# Project Name: Make an E-commerce Website for Sporty Shoes.

## GitHub Link:

## <https://github.com/tamasjit/myjavaproject/tree/master/Phase%203/Phase-3_Final%20Assesment/SportyShoes>

## Source Code:

## Webapp:

### Admin:

#### Catagories.jsp:

<%@ page language="java" contentType="text/html; charset=UTF-8"

    pageEncoding="UTF-8"%>

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

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

<%@ page isELIgnored="false" %>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Admin - Setup Product Categories</title>

<link rel="stylesheet" href="css/bootstrap.min.css">

</head>

<body>

<jsp:include page="/components/header1.jsp" ></jsp:include>

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

<jsp:include page="/components/admin-topbar.jsp" ></jsp:include>

<div class="container">

Total Categories: ${list.size()} &nbsp;&nbsp; <a class="btn btn-warning" style="text-align:right" href="admineditcat?id=0">Add Category</a><br>

<table class="table table-dark">

  <thead>

    <tr>

      <th scope="col">Category</th>

    </tr>

  </thead>

  <tbody>

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

    <tr>

      <th scope="row">${item.name }</th>

      <td>

      <a href="admineditcat?id=${item.ID}">Edit</a> | <a href="admindeletecat?id=${item.ID}">Delete</a>

    </td>

    </tr>

  </c:forEach>

  </tbody>

</table>

</div>

<jsp:include page="/components/admin-footer.jsp"></jsp:include>

</body>

</html>

#### Change-password.jsp:

<%@ page language="java" contentType="text/html; charset=UTF-8"

    pageEncoding="UTF-8"%>

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

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

<%@ page isELIgnored="false" %>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Admin - Change Password</title>

<link rel="stylesheet" href="css/bootstrap.min.css">

</head>

<body>

<jsp:include page="/components/header1.jsp" ></jsp:include>

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

<jsp:include page="/components/admin-topbar.jsp" ></jsp:include>

<form name=frmPwd method=post action="adminchangepwdaction">

 <div class="container">

 <p class="alert alert-danger">${error }</p>

<table class="table table-dark" style="border:1px solid black;">

  <tbody>

    <tr>

      <td scope="col">Enter new password\*</td>

      <td><input name=pwd maxlength=10 type="password"></td>

    </tr>

    <tr>

      <td width=25%>Confirm new Password\*</td>

      <td>

    <input name=pwd2 maxlength=10 type="password">

    </td>

    </tr>

  </tbody>

</table>

    <p style="">

      <button class="btn btn-warning">Login</button>

    </p>

</div>

</form>

<jsp:include page="/components/admin-footer.jsp"></jsp:include>

</body>

</html>

#### Dashboard.jsp:

<%@ page language="java" contentType="text/html; charset=UTF-8"

    pageEncoding="UTF-8"%>

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

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

<%@ page isELIgnored="false" %>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Admin Dashboard</title>

<link rel="stylesheet" href="css/bootstrap.min.css">

</head>

<body>

<jsp:include page="/components/header1.jsp" ></jsp:include>

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

<jsp:include page="/components/admin-topbar.jsp" ></jsp:include>

<jsp:include page="/components/admin-footer.jsp"></jsp:include>

</body>

</html>

#### Edit-category.jsp:

<%@ page language="java" contentType="text/html; charset=UTF-8"

    pageEncoding="UTF-8"%>

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

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

<%@ page isELIgnored="false" %>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Admin - Add/Edit Product Category</title>

<link rel="stylesheet" href="css/bootstrap.min.css">

</head>

<body>

<jsp:include page="/components/header1.jsp" ></jsp:include>

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

<jsp:include page="/components/admin-topbar.jsp" ></jsp:include>

<div class="container">

<%

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

    out.print(request.getParameter("error"));

%>

<form name=frmCat method=post action="admineditcataction">

    <input type=hidden name=id value=${category.ID}>

<table border=1 class="table table-bordered table-dark">

    <tr>

        <td width=25%>Category name\*</td>

        <td><input name=name  maxlength=50 value="${category.name}"></td>

    </tr>

    <tr>

        <td colspan=2>

            <button class="btn btn-warning">Save</button>

        </td>

    </tr>

 </table>

</form>

</div>

<jsp:include page="/components/admin-footer.jsp"></jsp:include>

</body>

</html>

#### edit-product.jsp:

<%@ page language="java" contentType="text/html; charset=UTF-8"

    pageEncoding="UTF-8"%>

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

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

<%@ page isELIgnored="false" %>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Admin - Add/Edit Product</title>

<link rel="stylesheet" href="css/bootstrap.min.css">

</head>

<body>

<jsp:include page="/components/header1.jsp" ></jsp:include>

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

<jsp:include page="/components/admin-topbar.jsp" ></jsp:include>

<div class="container">

<p class="alert alert-danger">

<%

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

    out.print(request.getParameter("error"));

%>

</p>

<form name=frmProduct method=post action="admineditproductaction">

    <input type=hidden name=id value="${product.ID }">

<table class="table table-bordered table-dark">

    <tr>

        <td width=25%>Product name\*</td>

        <td><input name=name  maxlength=100 value="${product.name }"></td>

    </tr>

    <tr>

        <td width=25%>Price\*</td>

        <td><input name=price type="numeric"  maxlength=6 value="${product.price }"></td>

    </tr>

    <tr>

        <td width=25%>Category\*</td>

        <td>

            <select name=category>

                <option value="0">Select category</option>

                ${catDropdown}

            </select>

        </td>

    </tr>

    <tr>

        <td colspan=2>

            <button class="btn btn-warning">Save</button>

        </td>

    </tr>

 </table>

</form>

</div>

<jsp:include page="/components/admin-footer.jsp"></jsp:include>

</body>

</html>

#### Login.jsp:

<%@ page language="java" contentType="text/html; charset=UTF-8"

    pageEncoding="UTF-8"%>

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

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

<%@ page isELIgnored="false" %>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Admin Login</title>

<link rel="stylesheet" href="css/bootstrap.min.css">

</head>

<body>

<jsp:include page="/components/header1.jsp" ></jsp:include>

<br>

<br>

<br>

<br>

<br>

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

${error}<br>

<form name=frmLogin action="adminloginaction" method="post">

 <table border=1 cellspacing=2 cellpadding=4>

    <tr>

        <td width=25%>Admin id\*</td>

        <td><input name=admin\_id maxlength=20></td>

    </tr>

    <tr>

        <td width=25%>Admin Password\*</td>

        <td><input name=admin\_pwd maxlength=10 type="password"></td>

    </tr>

    <tr>

        <td colspan=2>

            <button>Login</button>

        </td>

    </tr>

 </table>

</form>

<jsp:include page="/components/admin-footer.jsp"></jsp:include>

</body>

</html>

#### Members.jsp:

<%@ page language="java" contentType="text/html; charset=UTF-8"

    pageEncoding="UTF-8"%>

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

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

<%@ page isELIgnored="false" %>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Admin - Browse Members</title>

<link rel="stylesheet" href="css/bootstrap.min.css">

</head>

<body>

<jsp:include page="/components/header1.jsp" ></jsp:include>

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

<jsp:include page="/components/admin-topbar.jsp" ></jsp:include>

<div class="container">

Total Members: ${list.size() }

<table border=1 class="table table-bordered table-dark">

    <tr>

        <td><b>Name</b></td>

        <td><b>Email</b></td>

        <td><b>Age</b></td>

        <td><b>Address</b></td>

    </tr>

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

        <tr>

            <td>${item.fname}&nbsp;${item.lname }</td>

            <td>${item.email}</td>

            <td>${item.age}</td>

            <td>${item.address}</td>

        </tr>

      </c:forEach>

</table>

</div>

<jsp:include page="/components/admin-footer.jsp"></jsp:include>

</body>

</html>

#### Products.jsp:

<%@ page language="java" contentType="text/html; charset=UTF-8"

    pageEncoding="UTF-8"%>

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

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

<%@ page isELIgnored="false" %>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Admin Setup Products</title>

<link rel="stylesheet" href="css/bootstrap.min.css">

</head>

<body>

<jsp:include page="/components/header1.jsp" ></jsp:include>

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

<jsp:include page="/components/admin-topbar.jsp" ></jsp:include>

<div class="container">

Total Products: ${list.size()}  &nbsp;&nbsp; <a class="btn btn-warning" style="text-align:center;" href="admineditproduct?id=0">Add Product</a><br>

<table class="table table-bordered table-dark">

    <tr>

        <td><b>Product</b></td>

        <td><b>Price</b></td>

        <td><b>Category</b></td>

        <td></td>

    </tr>

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

        <tr>

            <td>${item.name }</td>

            <td>${item.price }</td>

            <td>${mapCats.get(item.ID)}</td>

            <td>

                <a href="admineditproduct?id=${item.ID}">Edit</a> | <a href="admindeleteproduct?id=${item.ID}">Delete</a>

            </td>

        </tr>

      </c:forEach>

</table>

</div>

<jsp:include page="/components/admin-footer.jsp"></jsp:include>

</body>

</html>

#### Purchases.jsp:

<%@ page language="java" contentType="text/html; charset=UTF-8"

    pageEncoding="UTF-8"%>

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

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

<%@ page isELIgnored="false" %>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Admin - Purchases Report</title>

