

Lab 12

Chức năng quản lý giỏ hàng

Mục tiêu

- Xây dựng chức năng quản lý giỏ hàng
 - o Thêm sản phẩm vào giỏ hàng (trực tiếp từ trang sản phẩm)
 - o Thay đổi số lượng sản phẩm trong giỏ hàng
 - o Xoá sản phẩm khỏi giỏ hàng

Phần I Bài tập step by step

Bài 1.1

Trong bài này chúng ta sẽ thực hiện module quản lý giỏ hàng.

STEP 1: Tạo view (các trang jsp cho giỏ hàng)

Thêm thông tin giỏ hàng cho tất cả các trang view bằng cách thêm vào trang header.jspf.

Mở trang header.jspf, ngay sau thẻ <body>, trong cặp thẻ <div id="container"> thêm đoạn thông tin sau:

```


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

    <c:choose>

        <c:when test="${cart.numberOfItems == null}">

            0

        </c:when>

        <c:otherwise>

            ${cart.numberOfItems}
```

```

        </c:otherwise>

    </c:choose>

    <c:choose>

        <c:when test="${cart.numberOfItems == 1}">

            item

        </c:when>

        <c:otherwise>

            items

        </c:otherwise>

    </c:choose>

</a>

```

Chúng ta cần thêm các key cần thiết vào resource-bundle (item và items).

Khi view các product, chúng ta cho phép user add product vào giỏ hàng. Trong trang index.jsp, tìm button có tên 'Buy Now', sửa lại cả dòng code này lại như sau:

```

<a href="<c:url value='addToCart?${product.productId}' />" class="buttonBuy
Now">

```

Dòng code này thêm đường dẫn tới xử lý addToCart (sẽ thực hiện trong controller): thực hiện thêm product vào giỏ hàng.

Trên trang product.jsp, tìm dòng code có key = 'Add to cart', thay thế bằng dòng lệnh sau:

```

<a href="<c:url value='addToCart?${selectedProduct.getProductId()}' />"
class="button">Add to cart</a>

```

Tạo trang viewCart.jsp để xem thông tin giỏ hàng hiện tại.

```

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

<div id="container">

    <div class="one">

```

```

<div class="heading_bg">

    <h2>Information of your cart</h2>

</div>

<div class="two-third">

    <c:if test="${!empty cart && cart.numberOfItems != 0}">

        <table border="0">

            <th>Product</th>

            <th>Price</th>

            <th>Quantity</th>

            <c:forEach var="cartItem" items="${cart.items}"
varStatus="iter">

                <c:set var="product"
value="${cartItem.product}"/>

                <tr>

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

                    <td class="">

                        <fmt:formatNumber type="currency"
currencySymbol="&dollar; " value="${cartItem.total}"/>

                        <br>

                        <span class="smallText">(

                            <fmt:formatNumber type="currency"
currencySymbol="&dollar; " value="${product.price}"/>

                            / unit )</span>

                        </td>

                    <td>

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

```

```

        <input type="hidden"
            name="productId"
            value="${product.productId}" />
        <input type="text"
            maxlength="2"
            size="2"
            value="${cartItem.quantity}"
            name="quantity"
            style="margin:5px" />
        <input type="submit"
            name="submit"
            value="update" />
    </form>
</td>
</tr>
</c:forEach>
<tr>
    <td></td>
    <td>Total:
        <fmt:formatNumber type="currency"
currencySymbol="&dollar; " value="${cart.subtotal}"/>
    </td>
    <td></td>
</tr>
</table>
</c:if>

```

```

</div>

<div class="sidebar_right">

    <c:set var="value">

        <c:choose>

            <!-- if 'selectedCategory' session object exists, send
user to previously viewed category --%>

            <c:when test="${!empty selectedCategory}">

                category

            </c:when>

            <!-- otherwise send user to welcome page --%>

            <c:otherwise>

                index.jsp

            </c:otherwise>

        </c:choose>

    </c:set>

    <ul class="sidebar_menu" style="margin:0">

        <li><a href="#"><strong>

            <c:choose>

                <c:when test="${cart.numberOfItems > 1}">

                    Your cart contains ${cart.numberOfItems}
items

                </c:when>

                <c:when test="${cart.numberOfItems == 1}">

                    Your cart contains ${cart.numberOfItems}
item

                </c:when>

