

Lab 08

Thiết kế giao diện và cơ sở dữ liệu

Mục tiêu

- Thiết kế giao diện với jspf (jsp fragments)
- Thiết kế CSDL dựa trên một số chức năng đã phân tích trong tài liệu SRS
- Thiết kế trang view sản phẩm

Phần I Bài tập step by step

Bài 1.1

Thiết kế giao diện cơ bản của trang web và các bảng trong CSDL

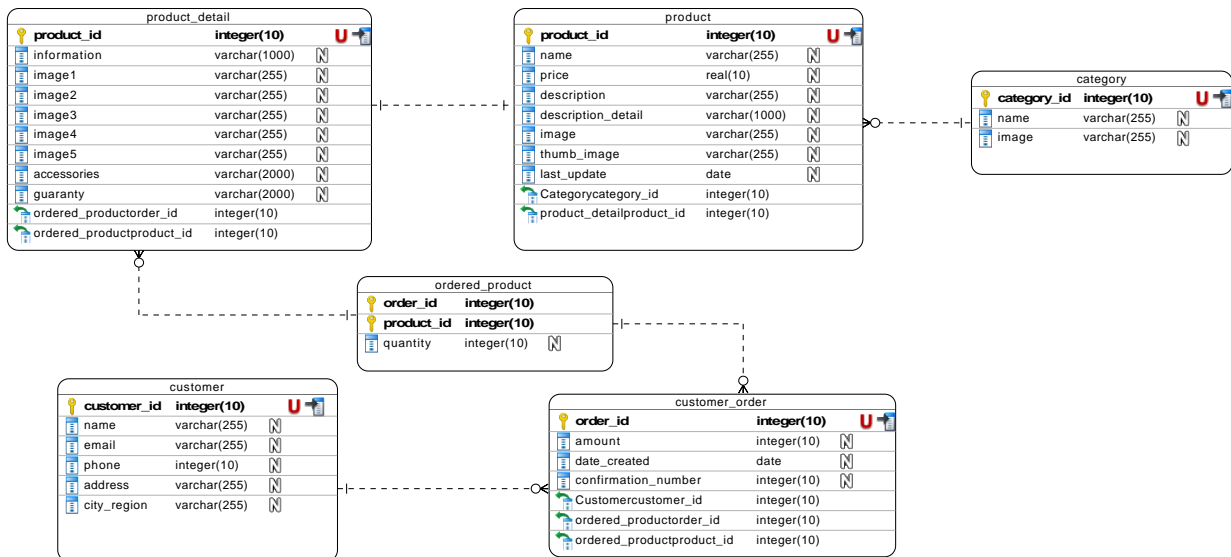
STEP 1: Tạo CSDL emarket trên MySQL hoặc SQL Server.

Ứng dụng TMĐT cần có các chức năng cơ bản sau:

- Quản lý người dùng
- Quản lý sản phẩm
- Quản lý đơn hàng
- Xem thông tin sản phẩm theo hạng mục, xem thông tin chi tiết sản phẩm
- Quản lý giỏ hàng

Chúng ta cần 6 tables: customer, category, customer_order, ordered_product, product và product_detail.

Tạo 6 bảng và đặt quan hệ giữa các bảng theo sơ đồ sau:



STEP 2: Tạo Connection Pool và JNDI DataSource cho project

Mở GlassFish Admin Console -> (hoặc localhost:4848) -> JDBC

JDBC Connection Pools -> New -> Đặt tên Connection Pool là MySQL/eMarket.

Resource Type: javax.sql.DataSource

Chọn Provider: MySQL (nếu dùng SQLServer thì chọn Microsoft SQL Server).

Thêm các thông tin về database:

Server name: localhost

Port: 3306 (MySQL)