<link rel="stylesheet" href="css/bootstrap.min.css">

</head>

<body>

<jsp:include page="/components/header1.jsp" ></jsp:include>

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

<jsp:include page="/components/admin-topbar.jsp" ></jsp:include>

<div class="container">

<br><br>Total ${list.size()} Orders: ${totalAmount }<br>

<table class="table table-bordered table-dark">

    <tr>

        <td><b>Order ID</b></td>

        <td><b>User</b>

        <td><b>Date</b></td>

        <td><b>Total</b></td>

        <td><b>Items</b></td>

    </tr>

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

        <tr>

            <td>${item.ID }</td>

            <td>

                ${mapUsers.get(item.ID)}

            </td>

            <td>${item.date }</td>

            <td>${item.total}</td>

            <td>

                ${mapItems.get(item.ID)}

            </td>

        </tr>

      </c:forEach>

</table>

</div>

<jsp:include page="/components/admin-footer.jsp"></jsp:include>

</body>

</html>

### Components:

#### Admin-footer.jsp:

<hr>

#### Admin-header.jsp:

<br>

<br>

<%@ page isELIgnored="false" %>

#### Admin-topbar.jsp:

<div class="container-fluid p-3 my-3 bg-dark text-white">

<h1 class="text-center">Admin Dashboard</h2>

<div class="text-center">

<button type="button" class="btn btn-light"><a class="text-success p-3" href="adminproducts" style="font-size: 20px; font-weight: bold;">Setup Products</a></button>

<button type="button" class="btn btn-light"><a class="text-success p-3" href="admincategories" style="font-size: 20px; font-weight: bold;">Setup Product Categories</a></button>

<button type="button" class="btn btn-light"><a class="text-success p-3" href="adminmembers" style="font-size: 20px; font-weight: bold;">Browse Members</a></button>

<button type="button" class="btn btn-light"><a class="text-success p-3" href="adminpurchases" style="font-size: 20px; font-weight: bold;">Purchase Report</a></button>

<button type="button" class="btn btn-light"><a class="text-success p-3" href="adminchangepassword" style="font-size: 20px; font-weight: bold;">Change Password</a></button>

</div>

</div>

<br>

#### Footer.jsp:

<hr>

<div class="d-flex flex-column h-100">

<footer class="footer mt-auto py-3 bg-light" style=

  "position: fixed;

  left: 0;

  bottom: 0;

  width: 100%;

  background-color: red;

  color: white;

  text-align: center;">

  <div class="container">

    <span class="text-muted">�Sportyshoestore.com(Tamasjit Ghoshal)</span>

  </div>

</footer>

</div>

#### Header.jsp:

<br>

<br>

<%@ page isELIgnored="false" %>

<h3 class="" style="text-align: center;"> ${pageTitle} </h3>

#### Header1.jsp:

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

    pageEncoding="ISO-8859-1"%>

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

<%@ page isELIgnored="false" %>

<nav class="navbar navbar-expand-lg fixed-top navbar-light headerback"  style="background-color: #ffffff;">

  <a class="navbar-brand text-success logo" href="<%=request.getContextPath()%>/home" style="font-size: 31px; font-weight: bold; ">SportyShoes</a>

  <div class="collapse navbar-collapse" id="navbarSupportedContent">

    <ul class="navbar-nav mr-auto">

      <li class="nav-item active">

        <a class="nav-link text-success" href="<%=request.getContextPath()%>/home" style="font-size: 25px; font-weight: bold; ">Home</a>

      </li>

    </ul>

    <form class="form-inline my-2 my-lg-0">

        <c:if test = "${user\_id != null }">

            <a style="margin-right: 6px; font-weight: bold;" class="btn btn-outline-success my-2 my-sm-0 buttonstylehome" type="submit" href="/dashboard">User Dashboard</a>

            <a style="margin-right: 6px; font-weight: bold;" class="btn btn-outline-success my-2 my-sm-0 buttonstylehome" type="submit" href="/logout">Logout</a>

        </c:if>

        <c:if test = "${admin\_id != null }">

            <a style="margin-right: 6px; font-weight: bold;" class="btn btn-outline-success my-2 my-sm-0 buttonstylehome" type="submit" href="/admindashboard">Admin Dashboard</a>

            <a style="margin-right: 6px; font-weight: bold;" class="btn btn-outline-success my-2 my-sm-0 buttonstylehome" type="submit" href="/logout">Logout</a>

        </c:if>

            <c:if test ="${user\_id == null && admin\_id == null}">

            <a style="margin-right: 6px; font-weight: bold;" class="btn btn-outline-success my-2 my-sm-0 login" type="submit" href="/login">Login</a>

        </c:if>

    </form>

  </div>

</nav>

#### Topbar.jsp:

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

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

<%@ page isELIgnored="false" %>

<div class="container">

<p style="text-align:center">

<% if (session.getAttribute("user\_id") == null ) { %>

    <a class="btn btn-primary" style="margin-right:9px;" href="login">Login/Signup</a> | <a href="adminlogin">Adminlogin</a>

<% }  else { %>

    <a class="btn btn-primary" style="margin-right:9px;" href="cart">Cart</a>

    <a class="btn btn-primary" style="margin-right:9px;" href="editprofile">Edit Profile</a>

    <a class="btn btn-primary" style="margin-right:9px;" href="memberpurchases">Your Orders</a>

<% }  %>

<br>

</p>

</div>

### Css:

#### Cart.jsp:

<%@ page language="java" contentType="text/html; charset=UTF-8"

    pageEncoding="UTF-8"%>

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

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

<%@ page isELIgnored="false"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Sporty Shoes - Your Cart</title>

<link rel="stylesheet" href="css/bootstrap.min.css">

<style>

.contain {

    padding: 0px 25%;

}

</style>

</head>

<body>

    <jsp:include page="/components/header1.jsp"></jsp:include>

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

    <jsp:include page="/components/topbar.jsp"></jsp:include>

    <div class="container">

        <div class="contain">

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

                <p class="alert alert-danger">

                    <c:out value="${error}" />

                <p>

            </c:if>

            Total Cart Value: ${cartValue }<br>

            <p style="color: orange; text-align: right">Total ${list.size()}

                Orders: ${totalAmount }</p>

            <table class="table">

                <thead>

                    <tr>

                        <th scope="col">Product</th>

                        <th scope="col">Qty</th>

                        <th scope="col">Price</th>

                    </tr>

                </thead>

                <tbody>

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

                        <tr>

                            <th scope="row">${item.name }</th>

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

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

                        </tr>

                    </c:forEach>

                </tbody>

            </table>

            <br>

            <c:if test="${cartItems.size() > 0}">

                <a class="btn btn-warning" href="checkout">Checkout Now</a>

            </c:if>

        </div>

    </div>

</body>

</html>

#### Checkout.jsp:

<%@ page language="java" contentType="text/html; charset=UTF-8"

    pageEncoding="UTF-8"%>

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

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

<%@ page isELIgnored="false" %>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Sporty Shoes - Checkout</title>

<link rel="stylesheet" href="css/bootstrap.min.css">

</head>

<body>

<jsp:include page="/components/header1.jsp" ></jsp:include>

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

<jsp:include page="/components/topbar.jsp" ></jsp:include>

<br>

<div class="container">

<h3 style="text-align:center; color:orange;">Your Total Order is worth ${cartValue}</h3><br><br>

<p style="text-align:center;"><a class="alert alert-warning" href="gateway">Pay via secure Payment Gateway</a></p>

</div>

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

</body>

</html>

#### Confirm.jsp:

<%@ page language="java" contentType="text/html; charset=UTF-8"

    pageEncoding="UTF-8"%>

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

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

<%@ page isELIgnored="false" %>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Sporty Shoes - Purchase Confirmation</title>

<link rel="stylesheet" href="css/bootstrap.min.css">

</head>

<body>

<jsp:include page="/components/header1.jsp" ></jsp:include>

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

<jsp:include page="/components/topbar.jsp" ></jsp:include>

<br><br>

<div class="container">

<p style="text-align:center; color:orange;">Your Order worth ${cartValue} has been completed successfully.</p><br><br>

<p style="text-align:center;"> Check your <A class="alert alert-warning" href="memberpurchases">Order History</A> </p>

</div>

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

</body>

</html>

#### Dashboard.jsp:

<%@ page language="java" contentType="text/html; charset=UTF-8"

    pageEncoding="UTF-8"%>

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

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

<%@ page isELIgnored="false" %>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Sporty Shoes - Dashboard</title>

<link rel="stylesheet" href="css/bootstrap.min.css">

</head>

<body>

<jsp:include page="/components/header1.jsp" ></jsp:include>

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

<jsp:include page="/components/topbar.jsp" ></jsp:include>

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

</body>

</html>

#### Edit-profile.jsp:

<%@ page language="java" contentType="text/html; charset=UTF-8"

    pageEncoding="UTF-8"%>

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

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

<%@ page isELIgnored="false" %>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Sporty Shoes - Edit Profile</title>

<link rel="stylesheet" href="css/bootstrap.min.css">

<style>

.detailsbox{

    padding:80px 25%;

}

</style>

</head>

<body>

<jsp:include page="/components/header1.jsp" ></jsp:include>

    <div class="detailsbox">

    <div class="margin p-3 my-3 bg-dark text-white">

        <form name=frmEdit action="editprofileaction" method="post">

          <h4>Update Profile</h4>

          <div class="row">

          <input type=hidden name=user\_id value="${user.ID}">

          <div class="form-group col">

            <label for="fname">First name</label>

            <input type="text" class="form-control" id="fname" name="fname"   maxlength="20" placeholder="Enter first name" value="${user.fname }" required>

          </div>

          <div class="form-group col">

            <label for="lname">Last name</label>

            <input type="text" class="form-control" id="lname" name="lname" maxlength="20" placeholder="Enter last name" value="${user.lname }" required>

          </div>

          </div>

          <div class="form-group">

            <label for="age">Age</label>

            <input type="tel" class="form-control" id="age" name="age" placeholder="Enter Age" value="${user.age }" required >