```

```

        <c:otherwise>

            Your cart is empty

        </c:otherwise>

    </c:choose></strong>

</a>

</li>

<c:if test="${!empty cart && cart.numberOfItems != 0}">

    <c:url var="url" value="viewCart">

        <c:param name="clear" value="true"/>

    </c:url>

    <li><a href="${url}">&#x279f; Clear your cart</a></li>

</c:if>

    <c:url var="url" value="${value}"/>

    <li><a href="${url}">&#x279f; Continue shopping</a></li>

    <c:if test="${!empty cart && cart.numberOfItems != 0}">

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

    </c:if>

</ul>

</div>

</div>

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

</div>

```

Thêm trang viewCart.jsp vào config của web.xml để config jsp fragment (header và footer).

```

<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>
<url-pattern>/viewCart.jsp</url-pattern>
<url-pattern>/checkout.jsp</url-pattern>
<url-pattern>/confirmation.jsp</url-pattern>
<include-pragma>/WEB-INF/jspf/header.jspf</include-pragma>
<include-coda>/WEB-INF/jspf/footer.jspf</include-coda>
</jsp-property-group>

```

Các xử lý trang view cho đơn hàng đã hoàn thành.

STEP 2: Tạo Model chứa thông tin giỏ hàng của customer.

Với mỗi sản phẩm trong giỏ hàng, chúng ta cần thông tin về product và số lượng.

Tạo package cart -> Tạo class ShoppingCartItem.java

```

package cart;

import entity.Product;

/**
 * @author ThanDieu
 */
public class ShoppingCartItem {

    Product product;
    short quantity;

    public ShoppingCartItem(Product product) {
        this.product = product;
        quantity = 1;
    }

    public Product getProduct() {
        return product;
    }

    public short getQuantity() {
        return quantity;
    }

    public void setQuantity(short quantity) {
        this.quantity = quantity;
    }

    public void incrementQuantity() {
        quantity++;
    }
}

```

```

    public void decrementQuantity() {
        quantity--;
    }

    public double getTotal() {
        double amount = 0;
        amount = (this.getQuantity() * product.getPrice().doubleValue());
        return amount;
    }
}

```

STEP 3: Tạo controller, add product vào giỏ hàng

Trong package cart -> Tạo class ShoppingCart.java

Khi thêm thông tin vào giỏ hàng, chúng ta cần thực hiện synchronized giỏ hàng hiện tại. Chúng ta cũng cần các function tính giá cả tổng thể của từng mặt hàng theo số lượng và toàn bộ giỏ hàng.

Class ShoppingCart.java

```

package cart;

import entity.Product;
import java.util.*;

/**
 *
 * @author ThanDieu
 */
public class ShoppingCart {
    List<ShoppingCartItem> items;
    int numberOfItems;
    double total;

    public ShoppingCart() {
        items = new ArrayList<ShoppingCartItem>();
        numberOfItems = 0;
        total = 0;
    }

    public synchronized void addItem(Product product) {

        boolean newItem = true;

        for (ShoppingCartItem scItem : items) {

            if (scItem.getProduct().getProductId()== product.getProductId())
        }
    }
}

```



```

        newItem = false;
        scItem.incrementQuantity();
    }
}

if (newItem) {
    ShoppingCartItem scItem = new ShoppingCartItem(product);
    items.add(scItem);
}
}

public synchronized void update(Product product, String quantity) {

    short qty = -1;

    // cast quantity as short
    qty = Short.parseShort(quantity);

    if (qty >= 0) {

        ShoppingCartItem item = null;

        for (ShoppingCartItem scItem : items) {

            if (scItem.getProduct().getProductId() ==
product.getProductId()) {

                if (qty != 0) {
                    // set item quantity to new value
                    scItem.setQuantity(qty);
                } else {
                    // if quantity equals 0, save item and break
                    item = scItem;
                    break;
                }
            }
        }

        if (item != null) {
            // remove from cart
            items.remove(item);
        }
    }
}

public synchronized List<ShoppingCartItem> getItems() {

    return items;
}

public synchronized int getNumberOfItems() {

    numberOfItems = 0;
}

