)

{

//check if all values complete

if ((prodCode.isEmpty() || prodDescription.isEmpty()) || prodPrice.isEmpty())

{

// forward to the view to get missing parameters

url = "/admin/prodAdd.jsp";

if (prodCode.isEmpty())

message = "You must enter a code for the product.";

else if (prodDescription.isEmpty())

message = "You must enter a description for the product.";

else if (prodPrice.isEmpty())

message = "You must enter a price for the product.";

request.setAttribute("message", message);

}

else //all complete

{

//check if price is valid double value

String doublePattern = "([0-9].\*)\\.([0-9].\*)";

match = Pattern.matches(doublePattern, (request.getParameter("prodPrice")));

if(match) //check if 2 decimal places

{

String[] splitter = (request.getParameter("prodPrice")).split("\\.");

splitter[0].length(); // Before Decimal Count

int decimalLength = splitter[1].length(); // After Decimal Count

if (decimalLength == 2)

{

// decide between update and insert

if (!isInsert)

{

ProductDB.update(product);

}

else

{

ProductDB.insert(product);

}

session.setAttribute("products", ProductDB.selectAll());

url = "/admin/prodList.jsp";

}

else

{

message = "You must enter a valid price for the product. Example: 1.00";

request.setAttribute("message", message);

url = "/admin/prodAdd.jsp";

}

}//if match

else

{

message = "You must enter a valid price for the product. Example: 1.00";

request.setAttribute("message", message);

url = "/admin/prodAdd.jsp";

}

}//end if all complete

}//end if match

else

{

message = "You must enter a valid price for the product. Example: 1.00";

request.setAttribute("message", message);

url = "/admin/prodAdd.jsp";

}

}//end add confirmed

else if (requestURI.endsWith("/deleteProduct"))

{

// set a product into the session if indicated, otherwise clear it

if (request.getParameter("productCode") != null)

{

session.setAttribute("product", ProductDB.selectByCode(request.getParameter("productCode")));

}

else

{

session.setAttribute("product", null);

}

url = "/admin/prodDelete.jsp";

}//end delete product

else if (requestURI.endsWith("/deleteConfirmed"))

{

Product delProduct = (Product) session.getAttribute("product");

if (delProduct != null)

{

ProductDB.delete(delProduct);

}

session.setAttribute("products", ProductDB.selectAll());

url = "/admin/prodList.jsp";

}//end delete confirmed

else if (requestURI.endsWith("/prodList"))

{

// if none of those, show all products

session.setAttribute("products", ProductDB.selectAll());

url = "/admin/prodList.jsp";

}//end display products

if (url.equals("") != true)

{

System.out.println("url=" + url);

getServletContext().getRequestDispatcher(url).forward(request, response);

}

else

{

// display 404 error page

response.sendError(404);

}

}

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException

{

doPost(request, response);

}

}

-------------------------------------

//UserMaintServlet

/\*

Bernice Templeman

CIS 5740/ Spring 2016

Music Store Product Maintenance

\*/

package music.admin;

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

import java.util.regex.Pattern;

import music.business.User;

import music.data.UserDB;

public class UserMaintServlet extends HttpServlet

{

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException

{

// get the products and place into session var

ServletContext sc = getServletContext();

HttpSession session = request.getSession();

session.setAttribute("users", UserDB.selectUsers());

session.setAttribute("error", null);

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

if (action == null)

action = "displayUsers";

String url="";

if (action.equals("displayUsers"))

{

session.setAttribute("users", UserDB.selectUsers());

url = "/admin/users.jsp";

}

else if (action.equals("addUser"))

{

// set a product into the session if indicated, otherwise clear it

if (request.getParameter("email") != null)

{

session.setAttribute("user", UserDB.selectUser(request.getParameter("email")));

}

else

{

session.setAttribute("user", null);

}

url = "/admin/addUser.jsp";

}

else if (action.equals( "addUserConfirmed"))

{

// create temp product

User user = UserDB.selectUser(request.getParameter("email"));

Boolean isInsert = false;

if (user == null)

{

isInsert = !isInsert;

user = new User();

}

// populate these first, in case of Double error

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

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

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

user.setEmail(email);

user.setFirstName(firstName);

user.setLastName(lastName);

// store the Product object in the session

session.setAttribute("user", user);

String message = "";

//check if all values complete

if ((email.isEmpty() || firstName.isEmpty()) || lastName.isEmpty())

{

// forward to the view to get missing parameters

url = "/admin/addUser.jsp";

if (email.isEmpty())

message = "You must enter an email for the user.";

if (firstName.isEmpty())

message = "You must enter a first name for the user.";

if (lastName.isEmpty())

message = "You must enter a last name for the user.";

request.setAttribute("message", message);

}

else //all complete

{

if (isInsert)

{

UserDB.insert(user);

}

else

{

UserDB.update(user);

}

// store the Products in the session

session.setAttribute("users", UserDB.selectUsers());

url = "/admin/users.jsp";

}//all complete

}//end if add confirmed

else if (action.equals( "deleteUser"))

{

// set a product into the session if indicated, otherwise clear it

if (request.getParameter("email") != null)

{

session.setAttribute("user", UserDB.selectUser(request.getParameter("email")));

} else

{

session.setAttribute("user", null);

}

url = "/admin/deleteUser.jsp";

}

else if (action.equals( "deleteUserConfirmed"))

{

User delUser = (User) session.getAttribute("user");

if (delUser != null)

{

UserDB.delete(delUser);

}

session.setAttribute("users", UserDB.selectUsers());

url = "/admin/users.jsp";

}

sc.getRequestDispatcher(url).forward(request, response);

}

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException

{

doPost(request, response);

}

}

-------------------------------------

//users servlet

package music.admin;

import java.io.\*;

import java.util.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

import music.business.User;

import music.data.UserDB;

public class UsersServlet extends HttpServlet {

@Override

protected void doPost(HttpServletRequest request,

HttpServletResponse response)

throws ServletException, IOException {

HttpSession session = request.getSession();

String url = "/index.jsp";

// get current action

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

if (action == null) {

action = "display\_users"; // default action

}

// perform action and set URL to appropriate page

if (action.equals("display\_users")) {

// get list of users

List<User> users = UserDB.selectUsers();

request.setAttribute("users", users);

}

else if (action.equals("display\_user")) {

String emailAddress = request.getParameter("email");

User user = UserDB.selectUser(emailAddress);

session.setAttribute("user", user);

url = "/user.jsp";

}

else if (action.equals("update\_user")) {

// get parameters from the request

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

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

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

// get and update user

User user = (User) session.getAttribute("user");

user.setEmail(email);

user.setFirstName(firstName);

user.setLastName(lastName);