          </div>

          <div class="form-group">

            <label for="address">Address</label>

            <input type="text" class="form-control" id="address" name="address" maxlength="100" placeholder="Enter Address" value="${user.address }" required>

          </div>

          <div class="form-group">

            <label for="pwd">Password</label>

            <input type="text" class="form-control" id="pwd" name="pwd" maxlength="20" placeholder="Enter password" required>

          </div>

          <div class="form-group">

            <label for="pwd2">Confirm Password</label>

            <input type="text" class="form-control" id="pwd2" name="pwd2" maxlength="20" placeholder="Enter password" required>

          </div>

          <div class="">

              <small id="phone" class="form-text text-muted">We'll never share your phone no and email with anyone else.</small>

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

              <h5 class="alert alert-danger"><c:out value="${error}"/></h5>

              </c:if>

              <button type="submit" class="btn btn-primary"> UPDATE </button>

          </div>

        </form>

    </div>

    </div>

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

</body>

</html>

#### Gateway.jsp:

<%@ page language="java" contentType="text/html; charset=UTF-8"

    pageEncoding="UTF-8"%>

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

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

<%@ page isELIgnored="false" %>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Sporty Shoes - Make Payment</title>

<link rel="stylesheet" href="css/bootstrap.min.css">

</head>

<body>

<jsp:include page="/components/header1.jsp" ></jsp:include>

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

<jsp:include page="/components/topbar.jsp" ></jsp:include>

<br><br>

<div class="container">

<p style="color:orange; text-align:center;">Your card will be charged an amount of ${cartValue}</p><br><br>

<p style="text-align:center;"><a class="alert alert-warning" href="completepurchase">Click to complete checkout</a></p>

</div>

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

</body>

</html>

#### Hellopage.jsp:

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

    pageEncoding="ISO-8859-1"%>

<!DOCTYPE HTML>

<html>

<head>

    <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

    <title>Sping MVC Web Application</title>

</head>

<body>

    <h2>Spring MVC Example</h2>

    <div id="welcomeMessage" style="margin: 20px; color: green;">

        <strong>${messageObj}</strong>

    </div>

</body>

</html>

#### Index.jsp:

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

    pageEncoding="ISO-8859-1"%>

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

<%@ page isELIgnored="false"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Sporty Shoes</title>

<link rel="stylesheet" href="css/bootstrap.min.css">

</head>

<style>

.box {

    height: 200px;

    width: 200px;

    background-image: url("");

}

.contain {

    padding: 0px 10%

}

</style>

<body>

    <jsp:include page="/components/header1.jsp"></jsp:include>

    <jsp:include page="/components/topbar.jsp"></jsp:include>

    <div class="contain">

        <div class="row" style="margin-top: 85px;">

            <div class="col-12">

                <div class="row">

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

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

                            <div class="col-4 mt-2 mb-2">

                                <div class="card text-white" style="background-color: #FE957B">

                                    <div class="card-body">

                                        <h3 style="margin-top: 16px;">${item.name }</h3>

                                        <h4 style="">Rs: ${item.price }</h4>

                                        <a class="btn btn-warning" href="cartadditem?id=${item.ID}">Add

                                            To Cart</a>

                                    </div>

                                </div>

                            </div>

                        </c:forEach>

                    </c:if>

                </div>

            </div>

        </div>

    </div>

</body>

</html>

#### Login.jsp:

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

    pageEncoding="ISO-8859-1"%>

<!DOCTYPE html>

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

<%@ page isELIgnored="false" %>

<html>

<head>

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

<title>Insert title here</title>

<link rel="stylesheet" href="css/bootstrap.min.css">

<style>

.contain{

    margin:0px 15%;

}

.topmar{

    margin:150px 0%;

}

</style>

</head>

<body>

<jsp:include page="/components/header1.jsp" ></jsp:include>

    <div class="container">

    <div class="row topmar">

    <div class="col-6" >

        <div class="contain p-3 my-3 bg-dark text-white">

        <h2 style="text-align:center">Login</h2>

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

        <p class="alert alert-warning"><c:out value="${regsucmessage}"/></p>

        </c:if>

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

            <div class="form-group">

                <label for="email\_id">User Name:</label>

                <input type="text" class="form-control" id="email\_id" placeholder="User Name" name="email\_id" required>

            </div>

            <div class="form-group">

                <label for="pwd">Password:</label>

                <input type="password" class="form-control" id="pwd" placeholder="Password" name="pwd" required>

            </div>

            <p class="text-danger"><c:out value="${error}"/></p>

            <button type="submit" class="btn btn-primary">Submit</button>

            <p>Not Registered Yet <a href="signup">Click here</a></p>

        </form>

        </div>

    </div>

    <div class="col-6">

        <div class="contain p-3 my-3 bg-dark text-white">

        <h2 style="text-align:center">Admin Login</h2>

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

            <div class="form-group">

                <label for="admin\_id">User Name:</label>

                <input type="text" class="form-control" id="admin\_id" placeholder="User Name" name="admin\_id" required>

            </div>

            <div class="form-group">

                <label for="uname">Password:</label>

                <input type="password" class="form-control" id="admin\_pwd" placeholder="Password" name="admin\_pwd" required>

            </div>

            <p class="text-danger"><c:out value="${adminerrlogmessage}"/></p>

            <button type="submit" class="btn btn-primary">Submit</button>

        </form>

        </div>

    </div>

    </div>

    </div>

</body>

</html>

#### Login1.jsp:

<%@ page language="java" contentType="text/html; charset=UTF-8"

    pageEncoding="UTF-8"%>

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

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

<%@ page isELIgnored="false" %>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Sporty Shoes - Login</title>

<link rel="stylesheet" href="css/bootstrap.min.css">

</head>

<body>

<jsp:include page="/components/header1.jsp" ></jsp:include>

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

<jsp:include page="/components/topbar.jsp" ></jsp:include>

${error}

<form name=frmLogin action="loginaction" method="post">

 <table border=1 cellspacing=2 cellpadding=4>

    <tr>

        <td width=25%>Email id\*</td>

        <td><input name=email\_id maxlength=50></td>

    </tr>

    <tr>

        <td width=25%>Password\*</td>

        <td><input name=pwd maxlength=10 type="password"></td>

    </tr>

    <tr>

        <td colspan=2>

            <button>Login</button><br>

            <a href="signup">Not a member? Signup</a>

        </td>

    </tr>

 </table>

</form>

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

</body>

</html>

#### Purchases.jsp:

<%@ page language="java" contentType="text/html; charset=UTF-8"

    pageEncoding="UTF-8"%>

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

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

<%@ page isELIgnored="false" %>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Sporty Shoes - Your Purchases</title>

<link rel="stylesheet" href="css/bootstrap.min.css">

</head>

<body>

<jsp:include page="/components/header1.jsp" ></jsp:include>

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

<jsp:include page="/components/topbar.jsp" ></jsp:include>

<div class="container p-3 my-3 bg-dark text-white">

<br style="color:orange;text-align:right">Total ${list.size()} Orders: ${totalAmount }<br>

<table class="table table-bordered">

  <thead>

    <tr>

      <th scope="col">Order ID</th>

      <th scope="col">Date</th>

      <th scope="col">Total</th>

      <th scope="col">Items</th>

    </tr>

  </thead>

  <tbody>

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

    <tr>

      <th scope="row">${item.ID }</th>

      <td>${item.date }</td>

      <td>${item.total}</td>

      <td>${mapItems.get(item.ID)}</td>

    </tr>

  </c:forEach>

  </tbody>

</table>

</div>

<br>

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

</body>

</html>

#### Register-confirm.jsp:

<%@ page language="java" contentType="text/html; charset=UTF-8"

    pageEncoding="UTF-8"%>

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

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

<%@ page isELIgnored="false" %>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Sporty Shoes - Registration Confirmation</title>

<link rel="stylesheet" href="css/bootstrap.min.css">

</head>

<body>

<jsp:include page="/components/header1.jsp" ></jsp:include>

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

<jsp:include page="/components/topbar.jsp" ></jsp:include>

<div class="container">

<p class="alert alert-success" style="text-align:center;">Your registration is confirmed. <a href="login">Login to continue shopping</a><br></p>

</div>

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

</body>

</html>

#### Register.jsp:

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

    pageEncoding="ISO-8859-1"%>

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

<%@ page isELIgnored="false" %>

<!DOCTYPE html>

<html>

<head>

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

<title>Insert title here</title>

<link rel="stylesheet" href="css/bootstrap.min.css">

<style>

.detailsbox{

    padding:80px 25%;

}

</style>

</head>

<body>

<body>

<jsp:include page="/components/header1.jsp" ></jsp:include>

    <div class="detailsbox">

    <div class="margin p-3 my-3 bg-dark text-white">

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

          <h4>Register</h4>

          <div class="row">

          <div class="form-group col">

            <label for="fname">First name</label>

            <input type="text" class="form-control" id="fname" name="fname"   maxlength="20" placeholder="Enter first name" required>

          </div>

          <div class="form-group col">

            <label for="lname">Last name</label>

            <input type="text" class="form-control" id="lname" name="lname" maxlength="20" placeholder="Enter last name" required>

          </div>

          </div>

          <div class="form-group">

            <label for="email\_id">Email address</label>

            <input type="email" class="form-control" id="email\_id" name="email\_id" maxlength="40" placeholder="Enter email" required>

          </div>

          <div class="form-group">

            <label for="age">Age</label>

            <input type="number" class="form-control" id="age" name="age" placeholder="Enter Age" required>

          </div>

          <div class="form-group">

            <label for="address">Address</label>

            <input type="text" class="form-control" id="address" name="address" maxlength="100" placeholder="Enter Address" required>