```

```

        for (ShoppingCartItem scItem : items) {
            numberOfItems += scItem.getQuantity();
        }

        return numberOfItems;
    }

    public synchronized double getSubtotal() {
        double amount = 0;

        for (ShoppingCartItem scItem : items) {
            Product product = (Product) scItem.getProduct();
            amount += (scItem.getQuantity() *
product.getPrice().doubleValue());
        }

        return amount;
    }

    public synchronized void calculateTotal(String surcharge) {
        double amount = 0;

        // cast surcharge as double
        double s = Double.parseDouble(surcharge);

        amount = this.getSubtotal();
        amount += s;

        total = amount;
    }

    public synchronized double getTotal() {
        return total;
    }

    public synchronized void clear() {
        items.clear();
        numberOfItems = 0;
        total = 0;
    }
}

```

STEP 4: Thêm xử lý addToCart trong main servlet của ứng dụng

Chúng ta cần thêm 2 xử lý trong ControllerServlet.java, addToCart và viewCart.

Thêm url patterns cho hai xử lý này:

```
WebServlet(name = "ControllerServlet",
    urlPatterns = {"/category",
        "/product",
        "/addToCart",
        "/viewCart",
        "/updateCart",
        "/checkout",
        "/purchase",
        "/chooseLanguage"})
```