UserDB.update(user);

// get and set updated users

List<User> users = UserDB.selectUsers();

request.setAttribute("users", users);

}

else if (action.equals("delete\_user")) {

// get the user

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

User user = UserDB.selectUser(email);

// delte the user

UserDB.delete(user);

// get and set updated users

List<User> users = UserDB.selectUsers();

request.setAttribute("users", users);

}

getServletContext()

.getRequestDispatcher(url)

.forward(request, response);

}

@Override

protected void doGet(HttpServletRequest request,

HttpServletResponse response)

throws ServletException, IOException {

doPost(request, response);

}

}

-------------------------------------

//cart.java

package music.business;

import java.util.\*;

import java.io.Serializable;

public class Cart implements Serializable {

private List<LineItem> items;

public Cart() {

items = new ArrayList<>();

}

public void setItems(List<LineItem> lineItems) {

items = lineItems;

}

public List<LineItem> getItems() {

return items;

}

public int getCount() {

return items.size();

}

public void addItem(LineItem item) {

//If the item already exists in the cart, only the quantity is changed.

String code = item.getProduct().getCode();

int quantity = item.getQuantity();

for (int i = 0; i < items.size(); i++) {

LineItem lineItem = items.get(i);

if (lineItem.getProduct().getCode().equals(code)) {

lineItem.setQuantity(quantity);

return;

}

}

items.add(item);

}

public void removeItem(LineItem item) {

String code = item.getProduct().getCode();

for (int i = 0; i < items.size(); i++) {

LineItem lineItem = items.get(i);

if (lineItem.getProduct().getCode().equals(code)) {

items.remove(i);

return;

}

}

}

}

-------------------------------------

package music.business;

import java.util.Date;

import java.io.Serializable;

import javax.persistence.Entity;

import javax.persistence.FetchType;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.ManyToOne;

import javax.persistence.Temporal;

import javax.persistence.TemporalType;

@Entity

public class Download implements Serializable {

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

private Long downloadId;

@ManyToOne(fetch=FetchType.EAGER)

private User user;

@Temporal(TemporalType.DATE)

private Date downloadDate;

private String productCode;

public Download() {

user = null;

downloadDate = new Date();

productCode = "";

}

public Long getDownloadId() {

return downloadId;

}

public void setDownloadId(Long downloadId) {

this.downloadId = downloadId;

}

public void setUser(User u) {

user = u;

}

public User getUser() {

return user;

}

public void setDownloadDate(Date date) {

downloadDate = date;

}

public Date getDownloadDate() {

return downloadDate;

}

public void setProductCode(String productCode) {

this.productCode = productCode;

}

public String getProductCode() {

return productCode;

}

}

package music.business;

import java.util.\*;

import java.text.\*;

import java.io.Serializable;

import javax.persistence.CascadeType;

import javax.persistence.Entity;

import static javax.persistence.FetchType.EAGER;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.ManyToOne;

import javax.persistence.OneToMany;

import javax.persistence.Temporal;

import javax.persistence.TemporalType;

@Entity

public class Invoice implements Serializable {

@ManyToOne

private User user;

@OneToMany(fetch=EAGER, cascade=CascadeType.PERSIST)

private List<LineItem> lineItems;

@Temporal(TemporalType.DATE)

private Date invoiceDate;

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

private Long invoiceNumber;

private boolean isProcessed;

public Invoice() {

}

public void setUser(User user) {

this.user = user;

}

public User getUser() {

return user;

}

public void setLineItems(List<LineItem> lineItems) {

this.lineItems = lineItems;

}

public List<LineItem> getLineItems() {

return lineItems;

}

public void setInvoiceDate(Date invoiceDate) {

this.invoiceDate = invoiceDate;

}

public Date getInvoiceDate() {

return invoiceDate;

}

public String getInvoiceDateDefaultFormat() {

DateFormat dateFormat = DateFormat.getDateInstance();

String invoiceDateFormatted = dateFormat.format(invoiceDate);

return invoiceDateFormatted;

}

public void setInvoiceNumber(Long invoiceNumber) {

this.invoiceNumber = invoiceNumber;

}

public Long getInvoiceNumber() {

return invoiceNumber;

}

public boolean isIsProcessed() {

return isProcessed;

}

public void setIsProcessed(boolean isProcessed) {

this.isProcessed = isProcessed;

}

public double getInvoiceTotal() {

double invoiceTotal = 0.0;

for (LineItem item : lineItems) {

invoiceTotal += item.getTotal();

}

return invoiceTotal;

}

public String getInvoiceTotalCurrencyFormat() {

double total = this.getInvoiceTotal();

NumberFormat currency = NumberFormat.getCurrencyInstance();

String formattedTotal = currency.format(total);

return formattedTotal;

}

}

package music.business;

import java.text.NumberFormat;

import java.io.Serializable;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.OneToOne;

@Entity

public class LineItem implements Serializable {

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

private Long lineItemId;

@OneToOne

private Product product;

private int quantity = 1;

public LineItem() {

}

public Long getLineItemId() {

return lineItemId;

}

public void setLineItemId(Long lineItemId) {

this.lineItemId = lineItemId;

}

public void setProduct(Product product) {

this.product = product;

}

public Product getProduct() {

return product;

}

public void setQuantity(int quantity) {

this.quantity = quantity;

}

public int getQuantity() {

return quantity;

}

public double getTotal() {

double total = product.getPrice() \* quantity;

return total;

}

public String getTotalCurrencyFormat() {

NumberFormat currency = NumberFormat.getCurrencyInstance();

return currency.format(this.getTotal());

}

}

package music.business;

import java.text.NumberFormat;

import java.io.Serializable;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

@Entity

public class Product implements Serializable

{

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

private Long productId;

private String code;

private String description;

private double price;

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";

}

}

package music.business;

import java.io.Serializable;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

@Entity

public class User implements Serializable {

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

private Long userId;

private String firstName;

private String lastName;

private String email;

private String companyName;

private String address1;

private String address2;

private String city;

private String st;

private String zip;

private String country;

private String creditCardType;

private String creditCardNumber;

private String creditCardExpirationDate;

public User() {

firstName = "";

lastName = "";

email = "";

companyName = "";

address1 = "";

address2 = "";

city = "";

st = "";

zip = "";

country = "";

creditCardType = "";

creditCardNumber = "";

creditCardExpirationDate = "";

}

public Long getId() {

return userId;

}

public void setId(Long userId) {

this.userId = userId;

}

public void setFirstName(String firstName) {

this.firstName = firstName;