          </div>

          <div class="form-group">

            <label for="pwd">Password</label>

            <input type="password" class="form-control" id="pwd" name="pwd" maxlength="20" placeholder="Enter password" required>

          </div>

          <div class="form-group">

            <label for="pwd2">Confirm Password</label>

            <input type="password" class="form-control" id="pwd2" name="pwd2" maxlength="20" placeholder="Enter password" required>

          </div>

          <div class="">

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

              <h5 class="alert alert-danger"><c:out value="${error}"/></h5>

              </c:if>

              <button type="submit" class="btn btn-primary"> REGISTER </button>

          </div>

        </form>

    </div>

    </div>

</body>

</html>

## Java\com\online:

### Controller:

#### Admincontroller.java:

package com.online.controller;

import java.math.BigDecimal;

import java.util.HashMap;

import java.util.List;

import javax.servlet.http.HttpSession;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.ui.ModelMap;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.bind.annotation.RequestParam;

import com.online.model.Admin;

import com.online.model.Category;

import com.online.model.Product;

import com.online.model.Purchase;

import com.online.model.PurchaseItem;

import com.online.model.User;

import com.online.service.AdminService;

import com.online.service.CategoryService;

import com.online.service.ProductService;

import com.online.service.PurchaseItemService;

import com.online.service.PurchaseService;

import com.online.service.UserService;

@Controller

public class AdminController {

    @Autowired

    private AdminService adminService;

    @Autowired

    private CategoryService categoryService;

    @Autowired

    private ProductService productService;

    @Autowired

    private PurchaseService purchaseService;

    @Autowired

    private PurchaseItemService purchaseItemService;

    @Autowired

    private UserService userService;

      @RequestMapping(value = "/adminlogin", method = RequestMethod.GET)

        public String login(ModelMap map, javax.servlet.http.HttpServletRequest request)

        {

          map.addAttribute("pageTitle", "ADMIN LOGIN");

            return "login";

        }

      @RequestMapping(value = "/adminloginaction", method = RequestMethod.POST)

        public String loginAction(ModelMap map, javax.servlet.http.HttpServletRequest request,

                 @RequestParam(value="admin\_id", required=true) String adminId,

                 @RequestParam(value="admin\_pwd", required=true) String adminPwd)

        {

          Admin admin = adminService.authenticate(adminId, adminPwd);

          if (admin == null) {

              map.addAttribute("adminerrlogmessage", "Admin login failed");

              return "login";

          }

          // store admin id in session

          HttpSession session = request.getSession();

          session.setAttribute("admin\_id", admin.getID());

            return "admin/dashboard";

        }

      @RequestMapping(value = "/admindashboard", method = RequestMethod.GET)

        public String dashboard(ModelMap map, javax.servlet.http.HttpServletRequest request)

        {

          // check if session is still alive

          HttpSession session = request.getSession();

          if (session.getAttribute("admin\_id") == null) {

              return "login";

          }

          map.addAttribute("pageTitle", "ADMIN DASHBOARD");

            return "admin/dashboard";

        }

      @RequestMapping(value = "/adminchangepassword", method = RequestMethod.GET)

        public String changePwd(ModelMap map, javax.servlet.http.HttpServletRequest request)

        {

          // check if session is still alive

          HttpSession session = request.getSession();

          if (session.getAttribute("admin\_id") == null) {

              return "login";

          }

          Admin admin = adminService.getAdminById((Long) session.getAttribute("admin\_id"));

          map.addAttribute("admin", admin);

          map.addAttribute("pageTitle", "ADMIN CHANGE PASSWORD");

            return "admin/change-password";

        }

      @RequestMapping(value = "/adminchangepwdaction", method = RequestMethod.POST)

        public String updatePassword(ModelMap map,  @RequestParam(value="pwd", required=true) String pwd,

                 @RequestParam(value="pwd2", required=true) String pwd2,

                 javax.servlet.http.HttpServletRequest request)

        {

          // check if session is still alive

          HttpSession session = request.getSession();

          if (session.getAttribute("admin\_id") == null) {

              return "login";

          }

          if (pwd == null || pwd2 == null || pwd.equals("") || pwd2.equals("")) {

              map.addAttribute("error", "Error , Incomplete passwords submitted.");

              return "admin/change-password";

          }

          if (!pwd.equals(pwd2)) {

              map.addAttribute("error", "Error , Passwords do not match.");

              return "admin/change-password";

          }

          Admin admin = adminService.getAdminById((Long) session.getAttribute("admin\_id"));

          admin.setAdminPwd(pwd);

          adminService.updatePwd(admin);

            return "admin/dashboard";

        }

      @RequestMapping(value = "/adminproducts", method = RequestMethod.GET)

        public String products(ModelMap map, javax.servlet.http.HttpServletRequest request)

        {

          // check if session is still alive

          HttpSession session = request.getSession();

          if (session.getAttribute("admin\_id") == null) {

              return "login";

          }

          List<Product> list = productService.listAll();

          // use a MAP to link category names to each product in list

          HashMap<Long, String> mapCats = new HashMap<Long, String>();

          for(Product product: list) {

              Category category = categoryService.getCategoryById(product.getCategoryId());

              if (category != null)

                  mapCats.put(product.getID(), category.getName());

          }

          map.addAttribute("list", list);

          map.addAttribute("mapCats", mapCats);

          map.addAttribute("pageTitle", "ADMIN SETUP PRODUCTS");

            return "admin/products";

        }

      @RequestMapping(value = "/admincategories", method = RequestMethod.GET)

        public String categories(ModelMap map, javax.servlet.http.HttpServletRequest request)

        {

          // check if session is still alive

          HttpSession session = request.getSession();

          if (session.getAttribute("admin\_id") == null) {

              return "login";

          }

          List<Category> list = categoryService.getAllCategories();

          map.addAttribute("list", list);

          map.addAttribute("pageTitle", "ADMIN SETUP PRODUCT CATEGORIES");

            return "admin/categories";

        }

      @RequestMapping(value = "/adminmembers", method = RequestMethod.GET)

        public String members(ModelMap map, javax.servlet.http.HttpServletRequest request)

        {

          // check if session is still alive

          HttpSession session = request.getSession();

          if (session.getAttribute("admin\_id") == null) {

              return "login";

          }

          List<User> list = userService.getAllUsers();

          map.addAttribute("list", list);

          map.addAttribute("pageTitle", "ADMIN BROWSE MEMBERS");

            return "admin/members";

        }

      @RequestMapping(value = "/adminpurchases", method = RequestMethod.GET)

        public String purchases(ModelMap map, javax.servlet.http.HttpServletRequest request)

        {

          // check if session is still alive

          HttpSession session = request.getSession();

          if (session.getAttribute("admin\_id") == null) {

              return "login";

          }

          List<Purchase> list = purchaseService.getAllItems();

          BigDecimal total = new BigDecimal(0.0);

          for(Purchase purchase: list) {

              total = total.add(purchase.getTotal());

          }

          // use MAPs to mape users to each purchase and item names to each purchase item row

          HashMap<Long, String> mapItems = new HashMap<Long, String>();

          HashMap<Long, String> mapUsers = new HashMap<Long, String>();

          for(Purchase purchase: list) {

              total = total.add(purchase.getTotal());

              User user = userService.getUserById(purchase.getUserId());

              if (user != null)

                  mapUsers.put(purchase.getID(), user.getFname() + " " + user.getLname());

              List<PurchaseItem> itemList = purchaseItemService.getAllItemsByPurchaseId(purchase.getID());

              StringBuilder sb = new StringBuilder("");

              for(PurchaseItem item: itemList) {

                  Product product = productService.getProductById(item.getProductId());

                  if (product != null)

                      sb.append(product.getName() + ", " +

                            item.getQty() + " units @" + item.getRate() + " = "

                            + item.getPrice() + "<br>");

              }

              mapItems.put(purchase.getID(), sb.toString());

          }

          map.addAttribute("totalAmount", total);

          map.addAttribute("list", list);

          map.addAttribute("mapItems", mapItems);

          map.addAttribute("mapUsers", mapUsers);

          map.addAttribute("pageTitle", "ADMIN PURCHASES REPORT");

            return "admin/purchases";

        }

      @RequestMapping(value = "/admindeletecat",  method = RequestMethod.GET)

        public String deleteCategory(ModelMap map,  @RequestParam(value="id", required=true) String id,

                javax.servlet.http.HttpServletRequest request)

        {

          // check if session is still alive

          HttpSession session = request.getSession();

          if (session.getAttribute("admin\_id") == null) {

              return "login";

          }

          long idValue = Long.parseLong(id);

          Category category = new Category();

          if (idValue > 0) {

              categoryService.deleteCategory(idValue);

          }

          return "redirect:admincategories";

        }

      @RequestMapping(value = "/admineditcat",  method = RequestMethod.GET)

        public String editCategory(ModelMap map,  @RequestParam(value="id", required=true) String id,

                javax.servlet.http.HttpServletRequest request)

        {

          // check if session is still alive

          HttpSession session = request.getSession();

          if (session.getAttribute("admin\_id") == null) {

              return "login";

          }

          long idValue = Long.parseLong(id);

          Category category = new Category();

          if (idValue > 0) {

              category = categoryService.getCategoryById(idValue);

          } else {

              category.setID(0);

          }

          map.addAttribute("category", category);

          map.addAttribute("pageTitle", "ADMIN EDIT PRODUCT CATEGORY");

            return "admin/edit-category";

        }

      @RequestMapping(value = "/admineditcataction", method = RequestMethod.POST)

        public String updateCategory(ModelMap map,  @RequestParam(value="id", required=true) String id,