Database Name: eMarket

User: root

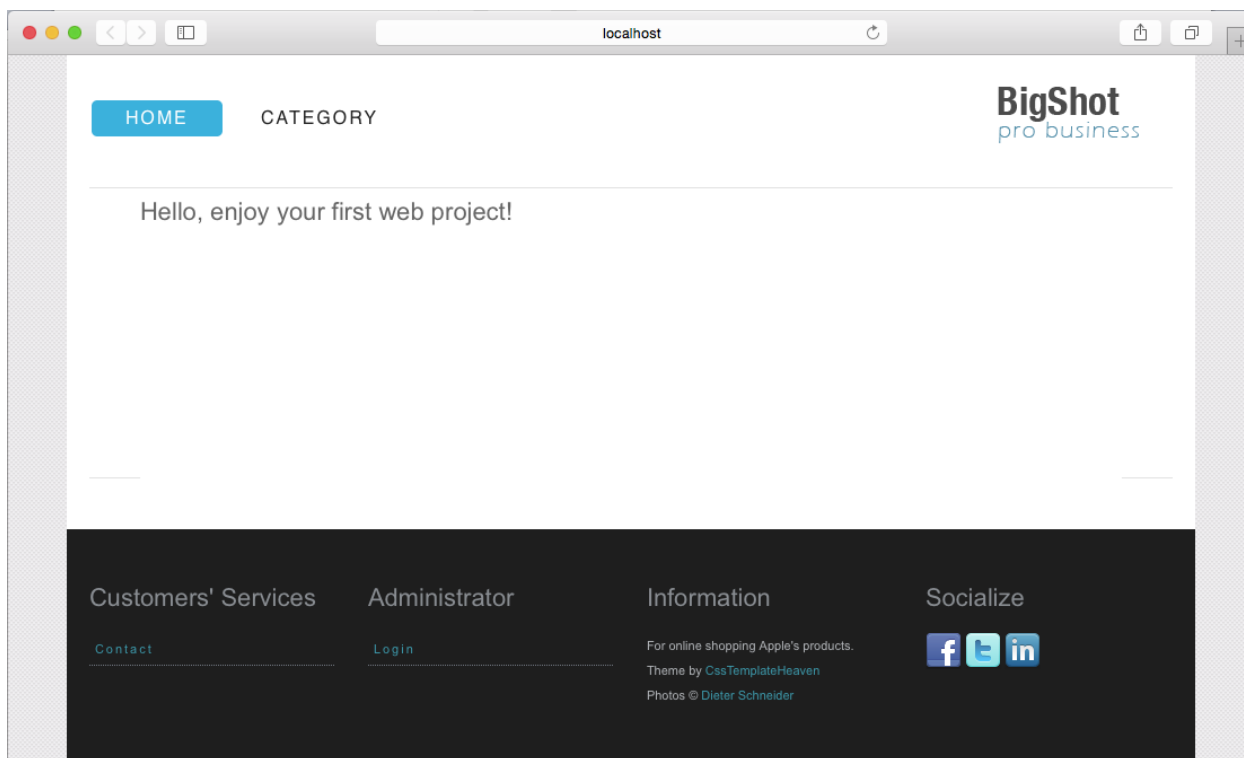
Password: 123456

Sau đó chọn JDBC Resources -> New JNDI Name: jdbc/eMarket

STEP 3: Tạo giao diện cơ bản cho trang web

Các trang web sẽ có chung một số phần giao diện, chỉ khác nhau về nội dung (body). Trong phần này chúng ta sẽ thiết kế các phần giao diện chung này và lưu trong các file jspf (jsp-fragments).

Trang web có giao diện như sau:



Phần chung sẽ gồm 2 phần: header và footer. Chúng ta sẽ tạo ra hai file header.jspf và footer.jspf.

Mở thư mục WEB-INF, tạo thư mục jspf chứa header và footer.

Tạo 2 trang header.jspf và footer.jspf.

Trong file web.xml, config để các trang web của ứng dụng đều chứa hai trang header và footer nói trên, phần config này có thể sẽ có thêm config cho giao diện của admin sau này:

```
<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>
    <!-- và các trang khác -->
    <include-prelude>/WEB-INF/jspf/header.jspf</include-prelude>
```

```

        <include-coda>/WEB-INF/jspf/footer.jspf</include-coda>
    </jsp-property-group>

    <!-- optional for admin GUI -->
    <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>

```

Trang header.jspf:

```

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

<%@page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <!-- CSS Files -->
        <link rel="stylesheet" type="text/css" media="screen" href="css/style.css">
        <link rel="stylesheet" type="text/css" media="screen" href="menu/css/simple_menu.css">
        <link rel="stylesheet" href="css/nivo-slider.css" type="text/css" media="screen">
        <link rel="stylesheet" type="text/css" href="boxes/css/style6.css">
        <link rel="stylesheet" type="text/css" href="js/fancybox/jquery.fancybox.css" media="all">
        <!-- JS Files -->
        <!-- ... -->
    </head>
    <body>
        <div class="header">
            <div id="site_title">
                <br>
            </div>
            <ol id="menu">
                <li class="active_menu_item"><a href="#" style="color:#FFF">Home</a></li>
                <li><a href="#">Category</a>
                    <ol>
                        <li><a href="#">Mac</a>
                            <ol>
                                <li><a href="#">iMac</a></li>
                                <li><a href="#">Mac Pro</a></li>
                                <li><a href="#">MacBook Air</a></li>
                                <li><a href="#">MacBook Pro</a></li>
                            </ol>
                        </li>
                        <li><a href="#">iPad</a></li>
                        <li><a href="#">iPhone</a></li>
                        <li><a href="#">Accessories</a></li>
                    </ol>
                </li>
            </ol>
        </div>