}

public String getFirstName() {

return firstName;

}

public void setLastName(String lastName) {

this.lastName = lastName;

}

public String getLastName() {

return lastName;

}

public void setEmail(String email) {

this.email = email;

}

public String getEmail() {

return email;

}

public void setCompanyName(String companyName) {

this.companyName = companyName;

}

public String getCompanyName() {

return companyName;

}

public void setAddress1(String address1) {

this.address1 = address1;

}

public String getAddress1() {

return address1;

}

public void setAddress2(String address2) {

this.address2 = address2;

}

public String getAddress2() {

return address2;

}

public void setCity(String city) {

this.city = city;

}

public String getCity() {

return city;

}

// State is a reserved SQL word, so the field is called "st" instead.

public void setState(String state) {

st = state;

}

public String getState() {

return st;

}

public void setZip(String zip) {

this.zip = zip;

}

public String getZip() {

return zip;

}

public void setCountry(String country) {

this.country = country;

}

public String getCountry() {

return country;

}

// this works, but it mixes the Model and the View

public String getAddressHTMLFormat() {

String address = firstName + " " + lastName + "<br>";

if (companyName == null || companyName.equals("") || companyName.equals(" ")) {

address += "";

} else {

address += companyName + "<br>";

}

address += address1 + "<br>";

if (address2 == null || address2.equals("") || address2.equals(" ")) {

address += "";

} else {

address += address2 + "<br>";

}

address += city + ", " + st + " " + zip + "<br>"

+ country;

return address;

}

public void setCreditCardType(String creditCartType) {

this.creditCardType = creditCartType;

}

public String getCreditCardType() {

return creditCardType;

}

public void setCreditCardNumber(String creditCardNumber) {

this.creditCardNumber = creditCardNumber;

}

public String getCreditCardNumber() {

return creditCardNumber;

}

public void setCreditCardExpirationDate(String creditCardExpirationDate) {

this.creditCardExpirationDate = creditCardExpirationDate;

}

public String getCreditCardExpirationDate() {

return creditCardExpirationDate;

}

}

package music.controllers;

import java.io.IOException;

import java.io.OutputStream;

import java.util.List;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import org.apache.poi.hssf.usermodel.HSSFWorkbook;

import org.apache.poi.ss.usermodel.Workbook;

import music.business.\*;

import music.data.\*;

public class AdminController extends HttpServlet {

@Override

public void doPost(HttpServletRequest request,

HttpServletResponse response)

throws IOException, ServletException {

String requestURI = request.getRequestURI();

String url = "/admin";

if (requestURI.endsWith("/displayInvoices")) {

url = displayInvoices(request, response);

} else if (requestURI.endsWith("/processInvoice")) {

url = processInvoice(request, response);

} else if (requestURI.endsWith("/displayReport")) {

displayReport(request, response);

}

getServletContext()

.getRequestDispatcher(url)

.forward(request, response);

}

@Override

public void doGet(HttpServletRequest request,

HttpServletResponse response)

throws IOException, ServletException {

String requestURI = request.getRequestURI();

String url = "/admin";

if (requestURI.endsWith("/displayInvoice")) {

url = displayInvoice(request, response);

} else if (requestURI.endsWith("/displayInvoices")) {

url = displayInvoices(request, response);

}

getServletContext()

.getRequestDispatcher(url)

.forward(request, response);

}

private String displayInvoices(HttpServletRequest request,

HttpServletResponse response) throws IOException {

List<Invoice> unprocessedInvoices

= InvoiceDB.selectUnprocessedInvoices();

String url;

if (unprocessedInvoices != null) {

if (unprocessedInvoices.size() <= 0) {

unprocessedInvoices = null;

}

}

HttpSession session = request.getSession();

session.setAttribute("unprocessedInvoices", unprocessedInvoices);

url = "/admin/invoices.jsp";

return url;

}

private String displayInvoice(HttpServletRequest request,

HttpServletResponse response) {

HttpSession session = request.getSession();

String invoiceNumberString = request.getParameter("invoiceNumber");

int invoiceNumber = Integer.parseInt(invoiceNumberString);

List<Invoice> unprocessedInvoices = (List<Invoice>)

session.getAttribute("unprocessedInvoices");

Invoice invoice = null;

for (Invoice unprocessedInvoice : unprocessedInvoices) {

invoice = unprocessedInvoice;

if (invoice.getInvoiceNumber() == invoiceNumber) {

break;

}

}

session.setAttribute("invoice", invoice);

return "/admin/invoice.jsp";

}

private String processInvoice(HttpServletRequest request,

HttpServletResponse response) throws IOException {

HttpSession session = request.getSession();

Invoice invoice = (Invoice) session.getAttribute("invoice");

InvoiceDB.update(invoice);

return "/adminController/displayInvoices";

}

private void displayReport(HttpServletRequest request,

HttpServletResponse response) throws IOException {

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

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

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

Workbook workbook;

if (reportName.equalsIgnoreCase("userEmail")) {

workbook = ReportDB.getUserEmail();

} else if (reportName.equalsIgnoreCase("downloadDetail")) {

workbook = ReportDB.getDownloadDetail(startDate, endDate);

} else {

workbook = new HSSFWorkbook();

}

response.setHeader("content-disposition",

"attachment; filename=" + reportName + ".xls");

try (OutputStream out = response.getOutputStream()) {

workbook.write(out);

}

}

}

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.http.\*;

import music.business.\*;

import music.data.\*;

import music.util.CookieUtil;

public class CatalogController extends HttpServlet {

@Override

public void doGet(HttpServletRequest request,

HttpServletResponse response)

throws ServletException, IOException {

String requestURI = request.getRequestURI();

String url;

if (requestURI.endsWith("/listen")) {

url = listen(request, response);

} else {

url = showProduct(request, response);

}

getServletContext()

.getRequestDispatcher(url)

.forward(request, response);

}

@Override

public void doPost(HttpServletRequest request,

HttpServletResponse response)

throws ServletException, IOException {

String requestURI = request.getRequestURI();

String url = "/catalog";

if (requestURI.endsWith("/register")) {

url = registerUser(request, response);

}

getServletContext()

.getRequestDispatcher(url)

.forward(request, response);

}

private String showProduct(HttpServletRequest request,

HttpServletResponse response) {

String productCode = request.getPathInfo();

// This should never be null. But just to be safe.

if (productCode != null) {

productCode = productCode.substring(1);

Product product = ProductDB.selectProduct(productCode);

HttpSession session = request.getSession();

session.setAttribute("product", product);

}

return "/catalog/" + productCode + "/index.jsp";