                 @RequestParam(value="name", required=true) String name,

                 javax.servlet.http.HttpServletRequest request)

        {

          long idValue = Long.parseLong(id);

          if (name == null || name.equals("") ) {

              map.addAttribute("error", "Error, A category name must be specified");

              return "redirect:admineditcat?id="+id;

          }

            Category category = new Category();

            category.setID(idValue);

            category.setName(name);

            categoryService.updateCategory(category);

            return "redirect:admincategories";

        }

      @RequestMapping(value = "/admindeleteproduct",  method = RequestMethod.GET)

        public String deleteProduct(ModelMap map,  @RequestParam(value="id", required=true) String id,

                javax.servlet.http.HttpServletRequest request)

        {

          // check if session is still alive

          HttpSession session = request.getSession();

          if (session.getAttribute("admin\_id") == null) {

              return "login";

          }

          long idValue = Long.parseLong(id);

          Product product = new Product();

          if (idValue > 0) {

              productService.deleteProduct(idValue);

          }

          return "redirect:adminproducts";

        }

      @RequestMapping(value = "/admineditproduct",  method = RequestMethod.GET)

        public String editProduct(ModelMap map,  @RequestParam(value="id", required=true) String id,

                javax.servlet.http.HttpServletRequest request)

        {

          // check if session is still alive

          HttpSession session = request.getSession();

          if (session.getAttribute("admin\_id") == null) {

              return "login";

          }

          long idValue = Long.parseLong(id);

          Product product = new Product();

          if (idValue > 0) {

              product = productService.getProductById(idValue);

          } else {

              product.setID(0);

              product.setCategoryId(0);

          }

          String dropDown = categoryService.getCategoriesDropDown(product.getCategoryId());

          map.addAttribute("product", product);

          map.addAttribute("catDropdown", dropDown);

          map.addAttribute("pageTitle", "ADMIN EDIT PRODUCT");

            return "admin/edit-product";

        }

      @RequestMapping(value = "/admineditproductaction", method = RequestMethod.POST)

        public String updateProduct(ModelMap map, javax.servlet.http.HttpServletRequest request,

                 @RequestParam(value="id", required=true) String id,

                 @RequestParam(value="name", required=true) String name,

                 @RequestParam(value="price", required=true) String price,

                 @RequestParam(value="category", required=true) String categoryId)

        {

          long idValue = Long.parseLong(id);

          long categoryIdValue = Long.parseLong(categoryId);

          BigDecimal priceValue = new BigDecimal(0.0);

          if (name == null || name.equals("") ) {

              map.addAttribute("error", "Error, A product name must be specified");

              return "redirect:admineditproduct?id="+id;

          }

          try {

              priceValue = new BigDecimal(price);

          } catch (Exception ex) {

              map.addAttribute("error", "Error, Price is invalid");

              return "redirect:admineditproduct?id="+id;

          }

            Product product = new Product();

            product.setID(idValue);

            product.setCategoryId(Long.parseLong(categoryId));

            product.setName(name);

            product.setPrice(priceValue);

            productService.updateProduct(product);

            return "redirect:adminproducts";

        }

      @RequestMapping(value = "/adminlogout", method = RequestMethod.GET)

        public String logout(ModelMap map, javax.servlet.http.HttpServletRequest request)

        {

            HttpSession session = request.getSession();

            session.invalidate();

            return "login";

        }

}

#### CartController.java:

package com.online.controller;

import java.math.BigDecimal;

import java.util.ArrayList;

import java.util.Calendar;

import java.util.List;

import javax.servlet.http.HttpSession;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.ui.ModelMap;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.bind.annotation.RequestParam;

import com.online.model.CartItem;

import com.online.model.Product;

import com.online.model.Purchase;

import com.online.model.PurchaseItem;

import com.online.service.ProductService;

import com.online.service.PurchaseItemService;

import com.online.service.PurchaseService;

@Controller

public class CartController {

    @Autowired

    private ProductService productService;

    @Autowired

    private PurchaseService purchaseService;

    @Autowired

    private PurchaseItemService purchaseItemService;

      @SuppressWarnings("unchecked")

    @RequestMapping(value = "/cart", method = RequestMethod.GET)

        public String cart(ModelMap map, javax.servlet.http.HttpServletRequest request)

        {

          // check if user is logged in

          HttpSession session = request.getSession();

          if (session.getAttribute("user\_id") == null) {

              map.addAttribute("error", "Error, You need to login before adding items to cart");

          } else {

              // if cart is already in session then retrieve it else create a new cart list and

              // save it to session

              List<CartItem> cartItems = new ArrayList<CartItem>();

              if (session.getAttribute("cart\_items") != null)

                  cartItems = (List<CartItem>) session.getAttribute("cart\_items");

              // get total of all cart items

              BigDecimal totalValue = getCartValue(cartItems);

              map.addAttribute("cartValue", totalValue);

              map.addAttribute("cartItems", cartItems);

          }

          map.addAttribute("pageTitle", "SPORTY SHOES - YOUR CART");

            return "cart";

        }

      @SuppressWarnings("unchecked")

    @RequestMapping(value = "/cartadditem", method = RequestMethod.GET)

        public String cartAddItem(ModelMap model, javax.servlet.http.HttpServletRequest request,

                @RequestParam(value="id", required=true) String productId)

        {

          // check if user is logged in

          HttpSession session = request.getSession();

          if (session.getAttribute("user\_id") == null) {

              model.addAttribute("error", "Error, You neems to cart");

          } else {

              long idValue = Long.parseLong(productId);

              // if cart is already in session then retrieve it else create a new cart list and

              // save it to session

              List<CartItem> cartItems = new ArrayList<CartItem>();

              if (session.getAttribute("cart\_items") != null)

                  cartItems = (List<CartItem>) session.getAttribute("cart\_items");

              if (isItemInCart(cartItems, idValue)) {

                  System.out.println("error already exists");

                  model.addAttribute("error", "Error, You need =ing items to cart");

              } else {

                  Product product = productService.getProductById(idValue);

                  CartItem item = new CartItem();

                  item.setProductId(idValue);

                  item.setQty(1);

                  item.setRate(product.getPrice());

                  BigDecimal dprice = item.getRate().multiply(new BigDecimal(item.getQty()));

                  item.setPrice(dprice);

                  item.setName(product.getName());

                  cartItems.add(item);

                  session.setAttribute("cart\_items", cartItems);

              }

          }

            return "forward:cart";

        }

      @RequestMapping(value = "/cartdeleteitem", method = RequestMethod.GET)

        public String cartDeleteItem(ModelMap map, javax.servlet.http.HttpServletRequest request,

                @RequestParam(value="id", required=true) String id)

        {

          // check if user is logged in

          HttpSession session = request.getSession();

          if (session.getAttribute("user\_id") == null) {

              map.addAttribute("error", "Error, You need to login before deleting items from cart");

          } else {

              long idValue = Long.parseLong(id);

              List<CartItem> cartItems = new ArrayList<CartItem>();

              if (session.getAttribute("cart\_items") != null)

                  cartItems = (List<CartItem>) session.getAttribute("cart\_items");

              for(CartItem item: cartItems) {

                  if (item.getProductId() == idValue) {

                      cartItems.remove(item);

                      session.setAttribute("cart\_items", cartItems);

                      break;

                  }

               }

          }

            return "redirect:cart";

        }

      @RequestMapping(value = "/checkout", method = RequestMethod.GET)

        public String checkout(ModelMap map, javax.servlet.http.HttpServletRequest request)

        {

          // check if user is logged in

          HttpSession session = request.getSession();

          if (session.getAttribute("user\_id") == null) {

              map.addAttribute("error", "Error, You need to login before checking out");

          } else {

              List<CartItem> cartItems = new ArrayList<CartItem>();

              if (session.getAttribute("cart\_items") != null)

                  cartItems = (List<CartItem>) session.getAttribute("cart\_items");

              BigDecimal totalValue = getCartValue(cartItems);

              map.addAttribute("cartValue", totalValue);

              map.addAttribute("cartItems", cartItems);

          }

          map.addAttribute("pageTitle", "SPORTY SHOES - CHECKOUT");

            return "checkout";

        }

      @RequestMapping(value = "/completepurchase", method = RequestMethod.GET)

        public String completePurchase(ModelMap map, javax.servlet.http.HttpServletRequest request)

        {

          // check if user is logged in

          HttpSession session = request.getSession();

          if (session.getAttribute("user\_id") == null) {

              map.addAttribute("error", "Error, You need to login before completing purchase");

          } else {

              // take items from cart and update the databae

              List<CartItem> cartItems = new ArrayList<CartItem>();

              if (session.getAttribute("cart\_items") != null)

                  cartItems = (List<CartItem>) session.getAttribute("cart\_items");

              BigDecimal totalValue = getCartValue(cartItems);

              long userId = (Long) session.getAttribute("user\_id") ;

              Purchase purchase = new Purchase();

              purchase.setUserId(userId);

              purchase.setDate(Calendar.getInstance().getTime());

              purchase.setTotal(totalValue);

              long purchaseId = purchaseService.updatePurchase(purchase);

              for(CartItem item: cartItems) {

                  PurchaseItem pItem = new PurchaseItem();

                  pItem.setPurchaseId(purchaseId);

                  pItem.setProductId(item.getProductId());

                  pItem.setUserId(userId);

                  pItem.setRate(item.getRate());

                  pItem.setQty(item.getQty());

                  pItem.setPrice(item.getPrice());

                  purchaseItemService.updateItem(pItem);

              }

              map.addAttribute("cartValue", totalValue);

              map.addAttribute("cartItems", cartItems);

          }

            return "redirect:confirm";

