

Lab 11

Chức năng xem chi tiết sản phẩm, thêm sản phẩm và review sản phẩm trước khi thêm

Mục tiêu

- Sử dụng JSP Implicit Objects: request, response, session.
- Xây dựng chức năng quản lý sản phẩm:
 - o Xem chi tiết 1 sản phẩm (customer và admin)
 - o Sửa thông tin chi tiết sản phẩm (admin)
 - o Thêm sản phẩm mới (admin)
 - o Xoá sản phẩm (admin)
 - o Review sản phẩm trước khi thêm và xoá (admin)

Phần I Bài tập step by step

Bài 1.1

Tạo module view product trong eMarket project.

Step 1: Tạo trang product.jsp

```
<%@page import="entity.ProductDetail"%>

<%@page import="entity.Product"%>

<%

    session.setAttribute("view", "/product ");

    Product selectedProduct = (Product)
session.getAttribute("selectedProduct");

    ProductDetail selectedProductDetail = (ProductDetail)
session.getAttribute("selectedProductDetail");

%>
```

```

<div id="container">

    <div class="one">

        <div class="heading_bg">

            <h2>

                <%=selectedProduct.getName()%>

            </h2>

        </div>

    </div>

    <div class="one-half">

        <div id="amazingslider-wrapper-1"
style="display:block;position:relative;max-width:450px;margin:0px auto
98px;">

            <div id="amazingslider-1"
style="display:block;position:relative;margin:0 auto;">

                <ul class="amazingslider-slides" style="display:none;">

                    <%

                        for (String img :
selectedProductDetail.getAllImages()) {

                            %>

                            <li></li>

                            <%

                                }

                            %>

                        </ul>

                        <ul class="amazingslider-thumbnails" style="display:none;">

                            <%

```

```

        for (String img :
selectedProductDetail.getAllImages()) {

        %>

        <li></li>

        <%

        }

        %>

    </ul>

</div>

</div>

</div>

<div class="one-half last">

    <ul id="tabify_menu" class="menu_tab" style="margin: 0;">

        <li><a href="#fane1">Product Details</a></li>

    </ul>

    <div id="fane1" class="tab_content">

        <h3>Technical Details</h3>

        <p>

            <%=selectedProductDetail.getInformation()%>

        </p>

        <h3>Accessories</h3>

        <p><%=selectedProductDetail.getAccessories()%></p>

        <h3>Warranty Strategy</h3>

        <p><%=selectedProductDetail.getGuaranty()%></p>

```

```

        <h3>Price</h3>

        <p><%=selectedProduct.getPrice()%> $</p>

        <p style="text-align:left; margin-right: 16px">

            <a href="#" class="button">Add to cart</a>

        </p>

    </div>

</div>

<div style="clear:both; height: 40px"></div>

</div>

```

Step 2: Trong entity class ProductDetail tạo function getAllImages() để view slider.

Class ProductDetail.java

```

package entity;

import java.io.Serializable;
import java.util.ArrayList;
import java.util.List;
import javax.persistence.Basic;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Lob;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.Table;
import javax.validation.constraints.Size;
import javax.xml.bind.annotation.XmlRootElement;

/**
 *
 * @author ThanDieu
 */
@Entity
@Table(name = "product_detail", catalog = "eMarket", schema = "")
@XmlRootElement
@NamedQueries({
    @NamedQuery(name = "ProductDetail.findAll", query = "SELECT p FROM ProductDetail p"),
    @NamedQuery(name = "ProductDetail.findByProductId", query = "SELECT p FROM ProductDetail p WHERE p.productId = :productId"),