}

private String listen(HttpServletRequest request,

HttpServletResponse response) {

HttpSession session = request.getSession();

User user = (User) session.getAttribute("user");

// if the User object doesn't exist, check for the email cookie

if (user == null) {

Cookie[] cookies = request.getCookies();

String emailAddress =

CookieUtil.getCookieValue(cookies, "emailCookie");

// if the email cookie doesn't exist, go to the registration page

if (emailAddress == null || emailAddress.equals("")) {

return "/catalog/register.jsp";

} else {

user = UserDB.selectUser(emailAddress);

// if a user for that email isn't in the database, go to the registration page

if (user == null) {

return "/catalog/register.jsp";

}

session.setAttribute("user", user);

}

}

Product product = (Product) session.getAttribute("product");

Download download = new Download();

download.setUser(user);

download.setProductCode(product.getCode());

DownloadDB.insert(download);

return "/catalog/" + product.getCode() + "/sound.jsp";

}

private String registerUser(HttpServletRequest request,

HttpServletResponse response) {

HttpSession session = request.getSession();

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

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

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

User user = new User();

user.setFirstName(firstName);

user.setLastName(lastName);

user.setEmail(email);

if (UserDB.emailExists(email)) {

UserDB.update(user);

} else {

UserDB.insert(user);

}

session.setAttribute("user", user);

Cookie emailCookie = new Cookie("emailCookie", email);

emailCookie.setMaxAge(60 \* 60 \* 24 \* 365 \* 2);

emailCookie.setPath("/");

response.addCookie(emailCookie);

Product product = (Product) session.getAttribute("product");

String url = "/catalog/" + product.getCode() + "/sound.jsp";

return url;

}

}

package music.controllers;

import java.io.IOException;

import javax.mail.MessagingException;

import javax.servlet.ServletException;

import javax.servlet.http.\*;

import music.business.\*;

import music.data.\*;

import music.util.\*;

public class OrderController extends HttpServlet {

private static final String defaultURL = "/cart/cart.jsp";

@Override

public void doPost(HttpServletRequest request,

HttpServletResponse response)

throws ServletException, IOException {

String requestURI = request.getRequestURI();

String url = "";

if (requestURI.endsWith("/addItem")) {

url = addItem(request, response);

} else if (requestURI.endsWith("/updateItem")) {

url = updateItem(request, response);

} else if (requestURI.endsWith("/removeItem")) {

url = removeItem(request, response);

} else if (requestURI.endsWith("/checkUser")) {

url = checkUser(request, response);

} else if (requestURI.endsWith("/processUser")) {

url = processUser(request, response);

} else if (requestURI.endsWith("/displayInvoice")) {

url = displayInvoice(request, response);

} else if (requestURI.endsWith("/displayUser")) {

url = "/cart/user.jsp";

} else if (requestURI.endsWith("/displayCreditCard")) {

url = "/cart/credit\_card.jsp";

} else if (requestURI.endsWith("/completeOrder")) {

url = completeOrder(request, response);

}

getServletContext()

.getRequestDispatcher(url)

.forward(request, response);

}

@Override

public void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

String requestURI = request.getRequestURI();

String url = defaultURL;

if (requestURI.endsWith("/showCart")) {

showCart(request, response);

} else if (requestURI.endsWith("/checkUser")) {

url = checkUser(request, response);

}

getServletContext()

.getRequestDispatcher(url)

.forward(request, response);

}

private String showCart(HttpServletRequest request,

HttpServletResponse response) {

HttpSession session = request.getSession();

Cart cart = (Cart) session.getAttribute("cart");

if (cart == null || cart.getCount() == 0) {

request.setAttribute("emptyCart", "Your cart is empty");

} else {

request.getSession().setAttribute("cart", cart);

}

return defaultURL;

}

private String addItem(HttpServletRequest request,

HttpServletResponse response) {

HttpSession session = request.getSession();

Cart cart = (Cart) session.getAttribute("cart");

if (cart == null)

cart = new Cart();

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

Product product = ProductDB.selectProduct(productCode);

if (product != null) {

LineItem lineItem = new LineItem();

lineItem.setProduct(product);

cart.addItem(lineItem);

}

session.setAttribute("cart", cart);

return defaultURL;

}

private String updateItem(HttpServletRequest request,

HttpServletResponse response) {

String quantityString = request.getParameter("quantity");

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

HttpSession session = request.getSession();

Cart cart = (Cart) session.getAttribute("cart");

int quantity;

try {

quantity = Integer.parseInt(quantityString);

if (quantity < 0)

quantity = 1;

} catch (NumberFormatException ex) {

quantity = 1;

}

Product product = ProductDB.selectProduct(productCode);

if (product != null && cart != null) {

LineItem lineItem = new LineItem();

lineItem.setProduct(product);

lineItem.setQuantity(quantity);

if (quantity > 0)

cart.addItem(lineItem);

else

cart.removeItem(lineItem);

}

return defaultURL;

}

private String removeItem(HttpServletRequest request,

HttpServletResponse response) {

HttpSession session = request.getSession();

Cart cart = (Cart) session.getAttribute("cart");

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

Product product = ProductDB.selectProduct(productCode);

if (product != null && cart != null) {

LineItem lineItem = new LineItem();

lineItem.setProduct(product);

cart.removeItem(lineItem);

}

return defaultURL;

}

private String checkUser(HttpServletRequest request,

HttpServletResponse response) {

HttpSession session = request.getSession();

User user = (User) session.getAttribute("user");

// if the User object exists with address1, skip User page

String url = "/cart/user.jsp";

if (user != null && !user.getAddress1().equals("")) {

url = "/order/displayInvoice";

} else { // otherwise, check the email cookie

Cookie[] cookies = request.getCookies();

String email

= CookieUtil.getCookieValue(cookies, "emailCookie");

if (email.equals("")) {

user = new User();

url = "/cart/user.jsp";

} else {

user = UserDB.selectUser(email);

if (user != null && !user.getAddress1().equals("")) {

url = "/order/displayInvoice";

}

}

}

session.setAttribute("user", user);

return url;

}

private String processUser(HttpServletRequest request,

HttpServletResponse response) {

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

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

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

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

String address1 = request.getParameter("address1");

String address2 = request.getParameter("address2");

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

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

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

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

HttpSession session = request.getSession();

User user = (User) session.getAttribute("user");

if (user == null) {

user = new User();

}

user.setFirstName(firstName);

user.setLastName(lastName);

user.setEmail(email);

user.setCompanyName(companyName);

user.setAddress1(address1);

user.setAddress2(address2);

user.setCity(city);

user.setState(state);

user.setZip(zip);

user.setCountry(country);

if (UserDB.emailExists(email)) {

UserDB.update(user);