Trong function doGet của ControllerServlet.java, chúng ta thêm hai đoạn else if sau:

```
else if (userPath.equals("/viewCart")) {
    String clear = request.getParameter("clear");

    if ((clear != null) && clear.equals("true")) {

        ShoppingCart cart = (ShoppingCart)
session.getAttribute("cart");
        cart.clear();
    }
}
else if (userPath.equals("/addToCart")) {

    // if user is adding item to cart for first time
    // create cart object and attach it to user session
    ShoppingCart cart = (ShoppingCart) session.getAttribute("cart");

    if (cart == null) {
        cart = new ShoppingCart();
        session.setAttribute("cart", cart);
    }

    // get user input from request
    String productId = request.getQueryString();

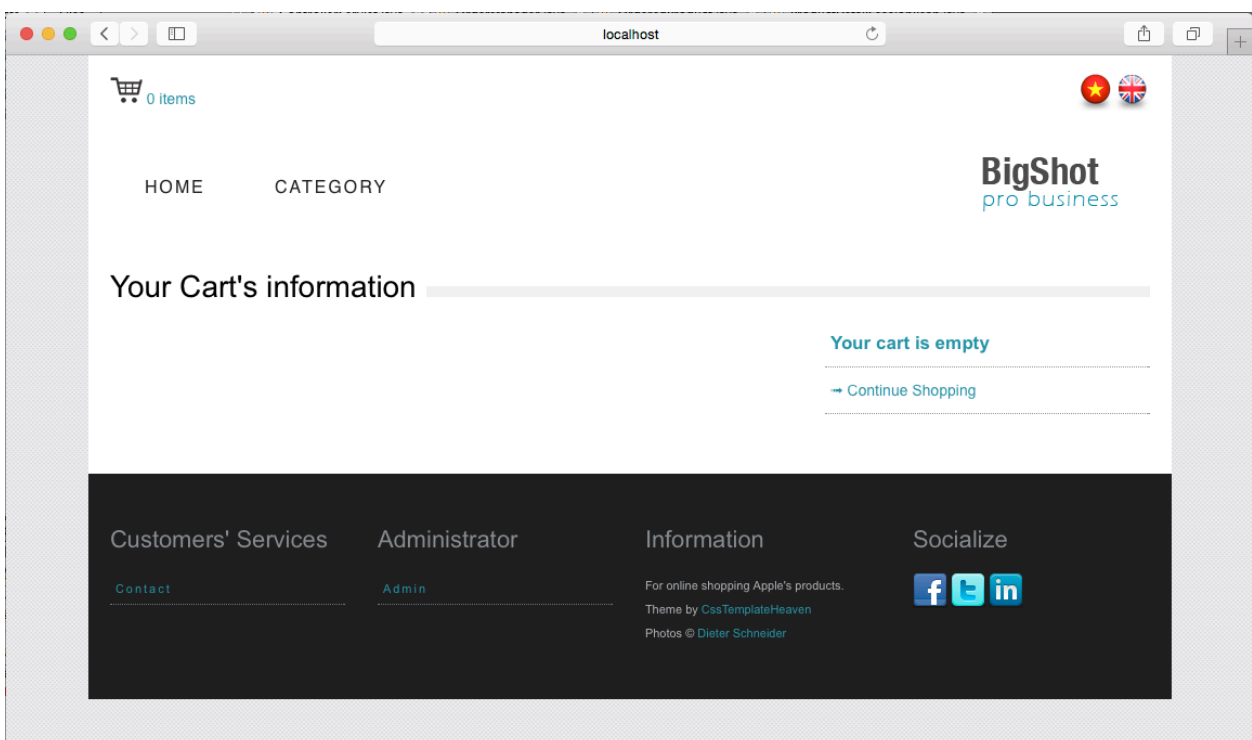
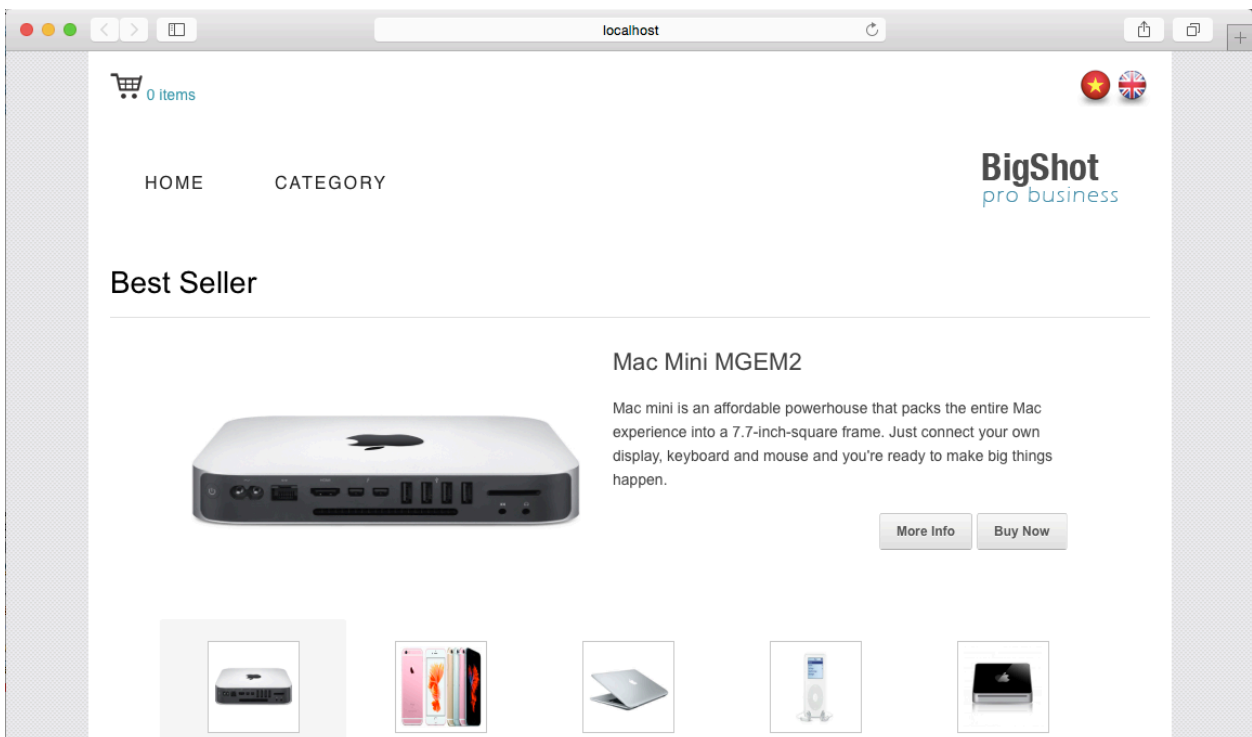
    if (!productId.isEmpty()) {

        Product product =
productSB.find(Integer.parseInt(productId));
        cart.addItem(product);
    }
    String userView = (String) session.getAttribute("view");
    userPath = userView;
}
```

Chúng ta lưu thông tin đơn hàng thông qua attribute 'cart' của session.