```

```

    @NamedQuery(name = "ProductDetail.findByImage1", query = "SELECT p FROM
ProductDetail p WHERE p.image1 = :image1"),
    @NamedQuery(name = "ProductDetail.findByImage2", query = "SELECT p FROM
ProductDetail p WHERE p.image2 = :image2"),
    @NamedQuery(name = "ProductDetail.findByImage3", query = "SELECT p FROM
ProductDetail p WHERE p.image3 = :image3"),
    @NamedQuery(name = "ProductDetail.findByImage4", query = "SELECT p FROM
ProductDetail p WHERE p.image4 = :image4"),
    @NamedQuery(name = "ProductDetail.findByImage5", query = "SELECT p FROM
ProductDetail p WHERE p.image5 = :image5"))}
public class ProductDetail implements Serializable {
    private static final long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Basic(optional = false)
    @Column(name = "product_id")
    private Integer productId;
    @Lob
    @Size(max = 2147483647)
    @Column(name = "information")
    private String information;
    @Size(max = 45)
    @Column(name = "image1")
    private String image1;
    @Size(max = 45)
    @Column(name = "image2")
    private String image2;
    @Size(max = 45)
    @Column(name = "image3")
    private String image3;
    @Size(max = 45)
    @Column(name = "image4")
    private String image4;
    @Size(max = 45)
    @Column(name = "image5")
    private String image5;
    @Lob
    @Size(max = 2147483647)
    @Column(name = "accessories")
    private String accessories;
    @Lob
    @Size(max = 2147483647)
    @Column(name = "guaranty")
    private String guaranty;

    public ProductDetail() {
    }

    public ProductDetail(Integer productId) {
        this.productId = productId;
    }

    public Integer getProductId() {
        return productId;
    }
}

```

```

    }

    public void setProductId(Integer productId) {
        this.productId = productId;
    }

    public String getInformation() {
        return information;
    }

    public void setInformation(String information) {
        this.information = information;
    }

    public String getImage1() {
        return image1;
    }

    public void setImage1(String image1) {
        this.image1 = image1;
    }

    public String getImage2() {
        return image2;
    }

    public void setImage2(String image2) {
        this.image2 = image2;
    }

    public String getImage3() {
        return image3;
    }

    public void setImage3(String image3) {
        this.image3 = image3;
    }

    public String getImage4() {
        return image4;
    }

    public void setImage4(String image4) {
        this.image4 = image4;
    }

    public String getImage5() {
        return image5;
    }

    public void setImage5(String image5) {
        this.image5 = image5;
    }

    public String getAccessories() {

```

```

        return accessories;
    }

    public void setAccessories(String accessories) {
        this.accessories = accessories;
    }

    public String getGuaranty() {
        return guaranty;
    }

    public void setGuaranty(String guaranty) {
        this.guaranty = guaranty;
    }

    public List<String> getAllImages(){
        List<String> images = new ArrayList<String>();
        if (image1 != null) images.add(image1);
        if (image2 != null) images.add(image2);
        if (image3 != null) images.add(image3);
        if (image4 != null) images.add(image4);
        if (image5 != null) images.add(image5);
        return images;
    }

    @Override
    public int hashCode() {
        int hash = 0;
        hash += (productId != null ? productId.hashCode() : 0);
        return hash;
    }

    @Override
    public boolean equals(Object object) {
        // TODO: Warning - this method won't work in the case the id fields
are not set
        if (!(object instanceof ProductDetail)) {
            return false;
        }
        ProductDetail other = (ProductDetail) object;
        if ((this.productId == null && other.productId != null) ||
(this.productId != null && !this.productId.equals(other.productId))) {
            return false;
        }
        return true;
    }

    @Override
    public String toString() {
        return "entity.ProductDetail[ productId=" + productId + " ]";
    }
}

```

Step 3: Thêm xử lý cho product view trong ControllerServlet

Thêm đoạn code sau vào sau lệnh if xử lý user path:

```
else if (userPath.equals("/product")){
    Product selectedProduct;
    ProductDetail selectedProductDetail;
    String productId = request.getQueryString();
    if (productId != null){
        selectedProduct =
productSB.find(Integer.parseInt(productId));
        selectedProductDetail =
productDetailSB.find(Integer.parseInt(productId));
        session.setAttribute("selectedProduct", selectedProduct);
        session.setAttribute("selectedProductDetail",
selectedProductDetail);
    }
}
```