} else {

UserDB.insert(user);

}

session.setAttribute("user", user);

return "/order/displayInvoice";

}

private String displayInvoice(HttpServletRequest request,

HttpServletResponse response) {

HttpSession session = request.getSession();

User user = (User) session.getAttribute("user");

Cart cart = (Cart) session.getAttribute("cart");

java.util.Date today = new java.util.Date();

Invoice invoice = new Invoice();

invoice.setUser(user);

invoice.setInvoiceDate(today);

invoice.setLineItems(cart.getItems());

session.setAttribute("invoice", invoice);

return "/cart/invoice.jsp";

}

private String completeOrder(HttpServletRequest request,

HttpServletResponse response) {

HttpSession session = request.getSession();

User user = (User)session.getAttribute("user");

Invoice invoice = (Invoice)session.getAttribute("invoice");

String creditCardType =

request.getParameter("creditCardType");

String creditCardNumber =

request.getParameter("creditCardNumber");

String creditCardExpMonth =

request.getParameter("creditCardExpirationMonth");

String creditCardExpYear =

request.getParameter("creditCardExpirationYear");

user.setCreditCardType(creditCardType);

user.setCreditCardNumber(creditCardNumber);

user.setCreditCardExpirationDate(creditCardExpMonth

+ "/" + creditCardExpYear);

// if a record for the User object exists, update it

if (UserDB.emailExists(user.getEmail())) {

UserDB.update(user);

} else { // otherwise, write a new record for the user

UserDB.insert(user);

}

// write a new invoice record

InvoiceDB.insert(invoice);

// set the emailCookie in the user's browser.

Cookie emailCookie = new Cookie("emailCookie",

user.getEmail());

emailCookie.setMaxAge(60\*24\*365\*2\*60);

emailCookie.setPath("/");

response.addCookie(emailCookie);

// remove all items from the user's cart

session.setAttribute("cart", null);

// send an email to the user to confirm the order.

String to = user.getEmail();

String from = "confirmation@freshcornrecords.com";

String subject = "Order Confirmation";

String body = "Dear " + user.getFirstName() + ",\n\n" +

"Thanks for ordering from us. " +

"You should receive your order in 3-5 business days. " +

"Please contact us if you have any questions.\n" +

"Have a great day and thanks again!\n\n" +

"Joe King\n" +

"Fresh Corn Records";

boolean isBodyHTML = false;

try {

MailUtil.sendMail(to, from, subject, body, isBodyHTML);

}

catch(MessagingException e) {

this.log(

"Unable to send email. \n" +

"You may need to configure your system as " +

"described in chapter 15. \n" +

"Here is the email you tried to send: \n" +

"=====================================\n" +

"TO: " + to + "\n" +

"FROM: " + from + "\n" +

"SUBJECT: " + subject + "\n" +

"\n" +

body + "\n\n");

}

return "/cart/complete.jsp";

}

}

package music.controllers;

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.http.\*;

import music.business.User;

import music.data.UserDB;

public class UserController extends HttpServlet {

@Override

public void doGet(HttpServletRequest request,

HttpServletResponse response)

throws IOException, ServletException {

String requestURI = request.getRequestURI();

String url = "";

if (requestURI.endsWith("/deleteCookies")) {

url = deleteCookies(request, response);

}

getServletContext()

.getRequestDispatcher(url)

.forward(request, response);

}

@Override

public void doPost(HttpServletRequest request,

HttpServletResponse response)

throws IOException, ServletException {

String requestURI = request.getRequestURI();

String url = "";

if (requestURI.endsWith("/subscribeToEmail")) {

url = subscribeToEmail(request, response);

}

getServletContext()

.getRequestDispatcher(url)

.forward(request, response);

}

private String deleteCookies(HttpServletRequest request,

HttpServletResponse response) {

Cookie[] cookies = request.getCookies();

for (Cookie cookie : cookies) {

cookie.setMaxAge(0); //delete the cookie

cookie.setPath("/"); //entire application

response.addCookie(cookie);

}

return "/delete\_cookies.jsp";

}

private String subscribeToEmail(HttpServletRequest request,

HttpServletResponse response) {

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

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

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

User user = new User();

user.setFirstName(firstName);

user.setLastName(lastName);

user.setEmail(email);

request.setAttribute("user", user);

String url;

String message;

//check that email address doesn't already exist

if (UserDB.emailExists(email)) {

message = "This email address already exists. <br>"

+ "Please enter another email address.";

request.setAttribute("message", message);

url = "/email/index.jsp";

} else {

UserDB.insert(user);

message = "";

request.setAttribute("message", message);

url = "/email/thanks.jsp";

}

return url;

}

}

package music.data;

import javax.persistence.EntityManagerFactory;

import javax.persistence.Persistence;

public class DBUtil {

private static final EntityManagerFactory emf =

Persistence.createEntityManagerFactory("userAdminPU");

public static EntityManagerFactory getEmFactory() {

return emf;

}

}

package music.data;

import javax.persistence.EntityManager;

import javax.persistence.EntityTransaction;

import music.business.Download;

public class DownloadDB {

public static void insert(Download download) {

EntityManager em = DBUtil.getEmFactory().createEntityManager();

EntityTransaction trans = em.getTransaction();

trans.begin();

try {

em.persist(download);

trans.commit();

} catch (Exception e) {

System.out.println(e);

trans.rollback();

} finally {

em.close();

}

}

}

package music.data;

import java.util.List;

import javax.persistence.EntityManager;

import javax.persistence.EntityTransaction;

import javax.persistence.NoResultException;

import javax.persistence.TypedQuery;

import music.business.Invoice;

public class InvoiceDB {

public static void insert(Invoice invoice) {

EntityManager em = DBUtil.getEmFactory().createEntityManager();

EntityTransaction trans = em.getTransaction();

trans.begin();

try {

em.persist(invoice);

trans.commit();

} catch (Exception e) {

System.out.println(e);

trans.rollback();

} finally {

em.close();

}

}

public static void update(Invoice invoice) {

invoice.setIsProcessed(true);

EntityManager em = DBUtil.getEmFactory().createEntityManager();

EntityTransaction trans = em.getTransaction();

trans.begin();

try {

em.merge(invoice);

trans.commit();

} catch (Exception e) {

System.out.println(e);

trans.rollback();

} finally {

em.close();

}

}

public static List<Invoice> selectUnprocessedInvoices() {

EntityManager em = DBUtil.getEmFactory().createEntityManager();

String qString = "SELECT i from Invoice i " +

"WHERE i.isProcessed = false";

TypedQuery<Invoice> q = em.createQuery(qString, Invoice.class);

List<Invoice> results = null;

try {

results = q.getResultList();

} catch (NoResultException ex) {

return null;

} finally {

em.close();

}

return results;

}

}

/\* If the Products table in the music\_jpa database doesn't contain any records,

\* you can use this class to populate the Product table.