        }

      @RequestMapping(value = "/gateway", method = RequestMethod.GET)

        public String gateway(ModelMap map, javax.servlet.http.HttpServletRequest request)

        {

          // check if user is logged in

          HttpSession session = request.getSession();

          if (session.getAttribute("user\_id") == null) {

              map.addAttribute("error", "Error, You need to login before making payment");

          } else {

              List<CartItem> cartItems = new ArrayList<CartItem>();

              if (session.getAttribute("cart\_items") != null)

                  cartItems = (List<CartItem>) session.getAttribute("cart\_items");

              BigDecimal totalValue = getCartValue(cartItems);

              map.addAttribute("cartValue", totalValue);

              map.addAttribute("cartItems", cartItems);

          }

          map.addAttribute("pageTitle", "SPORTY SHOES - PAYMENT GATEWAY");

            return "gateway";

        }

      @RequestMapping(value = "/confirm", method = RequestMethod.GET)

        public String confirm(ModelMap map, javax.servlet.http.HttpServletRequest request)

        {

          // check if user is logged in

          HttpSession session = request.getSession();

          if (session.getAttribute("user\_id") == null) {

              map.addAttribute("error", "Error, You need to login before completing the purchase");

          } else {

              // clear items from cart as order has completed

              List<CartItem> cartItems = new ArrayList<CartItem>();

              if (session.getAttribute("cart\_items") != null)

                  cartItems = (List<CartItem>) session.getAttribute("cart\_items");

              BigDecimal totalValue = getCartValue(cartItems);

              map.addAttribute("cartValue", totalValue);

              cartItems.clear();

              session.setAttribute("cart\_items", null);

          }

          map.addAttribute("pageTitle", "SPORTY SHOES - PURCHASE CONFIRMATION");

            return "confirm";

        }

      private boolean isItemInCart(List<CartItem> list, long item) {

          boolean retVal = false;

          for(CartItem thisItem: list) {

              if (item == thisItem.getProductId()) {

                  retVal = true;

                  break;

              }

          }

          return retVal;

      }

      private BigDecimal getCartValue(List<CartItem> list) {

          BigDecimal total = new BigDecimal(0.0);

          for(CartItem item: list) {

              BigDecimal dprice = item.getRate().multiply(new BigDecimal(item.getQty()));

              total= total.add(dprice);

           }

          return total;

      }

}

#### DashboardController.java:

package com.online.controller;

import org.springframework.stereotype.Controller;

import org.springframework.ui.ModelMap;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

@Controller

public class DashboardController {

      @RequestMapping(value = "/dashboard", method = RequestMethod.GET)

        public String dashboard(ModelMap map)

        {

          map.addAttribute("pageTitle", "SPORTY SHOES - DASHBOARD");

            return "dashboard";

        }

}

#### homeController.java:

package com.online.controller;

import java.util.HashMap;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestParam;

import com.online.model.\*;

import com.online.service.CategoryService;

import com.online.service.ProductService;

@Controller

public class HomeController {

    @Autowired

    private ProductService service;

    @Autowired

    private CategoryService categoryService;

    @RequestMapping(value = {"/", "/home"})

    public String viewHomePage(Model model) {

        List<Product> list = service.listAll();

         HashMap<Long, String> mapCats = new HashMap<Long, String>();

          for(Product product: list) {

              Category category = categoryService.getCategoryById(product.getCategoryId());

              if (category != null)

                  mapCats.put(product.getID(), category.getName());

          }

        List<Category> categorylist = categoryService.getAllCategories();

        model.addAttribute("categorylist", categorylist);

        model.addAttribute("mapCats", mapCats);

        model.addAttribute("list", list);

        model.addAttribute("pageTitle", "SPORTY SHOES - HOMEPAGE");

        return "index";

    }

    @RequestMapping(value = {"/bycategory"})

    public String viewCategory(Model model, @RequestParam(value="categoryname", required=true) String categoryname) {

        List<Product> list = service.listAll();

         HashMap<Long, String> mapCats = new HashMap<Long, String>();

          for(Product product: list) {

              Category category = categoryService.getCategoryById(product.getCategoryId());

              if (category != null)

                  mapCats.put(product.getID(), category.getName());

          }

          List<Category> categorylist = categoryService.getAllCategories();

        model.addAttribute("categorylist", categorylist);

        model.addAttribute("mapCats", mapCats);

        model.addAttribute("list", list);

        model.addAttribute("categoryname", categoryname);

        model.addAttribute("pageTitle", "SPORTY SHOES - HOMEPAGE");

        return "index";

    }

}

#### MemberController.java:

package com.online.controller;

import javax.servlet.http.HttpSession;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.ui.ModelMap;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.bind.annotation.RequestParam;

import com.online.model.\*;

import com.online.service.\*;

@Controller

public class MemberController {

    @Autowired

    private UserService userService;

      @RequestMapping(value = "/login", method = RequestMethod.GET)

        public String login(ModelMap map)

        {

          map.addAttribute("pageTitle", "SPORTY SHOES - MEMBER LOGIN");

            return "login";

        }

      @RequestMapping(value = "/loginaction", method = RequestMethod.POST)

        public String loginAction(ModelMap map,  HttpSession session,

                @RequestParam(value="email\_id", required=true) String emailId,

                 @RequestParam(value="pwd", required=true) String pwd)

        {

          User user = userService.authenticate(emailId, pwd);

          if (user == null) {

              map.addAttribute("error", "Login failed");

              return "login";

          }

          session.setAttribute("user\_id", user.getID());

          session.setAttribute("user\_name", user.getFname());

          return "redirect:dashboard";

        }

      @RequestMapping(value = "/signup", method = RequestMethod.GET)

        public String signup(ModelMap map)

        {

          map.addAttribute("pageTitle", "SPORTY SHOES - MEMBER REGISTRATION");

            return "register";

        }

      @RequestMapping(value = "/signupaction", method = RequestMethod.POST)

        public String signupAction(ModelMap map,  javax.servlet.http.HttpServletRequest request,

                @RequestParam(value="email\_id", required=true) String emailId,

                 @RequestParam(value="pwd", required=true) String pwd,

                 @RequestParam(value="pwd2", required=true) String pwd2,

                 @RequestParam(value="fname", required=true) String fname,

                 @RequestParam(value="lname", required=true) String lname,

                 @RequestParam(value="age", required=true) String age,

                 @RequestParam(value="address", required=true) String address)

        {

          if (emailId == null || emailId.equals("")) {

              map.addAttribute("error", "Email id is required.");

              return "register";

          }

          if (pwd == null || pwd2 == null || pwd.equals("") || pwd2.equals("")) {

              map.addAttribute("error", "Error , Incomplete passwords submitted.");

              return "register";

          }

          if (!pwd.equals(pwd2)) {

              map.addAttribute("error", "Error , Passwords do not match.");

              return "register";

          }

          if (fname == null || fname.equals("")) {

              map.addAttribute("error", "First name is required.");

              return "register";

          }

          if (lname == null || lname.equals("")) {

              map.addAttribute("error", "Last name is required.");

              return "register";

          }

          if (age == null || age.equals("")) {

              age = "0";

          }

          User user = userService.getUserByEmailId(emailId);

          if (user != null) {

              map.addAttribute("error", "This email id already exists.");

              return "register";

          }

          user = new User();

          user.setID(0);

          user.setEmail(emailId);

          user.setFname(fname);

          user.setLname(lname);

          user.setAge(Integer.parseInt(age));

          user.setAddress(address);

          user.setPwd(pwd);

          userService.updateUser(user);

            return "redirect:registerconfirm";

        }

      @RequestMapping(value = "/registerconfirm", method = RequestMethod.GET)

        public String registerConfirm(ModelMap map)

        {

            return "register-confirm";

        }

      @RequestMapping(value = "/editprofile", method = RequestMethod.GET)

        public String editProfile(ModelMap map, HttpSession session)

        {

          if (session.getAttribute("user\_id") == null) {

              return "login";

          }

          User user = userService.getUserById((Long) session.getAttribute("user\_id"));

          map.addAttribute("pageTitle", "SPORTY SHOES - MEMBER EDIT PROFILE");

          map.addAttribute("user", user);

            return "edit-profile";

        }

      @RequestMapping(value = "/editprofileaction", method = RequestMethod.POST)

        public String editProfileAction(ModelMap map,

                javax.servlet.http.HttpServletRequest request,

                @RequestParam(value="user\_id", required=true) String userId,

                 @RequestParam(value="pwd", required=true) String pwd,

                 @RequestParam(value="pwd2", required=true) String pwd2,

                 @RequestParam(value="fname", required=true) String fname,

                 @RequestParam(value="lname", required=true) String lname,

                 @RequestParam(value="age", required=true) String age,

                 @RequestParam(value="address", required=true) String address)

        {

          HttpSession session = request.getSession();

          if (session.getAttribute("user\_id") == null) {

              return "login";

          }

          User user = userService.getUserById((Long) session.getAttribute("user\_id"));

          map.addAttribute("user", user);

          if (pwd == null || pwd2 == null || pwd.equals("") || pwd2.equals("")) {

              map.addAttribute("error", "Error , Incomplete passwords submitted.");

              return "edit-profile";

          }

          if (!pwd.equals(pwd2)) {

              map.addAttribute("error", "Error , Passwords do not match.");

              return "edit-profile";

          }

          if (fname == null || fname.equals("")) {

              map.addAttribute("error", "First name is required.");

              return "edit-profile";

          }

          if (lname == null || lname.equals("")) {

              map.addAttribute("error", "Last name is required.");

              return "edit-profile";

          }

          if (age == null || age.equals("")) {

              age = "0";