Thêm urlPattern cho Servlet:

```
@WebServlet(name = "ControllerServlet",
    urlPatterns = {"/ControllerServlet",
        "/category",
        "/product"})
```

Inject code cho ProductSessionBean và ProductDetailSessionBean để xử lý các thao tác với CSDL liên quan đến product.

```
@EJB
private ProductSessionBean productSB;

@EJB
private ProductDetailSessionBean productDetailSB;
```

Step 4: Tạo các navigation đến /product

Mở trang index.jsp. Sửa lại như sau:

```
<script src="js/jquery.tools.min.js"></script>

<script>

    $(function () {

        $("#prod_nav ul").tabs("#panes > div", {

            effect: 'fade',

            fadeOutSpeed: 400
```



```

        });

    });

</script>

<script>

    $(document).ready(function () {

        $(".pane-list li").click(function () {

            window.location = $(this).find("a").attr("href");

            return false;

        });

    });

</script>

<c:set var='view' value='/index' scope='session' />

<div id="container">

    <h2>Best Seller</h2>

    <div id="prod_wrapper">

        <div id="panes">

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

                <div>

                    <h5>${product.getName()}</h5>

                    <p>${product.getDescriptionDetail()}</p>

                    <p style="text-align:right; margin-right: 16px">

```

```

        <a href="<c:url
value='product?${product.productId}' />" class="button">More Info</a>

        <a href="#" class="button">Buy Now</a>

    </p>

</div>

</c:forEach>

</div>

<!-- END tab panes -->

<br clear="all">

<!-- navigator -->

<div id="prod_nav">

    <ul>

        <c:set var="countProduct" scope="request" value="${1}" />

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

            <li>

                <a href="#${countProduct}">

                    <strong>${product.getName()}</strong> $
${product.getPrice()}

                </a>

            </li>

            <c:set var="countProduct" scope="request"
value="${countProduct+1}" />

        </c:forEach>

    </ul>

```

```

    </div>

    <!-- close navigator -->

</div>

<div style="clear:both"></div>

<c:forEach var="category" items="${categories}">

    <c:if test="${category.categoryId == 4}" >

        <div class="one-fourth last">

            <div class="heading_bg">

                <h2>${category.name}</h2>

            </div>

            <a href="<c:url value='category?${category.categoryId}' />">

            </a>

        </div>

    </c:if>

    <c:if test="${category.categoryId < 4}">

        <div class="one-fourth">

            <div class="heading_bg">

                <h2>${category.name}</h2>

            </div>

            <a href="<c:url value='category?${category.categoryId}' />">

```

```

    </a>

</div>

</c:if>

</c:forEach>

<div style="clear:both; height: 40px"></div>

</div>