\*

\* To do this in NetBeans, press SHIFT+F6

\* while in this window, or right click and select Run File. It is normal for

\* it to take a few minutes to run.

\*

\* After you have done this, you will want to change the

\* javax.persistence.schema-generation.database.action value in the

\* persistence.xml file to "none". Otherwise, the application will pause for

\* two or three minutes each time you start it when it first makes a database

\* call.

\*/

package music.data;

import java.util.HashMap;

import java.util.Map;

import javax.persistence.EntityManager;

import javax.persistence.EntityManagerFactory;

import javax.persistence.EntityTransaction;

import javax.persistence.Persistence;

import javax.persistence.spi.PersistenceUnitTransactionType;

import music.business.Product;

import static org.eclipse.persistence.config.EntityManagerProperties.JDBC\_DRIVER;

import static org.eclipse.persistence.config.EntityManagerProperties.JDBC\_PASSWORD;

import static org.eclipse.persistence.config.EntityManagerProperties.JDBC\_URL;

import static org.eclipse.persistence.config.EntityManagerProperties.JDBC\_USER;

import static org.eclipse.persistence.config.PersistenceUnitProperties.TARGET\_SERVER;

import static org.eclipse.persistence.config.PersistenceUnitProperties.TRANSACTION\_TYPE;

import org.eclipse.persistence.config.TargetServer;

public class PopulateDatabase {

private static EntityManagerFactory emf;

public static void insertProduct(Product product) {

EntityManager em = emf.createEntityManager();

EntityTransaction trans = em.getTransaction();

trans.begin();

try {

em.persist(product);

trans.commit();

} catch (Exception e) {

System.out.println(e);

} finally {

em.close();

}

}

public static void main(String[] args) {

Map props = new HashMap();

props.put(TRANSACTION\_TYPE,

PersistenceUnitTransactionType.RESOURCE\_LOCAL.name());

props.put(JDBC\_DRIVER, "com.mysql.jdbc.Driver");

props.put(JDBC\_URL,

"jdbc:mysql://localhost:3306/music\_jpa?zeroDateTimeBehavior=convertToNull");

props.put(JDBC\_USER, "root");

props.put(JDBC\_PASSWORD, "sesame");

props.put(TARGET\_SERVER, TargetServer.None);

emf = Persistence.createEntityManagerFactory("musicStorePU", props);

Product product1 = new Product();

product1.setCode("8601");

product1.setDescription("86 (the band) - True Life Songs and Pictures");

product1.setPrice(14.95);

insertProduct(product1);

Product product2 = new Product();

product2.setCode("pf01");

product2.setDescription("Paddlefoot - The first CD");

product2.setPrice(12.95);

insertProduct(product2);

Product product3 = new Product();

product3.setCode("pf02");

product3.setDescription("Paddlefoot - The second CD");

product3.setPrice(14.95);

insertProduct(product3);

Product product4 = new Product();

product4.setCode("jr01");

product4.setDescription("Joe Rut - Genuine Wood Grained Finish");

product4.setPrice(14.95);

insertProduct(

product4);

}

}

package music.data;

import java.util.List;

import javax.persistence.EntityManager;

import javax.persistence.EntityTransaction;

import javax.persistence.NoResultException;

import javax.persistence.TypedQuery;

import music.business.Product;

public class ProductDB

{

public static List<Product> selectAll() {

EntityManager em = DBUtil.getEmFactory().createEntityManager();

String qString = "SELECT u FROM Product u";

TypedQuery<Product> q = em.createQuery(qString, Product.class);

List<Product> products;

try {

products = q.getResultList();

} catch (NoResultException e) {

System.out.println(e);

products = null;

} finally {

em.close();

}

return products;

}

public static Product selectByCode(String code) {

EntityManager em = DBUtil.getEmFactory().createEntityManager();

String qString = "SELECT u FROM Product u WHERE u.code = :code";

TypedQuery<Product> q = em.createQuery(qString, Product.class);

q.setParameter("code", code);

Product product = null;

try {

product = q.getSingleResult();

} catch (NoResultException e) {

System.out.println(e);

product = null;

} finally {

em.close();

}

return product;

}

public static void update(Product product) {

EntityManager em = DBUtil.getEmFactory().createEntityManager();

EntityTransaction trans = em.getTransaction();

try {

trans.begin();

em.merge(product);

trans.commit();

} catch (Exception e) {

System.out.println(e);

trans.rollback();

} finally {

em.close();

}

}

public static void insert(Product product) {

EntityManager em = DBUtil.getEmFactory().createEntityManager();

EntityTransaction trans = em.getTransaction();

try {

trans.begin();

em.persist(product);

trans.commit();

} catch (Exception e) {

System.out.println(e);

trans.rollback();

} finally {

em.close();

}

}

public static void delete(Product product) {

EntityManager em = DBUtil.getEmFactory().createEntityManager();

EntityTransaction trans = em.getTransaction();

try {

trans.begin();

em.remove(em.merge(product));

trans.commit();

} catch (Exception e) {

System.out.println(e);

trans.rollback();

} finally {

em.close();

}

}

public static Product selectProduct(String code)

{

EntityManager em = DBUtil.getEmFactory().createEntityManager();

String qString = "SELECT p FROM Product p " +

"WHERE p.code = :code";

TypedQuery<Product> q = em.createQuery(qString, Product.class);

q.setParameter("code", code);

Product result = null;

try {

result = q.getSingleResult();

} catch (NoResultException ex) {

return null;

} finally {

em.close();

}

return (Product)result;

}

public static Product selectProduct(long productId) {

EntityManager em = DBUtil.getEmFactory().createEntityManager();

return em.find(Product.class, productId);

}

public List<Product> selectProducts() {

EntityManager em = DBUtil.getEmFactory().createEntityManager();

String qString = "SELECT p from Product p";

TypedQuery<Product> q = em.createQuery(qString, Product.class);

List<Product> results = null;

try {

results = q.getResultList();

} catch (NoResultException ex) {

return null;

} finally {

em.close();

}

return results;

}

}

package music.data;

import java.util.List;

import javax.persistence.EntityManager;

import javax.persistence.EntityTransaction;

import javax.persistence.NoResultException;

import javax.persistence.TypedQuery;

import music.business.Product;

public class ProductDB

{

public static List<Product> selectAll() {

EntityManager em = DBUtil.getEmFactory().createEntityManager();