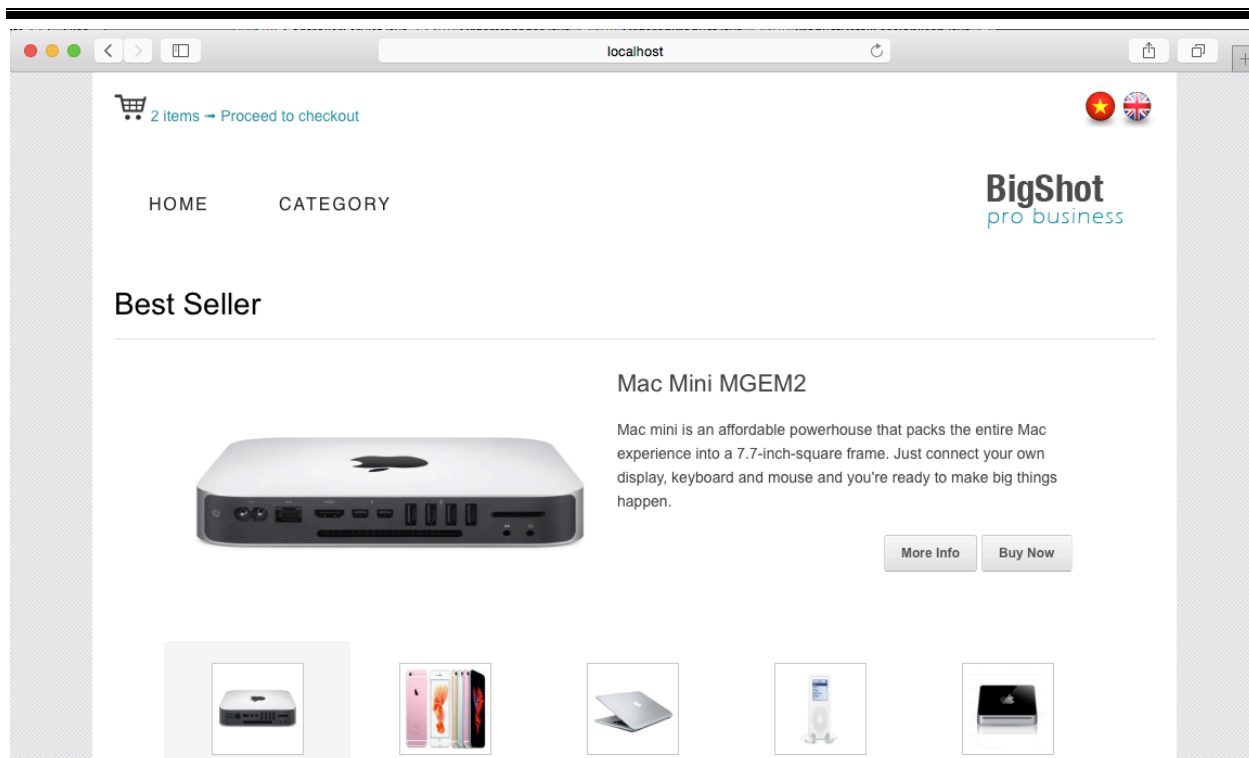
Model-View-Control đã hoàn thành, chúng ta chạy chương trình và review kết quả như sau:

Giỏ hàng chưa có sản phẩm nào:



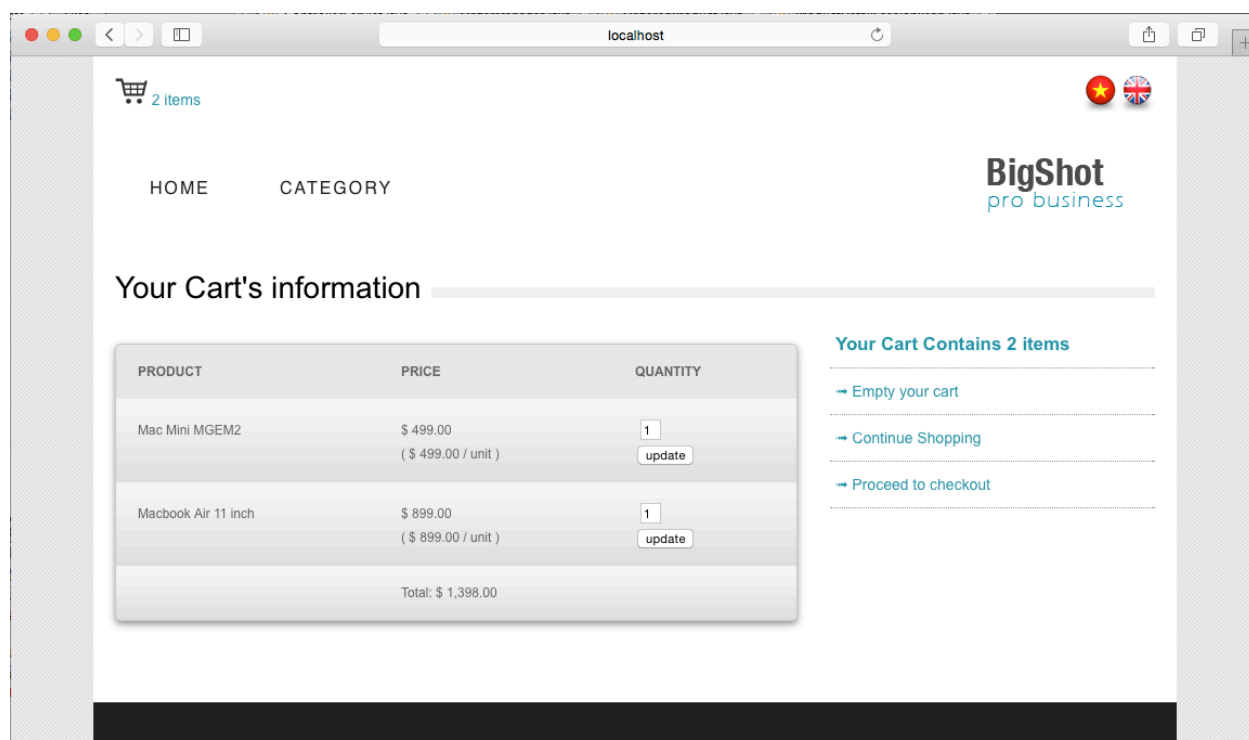
Với xử lý continue shopping, chúng ta đưa users quay trở lại trang index.jsp hoặc trang category.jsp nếu người dùng đang select 1 category nào đó để xem các product.

Thực hiện thêm product vào giỏ hàng, ấn buy now trên trang index.jsp hoặc add to cart trên product.jsp



Đoạn xử lý thêm cho checkout sẽ được thực hiện sau.