          }

          user.setFname(fname);

          user.setLname(lname);

          user.setAge(Integer.parseInt(age));

          user.setAddress(address);

          user.setPwd(pwd);

          userService.updateUser(user);

            return "redirect:dashboard";

        }

      @RequestMapping(value = "/logout", method = RequestMethod.GET)

        public String logout(ModelMap map, HttpSession session)

        {

            session.invalidate();

            return "redirect:home";

        }

}

#### purchaseController.java:

package com.online.controller;

import java.math.BigDecimal;

import java.util.HashMap;

import java.util.List;

import javax.servlet.http.HttpSession;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.ui.ModelMap;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import com.online.model.Product;

import com.online.model.Purchase;

import com.online.model.PurchaseItem;

import com.online.service.ProductService;

import com.online.service.PurchaseItemService;

import com.online.service.PurchaseService;

@Controller

public class PurchaseController {

    @Autowired

    private ProductService productService;

    @Autowired

    private PurchaseService purchaseService;

    @Autowired

    private PurchaseItemService purchaseItemService;

      @RequestMapping(value = "/memberpurchases", method = RequestMethod.GET)

        public String memberpurchases(ModelMap map, HttpSession session)

        {

          if (session.getAttribute("user\_id") == null) {

              return "login";

          }

          long userId = (Long) session.getAttribute("user\_id");

          List<Purchase> list = purchaseService.getAllItemsByUserId(userId);

          BigDecimal total = new BigDecimal(0.0);

          // map purchase items to each purchase for display

          HashMap<Long, String> mapItems = new HashMap<Long, String>();

          for(Purchase purchase: list) {

              total = total.add(purchase.getTotal());

              List<PurchaseItem> itemList = purchaseItemService.getAllItemsByPurchaseId(purchase.getID());

              StringBuilder sb = new StringBuilder("");

              for(PurchaseItem item: itemList) {

                  Product product = productService.getProductById(item.getProductId());

                  if (product != null) {

                      sb.append(product.getName() + ", " +

                            item.getQty() + " units @" + item.getRate() + " = "

                            + item.getPrice() + "<br>");

                  }

              }

              mapItems.put(purchase.getID(), sb.toString());

          }

          map.addAttribute("totalAmount", total);

          map.addAttribute("list", list);

          map.addAttribute("mapItems", mapItems);

          map.addAttribute("pageTitle", "SPORTY SHOES - YOUR ORDERS");

           return "purchases";

        }

}

### Model:

#### Admin.java:

package com.online.model;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity

@Table(name = "admin")

public class Admin {

    @Id

    @GeneratedValue(strategy = GenerationType.IDENTITY)

    @Column(name = "ID")

    private long ID;

    @Column(name = "admin\_id")

    private String adminId;

    @Column(name = "admin\_pwd")

    private String pwd;

    public Admin(long iD, String adminId, String pwd) {

        super();

        ID = iD;

        this.adminId = adminId;

        this.pwd = pwd;

    }

    public Admin() {

        super();

    }

    public long getID() {

        return ID;

    }

    public void setID(long iD) {

        ID = iD;

    }

    public String getAdminId() {

        return adminId;

    }

    public void setAdminId(String adminId) {

        this.adminId = adminId;

    }

    public String getPwd() {

        return pwd;

    }

    public void setAdminPwd(String pwd) {

        this.pwd = pwd;

    }

}

#### cartItem.java:

package com.online.model;

import java.math.BigDecimal;

public class CartItem {

    private long productId;

    private String name;

    private BigDecimal rate;

    private BigDecimal price;

    private int qty;

    public long getProductId() {

        return this.productId;

    }

    public String getName() {

        return this.name;

    }

    public BigDecimal getPrice() {

        return this.price;

    }

    public BigDecimal getRate() {

        return this.rate;

    }

    public int getQty() {

        return this.qty;

    }

    public void setProductId(long id) {

        this.productId = id;

    }

    public void setName(String value) {

        this.name = value;

    }

    public void setPrice(BigDecimal value) {

        this.price = value;

    }

    public void setRate(BigDecimal value) {

        this.rate = value;

    }

    public void setQty(int value) {

        this.qty = value;

    }

}

#### Category.java:

package com.online.model;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity

@Table(name = "category")

public class Category {

    @Id

    @GeneratedValue(strategy = GenerationType.IDENTITY)

    @Column(name = "ID")

    private long ID;

    @Column(name = "name")

    private String name;

    public long getID() {

        return this.ID;

    }

    public String getName() {

        return this.name;

    }

    public void setID(long id) {

        this.ID = id;

    }

    public void setName(String value) {

        this.name = value;

    }

}

#### Product.java:

package com.online.model;

import java.math.BigDecimal;

import java.time.LocalDate;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity

@Table(name = "eproduct")

public class Product {

    @Id

    @GeneratedValue(strategy = GenerationType.IDENTITY)

    @Column(name = "ID")

    private long ID;

    @Column(name = "name")

    private String name;

    @Column(name = "price")

    private BigDecimal price;

    @Column(name = "date\_added")

    private LocalDate dateAdded;

    @Column(name = "category\_id")

    private long categoryId;

    public long getID() {

        return this.ID;

    }

    public String getName() {

        return this.name;

    }

    public BigDecimal getPrice() {

        return this.price;

    }

    public long getCategoryId() {

        return this.categoryId;

    }

    public LocalDate getDateAdded() {

        return this.dateAdded;

    }

    public void setID(long id) {

        this.ID = id;

    }

    public void setName(String value) {

        this.name = value;

    }

    public void setPrice(BigDecimal value) {

        this.price = value;

    }

    public void setCategoryId(long value) {

        this.categoryId = value;

    }

    public void setDateAdded(LocalDate date) {

        this.dateAdded = date;

    }

}

#### Purchase.java:

package com.online.model;

import java.math.BigDecimal;

import java.util.Date;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity

@Table(name = "purchases")

public class Purchase {

    @Id

    @GeneratedValue(strategy = GenerationType.IDENTITY)

    @Column(name = "ID")

    private long ID;

    @Column(name = "user\_id")

    private long userId;

    @Column(name = "date")

    private Date date;

    @Column(name = "gross\_total")

    private BigDecimal total;

    public long getID() {

        return this.ID;

    }

    public long getUserId() {

        return this.userId;

    }

    public BigDecimal getTotal() {

        return this.total;

    }

    public Date getDate() {

        return this.date;

    }

    public void setID(long id) {

        this.ID = id;

    }

    public void setUserId(long value) {

        this.userId = value;

    }

    public void setTotal(BigDecimal value) {

        this.total = value;

    }

    public void setDate(Date date) {

        this.date = date;

    }

}

#### purchaseItem.java:

package com.online.model;

import java.math.BigDecimal;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity

@Table(name = "purchase\_items")

public class PurchaseItem {

    @Id

    @GeneratedValue(strategy = GenerationType.IDENTITY)

    @Column(name = "ID")

    private long ID;

    @Column(name = "purchase\_id")

    private long purchaseId;

    @Column(name = "product\_id")

    private long productId;

    @Column(name = "user\_id")

    private long userId;

    @Column(name = "rate")

    private BigDecimal rate;

    @Column(name = "qty")

    private int qty;

    @Column(name = "price")

    private BigDecimal price;

    public long getID() {

        return this.ID;

    }

    public long getPurchaseId() {

        return this.purchaseId;

    }

    public long getProductId() {

        return this.productId;

    }

    public long getUserId() {

        return this.userId;

    }

    public BigDecimal getRate() {

        return this.rate;

    }

    public int getQty() {

        return this.qty;

    }

    public BigDecimal getPrice() {

        return this.price;

    }

    public void setID(long id) {

        this.ID = id;

    }

    public void setPurchaseId(long value) {

        this.purchaseId = value;

    }

    public void setProductId(long value) {

        this.productId = value;

    }

    public void setUserId(long value) {

        this.userId = value;

    }

    public void setRate(BigDecimal value) {

        this.rate = value;

    }

    public void setQty(int value) {

        this.qty = value;

    }

    public void setPrice(BigDecimal value) {

        this.price = value;

    }

}

#### User.java:

package com.online.model;

import java.util.Date;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity

@Table(name = "users")

public class User {

    @Id

    @GeneratedValue(strategy = GenerationType.IDENTITY)

    @Column(name = "ID")

    private long ID;

    @Column(name = "fname")

    private String fname;

    @Column(name = "lname")

    private String lname;

    @Column(name = "address")

    private String address;

    @Column(name = "age")

    private int age;

    @Column(name = "date\_added")

    private Date dateAdded;

    @Column(name = "emailid")

    private String emailId;

    @Column(name = "pwd")

    private String pwd;

    public long getID() {

        return this.ID;

    }

    public String getEmail() {

        return this.emailId;

    }

    public String getFname() {

        return this.fname;

    }

    public String getLname() {

        return this.lname;

    }

    public String getAddress() {

        return this.address;

    }

    public String getPwd() {

        return this.pwd;

    }

    public int getAge() {

        return this.age;

    }

    public Date getDateAdded() {

        return this.dateAdded;

    }

    public void setID(long id) {

        this.ID = id;

    }

    public void setEmail(String value) {

        this.emailId = value;

    }

    public void setFname(String value) {

        this.fname = value;

    }

    public void setLname(String value) {

        this.lname = value;

    }

    public void setAddress(String value) {

        this.address = value;

    }

    public void setPwd(String value) {

        this.pwd = value;

    }

    public void setAge(int value) {

        this.age = value;

    }

    public void setDateAdded(Date date) {

        this.dateAdded = date;