String qString = "SELECT u FROM Product u";

TypedQuery<Product> q = em.createQuery(qString, Product.class);

List<Product> products;

try {

products = q.getResultList();

} catch (NoResultException e) {

System.out.println(e);

products = null;

} finally {

em.close();

}

return products;

}

public static Product selectByCode(String code) {

EntityManager em = DBUtil.getEmFactory().createEntityManager();

String qString = "SELECT u FROM Product u WHERE u.code = :code";

TypedQuery<Product> q = em.createQuery(qString, Product.class);

q.setParameter("code", code);

Product product = null;

try {

product = q.getSingleResult();

} catch (NoResultException e) {

System.out.println(e);

product = null;

} finally {

em.close();

}

return product;

}

public static void update(Product product) {

EntityManager em = DBUtil.getEmFactory().createEntityManager();

EntityTransaction trans = em.getTransaction();

try {

trans.begin();

em.merge(product);

trans.commit();

} catch (Exception e) {

System.out.println(e);

trans.rollback();

} finally {

em.close();

}

}

public static void insert(Product product) {

EntityManager em = DBUtil.getEmFactory().createEntityManager();

EntityTransaction trans = em.getTransaction();

try {

trans.begin();

em.persist(product);

trans.commit();

} catch (Exception e) {

System.out.println(e);

trans.rollback();

} finally {

em.close();

}

}

public static void delete(Product product) {

EntityManager em = DBUtil.getEmFactory().createEntityManager();

EntityTransaction trans = em.getTransaction();

try {

trans.begin();

em.remove(em.merge(product));

trans.commit();

} catch (Exception e) {

System.out.println(e);

trans.rollback();

} finally {

em.close();

}

}

public static Product selectProduct(String code)

{

EntityManager em = DBUtil.getEmFactory().createEntityManager();

String qString = "SELECT p FROM Product p " +

"WHERE p.code = :code";

TypedQuery<Product> q = em.createQuery(qString, Product.class);

q.setParameter("code", code);

Product result = null;

try {

result = q.getSingleResult();

} catch (NoResultException ex) {

return null;

} finally {

em.close();

}

return (Product)result;

}

public static Product selectProduct(long productId) {

EntityManager em = DBUtil.getEmFactory().createEntityManager();

return em.find(Product.class, productId);

}

public List<Product> selectProducts() {

EntityManager em = DBUtil.getEmFactory().createEntityManager();

String qString = "SELECT p from Product p";

TypedQuery<Product> q = em.createQuery(qString, Product.class);

List<Product> results = null;

try {

results = q.getResultList();

} catch (NoResultException ex) {

return null;

} finally {

em.close();

}

return results;

}

}

package music.data;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.List;

import javax.persistence.EntityManager;

import javax.persistence.NoResultException;

import javax.persistence.TypedQuery;

import music.business.Download;

import music.business.User;

import org.apache.poi.hssf.usermodel.HSSFWorkbook;

import org.apache.poi.ss.usermodel.Row;

import org.apache.poi.ss.usermodel.Sheet;

import org.apache.poi.ss.usermodel.Workbook;

public class ReportDB {

public static Workbook getUserEmail() {

// get the data from the database

EntityManager em = DBUtil.getEmFactory().createEntityManager();

String qString = "SELECT u FROM User u " +

"ORDER BY u.lastName";

TypedQuery<User> q = em.createQuery(qString, User.class);

List<User> users = null;

try {

users = q.getResultList();

} catch (NoResultException e) {

System.err.println(e);

return null;

} finally {

em.close();

}

// create the workbook, its worksheet, and its title row

Workbook workbook = new HSSFWorkbook();

Sheet sheet = workbook.createSheet("User Email Report");

Row row = sheet.createRow(0);

row.createCell(0).setCellValue("The User Email report");

// create the header row

row = sheet.createRow(2);

row.createCell(0).setCellValue("LastName");

row.createCell(1).setCellValue("FirstName");

row.createCell(2).setCellValue("Email");

row.createCell(3).setCellValue("CompanyName");

row.createCell(4).setCellValue("Address1");

row.createCell(5).setCellValue("Address2");

row.createCell(6).setCellValue("City");

row.createCell(7).setCellValue("State");

row.createCell(8).setCellValue("Zip");

row.createCell(9).setCellValue("Country");

row.createCell(10).setCellValue("UserID");

// create the data rows

int i = 3;

for (User u : users) {

row = sheet.createRow(i);

row.createCell(0).setCellValue(u.getLastName());

row.createCell(1).setCellValue(u.getFirstName());

row.createCell(2).setCellValue(u.getEmail());

row.createCell(3).setCellValue(u.getCompanyName());

row.createCell(4).setCellValue(u.getAddress1());

row.createCell(5).setCellValue(u.getAddress2());

row.createCell(6).setCellValue(u.getCity());

row.createCell(7).setCellValue(u.getState());

row.createCell(8).setCellValue(u.getZip());

row.createCell(9).setCellValue(u.getCountry());

row.createCell(10).setCellValue(u.getId());

i++;

}

return workbook;

}

public static Workbook getDownloadDetail(

String startDate, String endDate) {

// get the data from the database

EntityManager em = DBUtil.getEmFactory().createEntityManager();

String qString = "SELECT d from Download d " +

"WHERE d.downloadDate >= :startDate AND " +

"d.downloadDate <= :endDate ORDER BY d.downloadDate DESC";

TypedQuery <Download> q = em.createQuery(qString, Download.class);

List<Download> downloads = null;

SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");

try {

q.setParameter("startDate", dateFormat.parse(startDate));

q.setParameter("endDate", dateFormat.parse(endDate));

downloads = q.getResultList();

} catch (ParseException e) {

System.err.println(e.toString());

return null;

} finally {

em.close();

}

// create the workbook, its worksheet, and its title row

Workbook workbook = new HSSFWorkbook();

Sheet sheet = workbook.createSheet("Download Report");

Row row = sheet.createRow(0);

row.createCell(0).setCellValue("The Download report");

// create the header rows

row = sheet.createRow(2);

row.createCell(0).setCellValue("Start Date: " + startDate);

row = sheet.createRow(3);

row.createCell(0).setCellValue("End Date: " + endDate);

row = sheet.createRow(5);

row.createCell(0).setCellValue("DownloadDate");

row.createCell(1).setCellValue("ProductCode");

row.createCell(2).setCellValue("Email");

row.createCell(3).setCellValue("FirstName");

row.createCell(4).setCellValue("LastName");