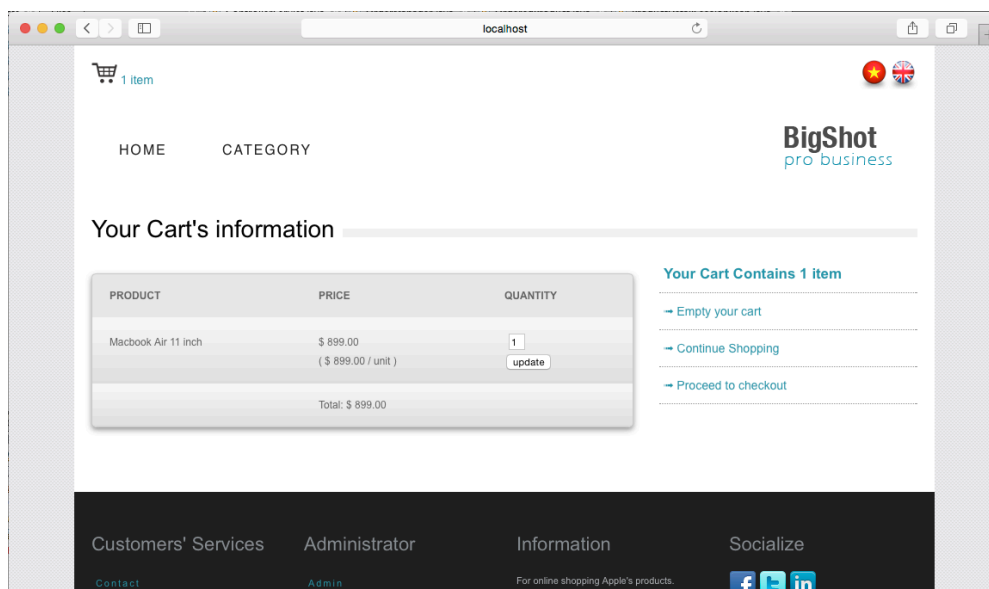
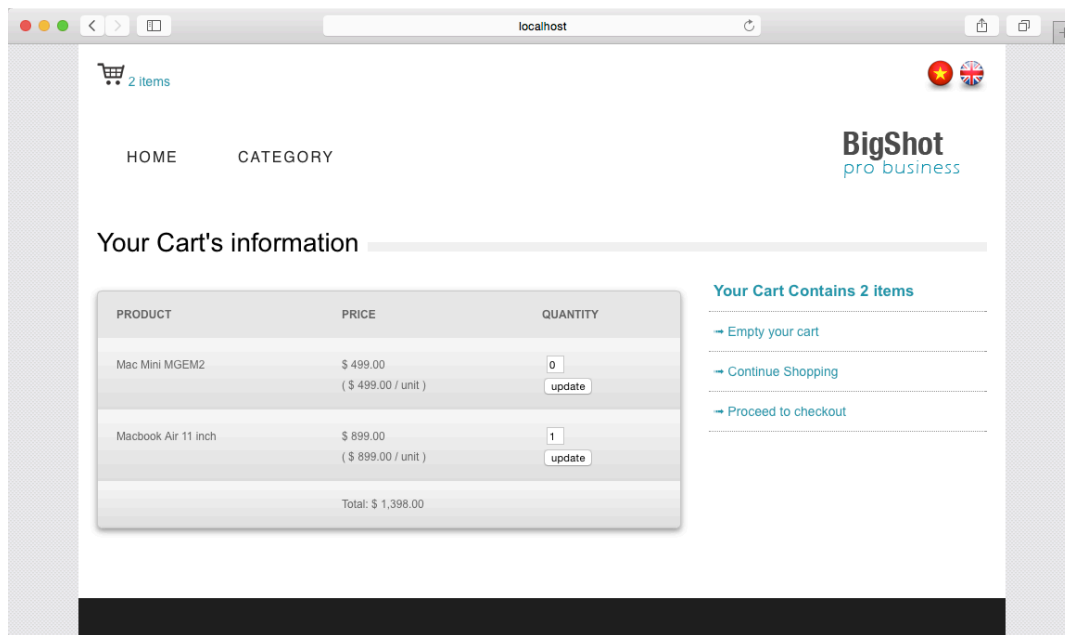
Bây giờ view giỏ hàng:



User sẽ có 3 lựa chọn tiếp theo: tiếp tục mua hàng, clear giỏ hàng hoặc checkout.

PHẦN 2: BÀI TẬP LÀM THÊM

Thực hiện cập nhật giỏ hàng cho button update trên viewCart.jsp. Khi quantity = 0 → Xóa sản phẩm khỏi đơn hàng. Thay đổi quantity → Cập nhật lại số lượng sản phẩm trong giỏ hàng và giá.



Gợi ý: trong ControllerServlet.java, thêm xử lý cho updateCart.

HẾT