```

Trang footer.jspf:

```
<div id="footer">
  <!-- First Column -->
  <div class="one-fourth">
    <h3>Customers' Services</h3>
    <ul class="footer_links">
      <li><a href="contact.jsp">Contact</a></li>
    </ul>
  </div>
  <!-- Second Column -->
  <div class="one-fourth">
    <h3>Administrator</h3>
    <ul class="footer_links">
      <li><a href="#">Login</a></li>
    </ul>
  </div>
  <!-- Third Column -->
  <div class="one-fourth">
    <h3>Information</h3>
    For online shopping Apple's products.
    <div id="social_icons"> Theme by <a href="http://www.csstemplateheaven.com">CssTemplateHeaven</a><br>
      Photos & <a href="http://dieterschneider.net">Dieter Schneider</a> </div>
  </div>
  <!-- Fourth Column -->
  <div class="one-fourth last">
    <h3>Socialize</h3>
       </div>
  <div style="clear:both"></div>
</div>
<!-- END footer -->
</body>
</html>
```

Toàn bộ nội dung của các trang sẽ nằm trong cặp thẻ <div id="container">...</div>

Ví dụ, trang index.jsp như sau:

```
<div id="container">

  <div id="prod_wrapper">

    <div id="panes">

      <h5>Hello, enjoy your first web project!</h5>

    </div>

  </div>

</div>
```

NOTE: Các nhóm có thể thiết kế giao diện khác nhau, đây chỉ là một trong các template và file mẫu của lab guide, cần phải có các file js và css đi kèm với template này.

BÀI 1.2:

Mục tiêu:

- **Hiển thị một vài sản phẩm mới**
- **Hiển thị category trên trang index.jsp**

STEP 1: Config JNDI trong file web.xml

```
<resource-ref>  
    <description>DB Connection</description>  
    <res-ref-name>jdbc/eMarket</res-ref-name>  
    <res-type>javax.sql.ConnectionPoolDataSource</res-type>  
    <res-auth>Container</res-auth>  
</resource-ref>
```

STEP 2: Tạo entity bean cho table Product

Tạo package entity -> Tạo java Class Product.java

```
package entity;

import java.util.Date;

/**
 *
 * @author ThanDieu
 */
public class Product {
    private int productID;
    private String name;
    private float price;
    private String description;
    private Date lastUpdate;
    private int categoryID;
    private String image;
    private String thumbImage;
    private String descriptionDetail;

    public int getProductID() {
        return productID;
    }

    public void setProductID(int productID) {
        this.productID = productID;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public float getPrice() {
        return price;
    }

    /**...*/
}
```

STEP 3: Tạo ProductBLL thực hiện các business logic liên quan đến table Product

Tạo package bll -> Tạo java class ProductBLL.java

```
package bll;

import entity.Product;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;

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

    public List<Product> getNewProducts(int number){
        try{
            Context initContext = new InitialContext();
            DataSource ds = (DataSource) initContext.lookup("java:comp/env/jdbc/eMarket");
            Connection conn = ds.getConnection();
            Statement sttm = conn.createStatement();
            String sql = "select * from Product";
            ResultSet rs = sttm.executeQuery(sql);
            ArrayList<Product> prods = new ArrayList<>();
            while (rs.next()) {
                Product p = new Product();
                p.setProductID(rs.getInt("product_id"));
                p.setCategoryID(rs.getInt("category_id"));
                p.setPrice(rs.getFloat("price"));
                p.setDescription(rs.getString("description"));
                p.setImage(rs.getString("image"));
                p.setName(rs.getString("name"));
                p.setThumbImage(rs.getString("thumb_image"));
                p.setLastUpdate(rs.getDate("last_update"));
                p.setDescriptionDetail(rs.getString("description_detail"));
                prods.add(p);
            }
            return prods;
        } catch (SQLException ex) {
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(ProductBLL.class.getName()).log(Level.SEVERE, null, ex);
        }
        return null;
    }
}
```

STEP 4: Tạo servlet xử lý request

Tạo package controller -> Tạo servlet ControllerServlet.