// create the data rows

int total = 0;

int i = 6;

for (Download download : downloads) {

row = sheet.createRow(i);

row.createCell(0).setCellValue(download.getDownloadDate().toString());

row.createCell(1).setCellValue(download.getProductCode());

row.createCell(2).setCellValue(download.getUser().getEmail());

row.createCell(3).setCellValue(download.getUser().getFirstName());

row.createCell(4).setCellValue(download.getUser().getLastName());

total++;

i++;

}

row = sheet.createRow(i+1);

row.createCell(0).setCellValue("Total Number of Downloads: " + total);

return workbook;

}

}

package music.data;

import java.util.List;

import javax.persistence.EntityManager;

import javax.persistence.EntityTransaction;

import javax.persistence.NoResultException;

import javax.persistence.TypedQuery;

import music.business.User;

public class UserDB {

public static void insert(User user) {

EntityManager em = DBUtil.getEmFactory().createEntityManager();

EntityTransaction trans = em.getTransaction();

trans.begin();

try {

em.persist(user);

trans.commit();

} catch (Exception e) {

System.out.println(e);

trans.rollback();

} finally {

em.close();

}

}

public static void update(User user) {

EntityManager em = DBUtil.getEmFactory().createEntityManager();

EntityTransaction trans = em.getTransaction();

trans.begin();

try {

em.merge(user);

trans.commit();

} catch (Exception e) {

System.out.println(e);

trans.rollback();

} finally {

em.close();

}

}

public static void delete(User user) {

EntityManager em = DBUtil.getEmFactory().createEntityManager();

EntityTransaction trans = em.getTransaction();

trans.begin();

try {

em.remove(em.merge(user));

trans.commit();

} catch (Exception e) {

System.out.println(e);

trans.rollback();

} finally {

em.close();

}

}

public static User selectUser(String email) {

EntityManager em = DBUtil.getEmFactory().createEntityManager();

String qString = "SELECT u FROM User u " +

"WHERE u.email = :email";

TypedQuery<User> q = em.createQuery(qString, User.class);

q.setParameter("email", email);

try {

User user = q.getSingleResult();

return user;

} catch (NoResultException e) {

return null;

} finally {

em.close();

}

}

public static boolean emailExists(String email) {

User u = selectUser(email);

return u != null;

}

public static List<User> selectUsers() {

EntityManager em = DBUtil.getEmFactory().createEntityManager();

String qString = "SELECT u from User u";

TypedQuery<User> q = em.createQuery(qString, User.class);

List<User> users;

try {

users = q.getResultList();

if (users == null || users.isEmpty())

users = null;

} finally {

em.close();

}

return users;

}

}

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package music.tags;

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

import javax.servlet.jsp.\*;

import javax.servlet.jsp.tagext.\*;

import java.io.\*;

import java.util.regex.Pattern;

import static javax.servlet.jsp.tagext.Tag.SKIP\_BODY;

public class IfEmptyMarkTag extends TagSupport {

private String field;

private String color = "blue";

public void setField(String field) {

this.field = field;

}

public void setColor(String color) {

this.color = color;

}

@Override

public int doStartTag() throws JspException

{

boolean match = false;

try

{

JspWriter out = pageContext.getOut();

String doublePattern = "([0-9].\*)\\.([0-9].\*)";

match = Pattern.matches(doublePattern, field);

if(match)

{

if(Double.valueOf(field)<=0)

{

match = true;

}

else

{

//check if 2 decimal places

String[] splitter = (field.trim().split("\\."));

splitter[0].length(); // Before Decimal Count

int decimalLength = splitter[1].length(); // After Decimal Count

if (decimalLength == 2)

match = false;

else

match = true;

}

}

if ((field == null || field.length() == 0)|| match)

{

out.print("<font color=" + color + "> \*</font>");

}

} catch (IOException ioe) {

System.out.println(ioe);

}

return SKIP\_BODY;

}

}

package music.util;

import javax.servlet.http.\*;

public class CookieUtil {

public static String getCookieValue(

Cookie[] cookies, String cookieName) {

String cookieValue = "";

if (cookies != null) {

for (Cookie cookie: cookies) {

if (cookieName.equals(cookie.getName())) {

cookieValue = cookie.getValue();

}

}

}

return cookieValue;

}

}

package music.util;

import java.util.Properties;

import javax.mail.\*;

import javax.mail.internet.\*;

public class MailUtil {

public static void sendMail(String to, String from,

String subject, String body, boolean bodyIsHTML)

throws MessagingException {

// 1 - get a mail session

Properties props = new Properties();

//props.put("mail.transport.protocol", "smtp");

props.put("mail.smtp.host", "localhost");

//props.put("mail.smtp.port", 25);

Session session = Session.getDefaultInstance(props);

session.setDebug(true);

// 2 - create a message

Message message = new MimeMessage(session);

message.setSubject(subject);

if (bodyIsHTML) {

message.setContent(body, "text/html");

} else {

message.setText(body);

}

// 3 - address the message

Address fromAddress = new InternetAddress(from);

Address toAddress = new InternetAddress(to);

message.setFrom(fromAddress);

message.setRecipient(Message.RecipientType.TO, toAddress);

// 4 - send the message

Transport.send(message);

}

}

package music.util;

import javax.servlet.\*;

import java.util.\*;

public class MusicStoreContextListener implements ServletContextListener {

@Override

public void contextInitialized(ServletContextEvent event) {

ServletContext sc = event.getServletContext();

// get the absolute paths for swithing regular and secure connections

String contextPath = sc.getContextPath();

String absolutePath = "http://localhost:8080" + contextPath;

String absolutePathSecure = "https://localhost:8443" + contextPath;

sc.setAttribute("absolutePath", absolutePath);

sc.setAttribute("absolutePathSecure", absolutePathSecure);

// initialize the customer service email address that's used throughout the web site

String custServEmail = sc.getInitParameter("custServEmail");

sc.setAttribute("custServEmail", custServEmail);

// initialize the current year that's used in the copyright notice

GregorianCalendar currentDate = new GregorianCalendar();

int currentYear = currentDate.get(Calendar.YEAR);

sc.setAttribute("currentYear", currentYear);

// initialize the array of years that's used for the credit card year

ArrayList<String> creditCardYears = new ArrayList<>();

for (int i = 0; i < 8; i++) {

int year = currentYear + i;

String yearString = Integer.toString(year);

creditCardYears.add(yearString);

}

sc.setAttribute("creditCardYears", creditCardYears);

}

@Override

public void contextDestroyed(ServletContextEvent event) {

// no cleanup necessary

}

}