    }

}

### Repository:

#### adminRepository.java:

package com.online.repository;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.data.jpa.repository.Query;

import com.online.model.Admin;

public interface AdminRepository extends JpaRepository<Admin, Long> {

      @Query("select a from Admin a where a.adminId = ?1 and a.pwd = ?2")

      Admin authenticate(String admin\_id, String admin\_pwd);

      Admin findById(long id);

}

#### categoryRepository.java:

package com.online.repository;

import org.springframework.data.jpa.repository.JpaRepository;

import com.online.model.Category;

public interface CategoryRepository extends JpaRepository<Category, Long> {

      Category findById(long id);

}

#### productRepository.java:

package com.online.repository;

import org.springframework.data.jpa.repository.JpaRepository;

import com.online.model.Product;

public interface ProductRepository extends JpaRepository<Product, Long> {

    Product findById(long id);

}

#### purchaseItemRepsitory.java:

package com.online.repository;

import java.util.List;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.data.jpa.repository.Query;

import com.online.model.PurchaseItem;

public interface PurchaseItemRepository extends JpaRepository<PurchaseItem, Long> {

      PurchaseItem findById(long id);

      @Query("select pi from PurchaseItem pi where pi.purchaseId=?1")

      List<PurchaseItem> getAllItemsByPurchaseId(long purchase\_id);

      @Query("delete from PurchaseItem where purchaseId=?1")

      List<PurchaseItem> deleteAllItemsByPurchaseId(long purchase\_id);

}

#### purchaseRepository.java:

package com.online.repository;

import java.util.List;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.data.jpa.repository.Query;

import com.online.model.Purchase;

public interface PurchaseRepository extends JpaRepository<Purchase, Long> {

      Purchase findById(long id);

      @Query("select p from Purchase p where p.userId = ?1 order by ID desc")

      List<Purchase> getAllItemsByUserId(long userId);

}

#### userRepository.java:

package com.online.repository;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.data.jpa.repository.Query;

import com.online.model.User;

public interface UserRepository extends JpaRepository<User, Long> {

      User findByemailId(String emailId);

      @Query("select u from User u where u.emailId = ?1 and u.pwd = ?2")

      User authenticate(String emailId, String pwd);

      User findById(long id);

}

Service:  
adminService.java:

package com.online.service;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Component;

import org.springframework.transaction.annotation.Transactional;

import com.online.model.Admin;

import com.online.repository.AdminRepository;

@Component

public class AdminService {

     @Autowired

     private AdminRepository repo;

        @Transactional

        public Admin authenticate(String adminId, String pwd) {

            return repo.authenticate(adminId, pwd);

        }

        @Transactional

        public Admin getAdminById(long id) {

            return repo.findById(id);

        }

        @Transactional

        public void updatePwd(Admin admin) {

            repo.save(admin);

        }

}

categoryService.java:  
package com.online.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

import com.online.model.\*;

import com.online.repository.CategoryRepository;

@Service

@Transactional

public class CategoryService {

     @Autowired

     private CategoryRepository repo;

     public Category getCategoryById(long id) {

         return repo.findById(id);

     }

     public void updateCategory(Category category) {

         repo.save(category);

     }

     public void deleteCategory(long id) {

         repo.deleteById(id);

     }

     public List<Category> getAllCategories() {

         return repo.findAll();

     }

     public String getCategoriesDropDown(long id) {

         StringBuilder sb = new StringBuilder("");

         List<Category> list = repo.findAll();

         for(Category cat: list) {

             if (cat.getID() == id)

                 sb.append("<option value=" + String.valueOf(cat.getID()) + " selected>" + cat.getName() + "</option>");

             else

                 sb.append("<option value=" + String.valueOf(cat.getID()) + ">" + cat.getName() + "</option>");

         }

         return sb.toString();

     }

}

#### productService.java:

package com.online.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

import com.online.model.\*;

import com.online.repository.ProductRepository;

@Service

@Transactional

public class ProductService {

    @Autowired

    private ProductRepository repo;

    public List<Product> listAll() {

        return repo.findAll();

    }

    public void save(Product product) {

        repo.save(product);

    }

    public void updateProduct(Product product) {

        repo.save(product);

    }

    public Product getProductById(long id) {

        return repo.findById(id);

    }

    public void deleteProduct(long id) {

        repo.deleteById(id);

    }

}

#### purchaseItemService.java:

package com.online.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

import com.online.model.\*;

import com.online.repository.PurchaseItemRepository;

@Transactional

@Service

public class PurchaseItemService {

     @Autowired

     private PurchaseItemRepository repo;

        public PurchaseItem getItemById(long id) {

            return repo.findById(id);

        }

        public List<PurchaseItem> getAllItemsByPurchaseId(long purchaseId) {

            return repo.getAllItemsByPurchaseId(purchaseId);

        }

        public void updateItem(PurchaseItem item) {

            repo.save(item);

        }

        public void deleteItem(long id) {

            repo.deleteById(id);

        }

        public void deleteAllItemsForPurchaseId(long purchaseId) {

            repo.deleteAllItemsByPurchaseId(purchaseId);

        }

}

#### purchaseService.java:

package com.online.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

import com.online.model.\*;

import com.online.repository.PurchaseRepository;

@Service

@Transactional

public class PurchaseService {

     @Autowired

     private PurchaseRepository repo;

     public Purchase getPurchaseById(long id) {

         return repo.findById(id);

     }

     public List<Purchase> getAllItems() {

         return repo.findAll();

     }

     public List<Purchase> getAllItemsByUserId(long userId) {

         return repo.getAllItemsByUserId(userId);

     }

     public long updatePurchase(Purchase purchase) {

         repo.save(purchase);

         return purchase.getID();

     }

     public void deletePurchase(long id) {

         repo.deleteById(id);

     }

}

#### UserService.java:

package com.online.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

import com.online.model.\*;

import com.online.repository.UserRepository;

@Service

@Transactional

public class UserService {

     @Autowired

     private UserRepository repo;

        public User authenticate(String userId, String pwd) {

            return repo.authenticate(userId, pwd);

        }

        public User getUserById(long id) {

            return repo.findById(id);

        }

        public User getUserByEmailId(String emailId) {

            return repo.findByemailId(emailId);

        }

        public void updateUser(User user) {

            repo.save(user);

        }

        public List<User> getAllUsers() {

         return repo.findAll();

        }

}

#### sportyShoesApplication.java:

package com.online;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SportyShoesApplication {

    public static void main(String[] args) {

        SpringApplication.run(SportyShoesApplication.class, args);

    }

}

### Resources:

#### Application.properties:

spring.jpa.hibernate.ddl-auto=update

spring.datasource.url=jdbc:mysql://localhost:3306/sportyshoesdb

spring.datasource.username=root

spring.datasource.password=Abc@123

spring.jpa.database-platform=org.hibernate.dialect.MySQL8Dialect

server.port=8484

spring.jpa.open-in-view=false

spring.mvc.view.suffix=.jsp

logging.level.root=info

### maven dependency:

#### pom.xml

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

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

    <modelVersion>4.0.0</modelVersion>

    <parent>

        <groupId>org.springframework.boot</groupId>

        <artifactId>spring-boot-starter-parent</artifactId>

        <version>2.6.5</version>

        <relativePath/> <!-- lookup parent from repository -->

    </parent>

    <groupId>com.online</groupId>

    <artifactId>SportyShoes</artifactId>

    <version>0.0.1-SNAPSHOT</version>

    <name>SportyShoes</name>

    <description>Demo project for Spring Boot</description>

    <properties>

        <java.version>17</java.version>

    </properties>

    <dependencies>

        <dependency>

            <groupId>org.springframework.boot</groupId>

            <artifactId>spring-boot-starter</artifactId>

        </dependency>

        <dependency>

            <groupId>org.springframework.boot</groupId>

            <artifactId>spring-boot-starter-test</artifactId>

            <scope>test</scope>

        </dependency>

        <dependency>

        <groupId>org.springframework.boot</groupId>

        <artifactId>spring-boot-starter-data-jpa</artifactId>

    </dependency>

        <dependency>

    <groupId>javax.servlet</groupId>

    <artifactId>javax.servlet-api</artifactId>

    <scope>provided</scope>

</dependency>

<dependency>

        <groupId>org.springframework</groupId>

        <artifactId>spring-web</artifactId>

    </dependency>

<dependency>

            <groupId>org.springframework.boot</groupId>

            <artifactId>spring-boot-devtools</artifactId>

            <scope>runtime</scope>

            <optional>true</optional>

        </dependency>

        <dependency>

            <groupId>mysql</groupId>

            <artifactId>mysql-connector-java</artifactId>

            <scope>runtime</scope>

        </dependency>

        <dependency>

            <groupId>org.apache.tomcat</groupId>

            <artifactId>tomcat-jasper</artifactId>

            <version>9.0.41</version>

        </dependency>

        <dependency>

            <groupId>javax.servlet</groupId>

            <artifactId>jstl</artifactId>

        </dependency>

    <dependency>

        <groupId>org.springframework</groupId>

        <artifactId>spring-beans</artifactId>

    </dependency>

    <dependency>

        <groupId>org.springframework</groupId>

        <artifactId>spring-context</artifactId>

    </dependency>

    <dependency>

    <groupId>org.springframework.boot</groupId>

    <artifactId>spring-boot-starter-webflux</artifactId>

</dependency>

<dependency>

    <groupId>org.springframework.boot</groupId>

    <artifactId>spring-boot-starter-web</artifactId>

</dependency>

</dependencies>

    <build>

        <plugins>

            <plugin>

                <groupId>org.springframework.boot</groupId>

                <artifactId>spring-boot-maven-plugin</artifactId>

            </plugin>

        </plugins>

    </build>

</project>