```

Thêm config cho trang product.jsp trong file web.xml

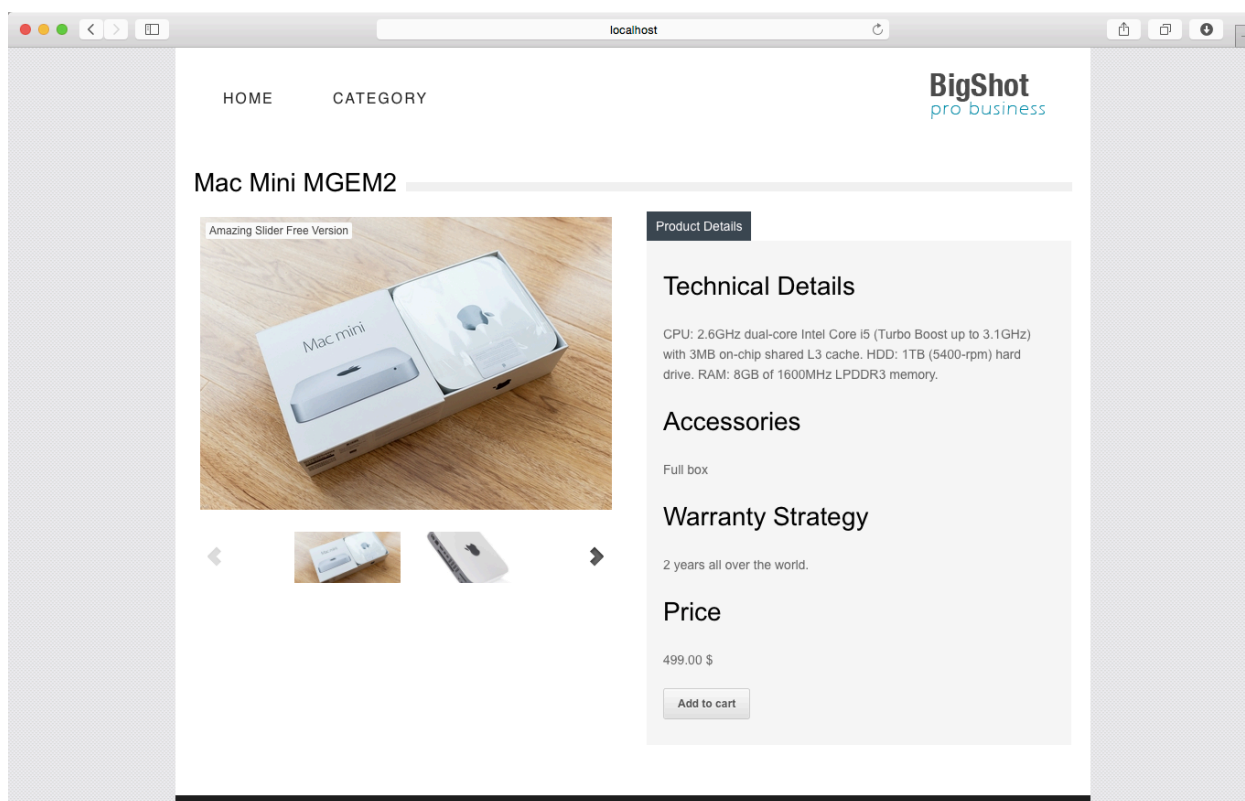
```

<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd">
    <listener>
        <description>ServletContextListener</description>
        <listener-class>listener.ControllerServletListener</listener-class>
    </listener>
    <session-config>
        <session-timeout>
            30
        </session-timeout>
    </session-config>
    <jsp-config>
        <jsp-property-group>
            <description>JSP configuration for the store front</description>
            <url-pattern>/index.jsp</url-pattern>
            <url-pattern>/contact.jsp</url-pattern>
            <url-pattern>/category.jsp</url-pattern>
            <url-pattern>/product.jsp</url-pattern>
            <include-prelude>/WEB-INF/jspf/header.jspf</include-prelude>
            <include-coda>/WEB-INF/jspf/footer.jspf</include-coda>
        </jsp-property-group>
        <jsp-property-group>
            <description>JSP configuration for the admin
console</description>
            <url-pattern>/admin/*</url-pattern>
            <include-prelude>/WEB-INF/jspf/header.jspf</include-prelude>
            <include-coda>/WEB-INF/jspf/footer.jspf</include-coda>
        </jsp-property-group>
    </jsp-config>
    <context-param>
        <param-name>imgProductPath</param-name>

```

```
<param-value>img/demo/</param-value>
</context-param>
<context-param>
  <param-name>imgProductDetailPath</param-name>
  <param-value>images/</param-value>
</context-param>
</web-app>
```

Chạy chương trình review kết quả:



PHẦN 2: BÀI TẬP LÀM THÊM

Tạo module addProduct và view lại product trước khi thực hiện add vào CSDL.

HẾT