Để đảm bảo ngay khi ứng dụng vừa run (trang index.jsp được gọi tới theo mặc định) thì servlet này cũng được load chúng ta sử dụng thuộc tính “loadOnStartup=1”

Ngay khi servlet được khởi tạo, chúng ta sẽ tìm danh sách 5 sản phẩm mới từ table Product để hiển thị trên index.jsp. Công việc này được thực hiện bởi ProductBLL và đặt trong hàm init() của servlet.

```
package controller;

import bll.ProductBLL;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletConfig;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 *
 * @author ThanDieu
 */
@WebServlet(name = "ControllerServlet",
        loadOnStartup = 1,
        urlPatterns = {"/ControllerServlet"})
public class ControllerServlet extends HttpServlet {

    @Override
    public void init(ServletConfig servletConfig) throws ServletException {

        super.init(servletConfig);

        // store new product list in servlet context
        ProductBLL productBLL = new ProductBLL();
        getServletContext().setAttribute("newProducts", productBLL.getNewProducts(5));
    }

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

    }

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

    }

}
```

STEP 5: Tạo view các sản phẩm mới trong index.jsp

Chúng ta sẽ sử dụng thẻ `<c:forEach>` của JSTL.

Chúng ta cũng đặt đường dẫn của image resources thông qua context-param trong web.xml:

```
<context-param>
  <param-name>imgProductPath</param-name>
  <param-value>img/demo/</param-value>
</context-param>
```

Để lấy về đường dẫn tới image resources trong config này, chúng ta sẽ sử dụng implicit objects của JSP: `initParam.[tên tham số trong config]`

File index.jsp:

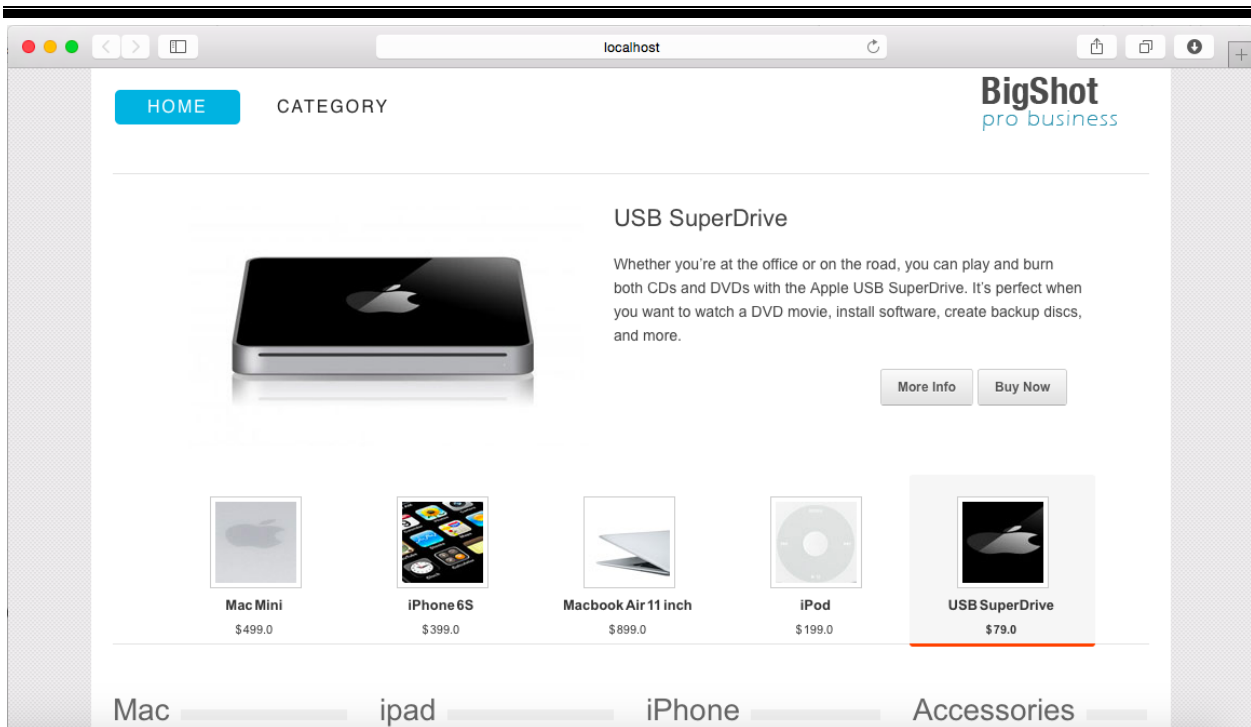
```
<div id="container">
  <div id="prod_wrapper">
    <div id="panes">
      <c:forEach var="product" items="${newProducts}">
        <div>
          
          <h5>${product.getName()}</h5>
          <p>${product.getDescription()}</p>
          <p style="text-align:right; margin-right: 16px">
            <a href="#" 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>
<div class="one-fourth">
  <div class="heading_bg">
    <h2>Mac</h2>
  </div>
   </div>
<!-- Second Column -->
<div class="one-fourth">
  <div class="heading_bg">
    <h2>iPad</h2>
  </div>
   </div>
<!-- Third Column -->
<div class="one-fourth">
  <div class="heading_bg">
    <h2>iPhone</h2>
  </div>
   </div>
<!-- Fourth Column -->
<div class="one-fourth last">
  <div class="heading_bg">
    <h2>Accessories</h2>
  </div>
   </div>
<div style="clear:both; height: 40px"></div>
</div>

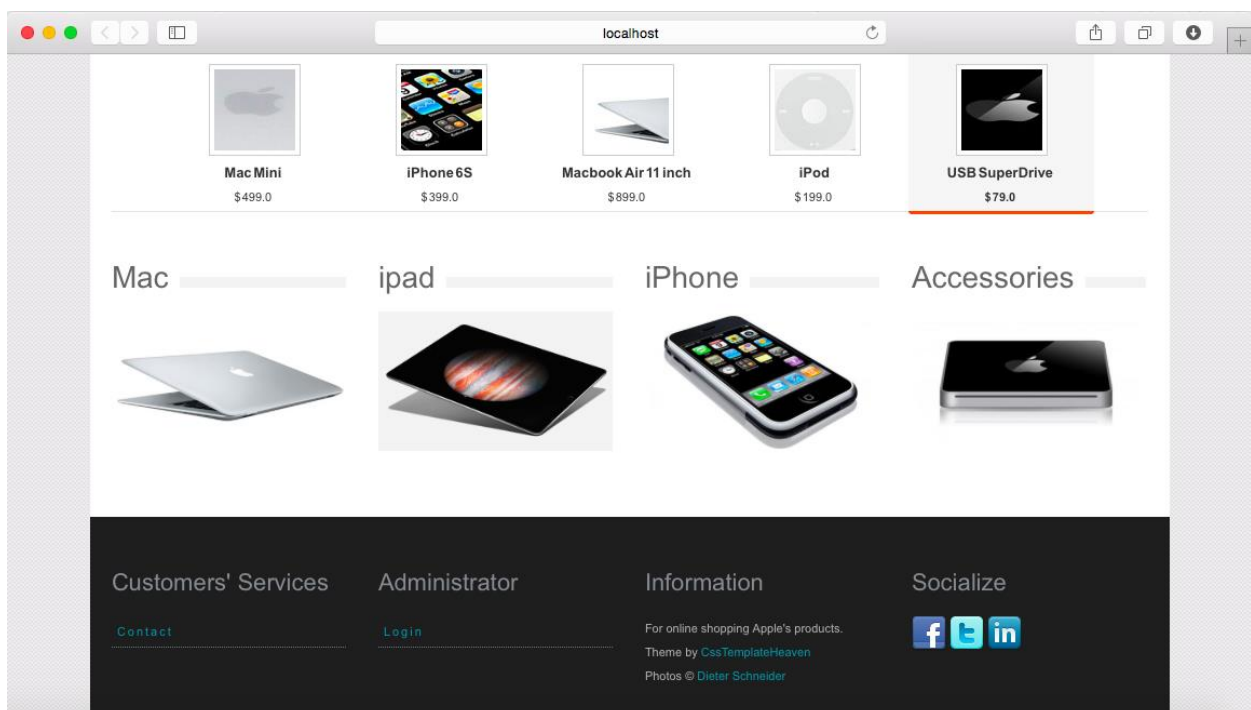
```

Chạy chương trình và hiển thị kết quả (demo)



Phần bài tập làm thêm:

Hiển thị các category của sản phẩm như sau:



Thông tin category lấy từ bảng Category.

Gợi ý: làm tương tự như với Product, tạo một attribute mới gán cho context trong hàm init của ControllerServlet.

NOTE: Mọi thông tin về template giao diện và các image chỉ là demo. Khuyến khích các nhóm tự tạo giao diện khác với lab guide.

HẾT