

# PROJECT I

Java Web using Servlet/JSP

Bùi Quang Đức - 20173023

Giảng viên hướng dẫn: TS. Nguyễn Bình Minh

Website Java bán thực phẩm

Ngày 11 tháng 12 năm 2019

# Overview

- 1 Một số khái niệm
- 2 Nội dung của sản phẩm
- 3 Kiểm tra và thử nghiệm sản phẩm

# Servlets là gì?

- Công nghệ Servlet được sử dụng để tạo ra ứng dụng web (nằm ở phía máy chủ và tạo ra trang web động).
- Sử dụng Servlet, bạn có thể thu thập thông tin đầu vào từ người dùng thông qua các trang web, hiển thị các bản ghi từ một cơ sở dữ liệu hoặc một nguồn khác.
- Công nghệ servlet là mạnh mẽ và có khả năng mở rộng. Trước Servlet, ngôn ngữ kịch bản CGI (Common Gateway Interface) được sử dụng phổ biến như một ngôn ngữ lập trình phía máy chủ.
- Bạn phải viết code Java + HTML trong cùng một file tạo cảm giác rối rắm.
- Nếu bạn đang đi theo hướng java web thì bạn nên biết về servlet vì nó là core của java web.
- Nhưng bạn chỉ cần học servlet ở mức độ biết là đủ, không cần đi sâu. Vì hiện nay có khá nhiều framework mạnh mẽ được sử dụng để tạo ra một trang web bằng java. Như Spring MVC Web, Struts, ...

# JSP là gì?

- JSP (Java server page hay Java Scripting Preprocessor – tạm dịch là “Bộ tiền xử lý văn lệnh Java”) là một công nghệ Java cho phép các nhà phát triển tạo nội dung HTML, XML hay một số định dạng khác khiến cho trang web sinh động hơn.
- Ưu điểm của JSP
  - Hỗ trợ cho việc thiết kế giao diện web dễ dàng hơn.
  - Có vai trò lớn trong việc cho phép thiết kế web tạo nên những trang web động.
  - Có thể viết một nơi và chạy bất cứ nơi nào (Write once, bugs everywhere).
- Nhược điểm của JSP
  - Tiêu tốn dung lượng lưu trữ phía server gấp đôi.
  - Lần đầu truy cập vào trang JSP sẽ mất nhiều thời gian chờ.

- Ưu điểm
  - ① Nhẹ tiết kiệm băng thông (không sử dụng viewstate) và đơn giản

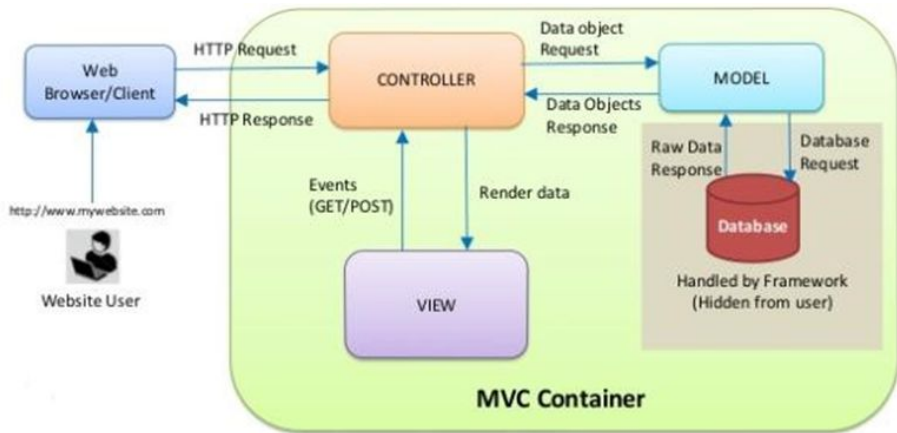
- Ưu điểm
  - ① Nhẹ tiết kiệm băng thông (không sử dụng viewstate) và đơn giản
  - ② Kiểm tra dễ dàng, rà soát, cập nhật

- Ưu điểm
  - ① Nhẹ tiết kiệm băng thông (không sử dụng viewstate) và đơn giản
  - ② Kiểm tra dễ dàng, rà soát, cập nhật
  - ③ Phân tách rõ Model - View - Control

- Ưu điểm
  - ① Nhẹ tiết kiệm băng thông (không sử dụng viewstate) và đơn giản
  - ② Kiểm tra dễ dàng, rà soát, cập nhật
  - ③ Phân tách rõ Model - View - Control
- Nhược điểm MVC thường dùng vào những dự án lớn. Với ứng dụng nhỏ MVC gây công kênh, tốn thời gian trong quá trình phát triển, trung chuyển dữ liệu.



# Mô hình MVC



- Cùng với sự phát triển của Internet, kinh doanh online trở thành ngành nghề đem lại lợi nhuận cao. Kèm theo đó, cần phải có một website bán hàng để tăng độ nhận biết cho sản phẩm bạn đang kinh doanh. Vậy nên website bán hàng thực sự quan trọng.
- Hiểu rõ hơn về Java Web: Servlet, JSP...
- Hiểu rõ hơn về O.O.P, JDBC, HTML, CSS, JavaScript.
- Thực hành lập trình.

# Những gì đã được sử dụng?

- Tools: NetBeans IDE, MySQL Workbench, TextEditor, Browser, Glassfish Server. . .
- MariaDB, Servlet, JSP, JDBC, JavaCore. . .
- Template: HTML, CSS, Bootstrap, JavaScript.

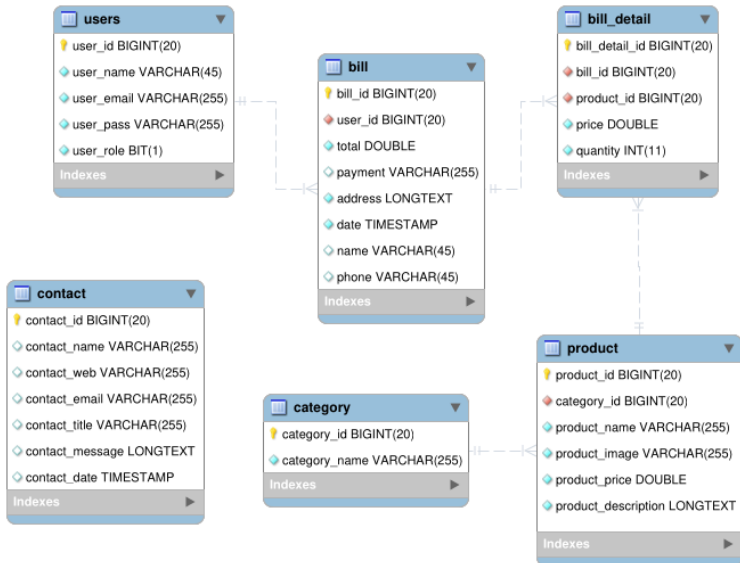
- Đối với khách vãng lai (guest)
  - Xem danh sách sản phẩm, xem thông tin chi tiết sản phẩm
  - Tìm kiếm sản phẩm theo tên, danh mục
  - Xem thống kê sản phẩm theo danh mục
  - Đăng ký tài khoản

- Đối với khách vắng lai (guest)
  - Xem danh sách sản phẩm, xem thông tin chi tiết sản phẩm
  - Tìm kiếm sản phẩm theo tên, danh mục
  - Xem thống kê sản phẩm theo danh mục
  - Đăng ký tài khoản
- Đối với khách hàng đã đăng nhập tài khoản (user)
  - Có tài khoản, có thể chỉnh sửa tài khoản, đăng nhập, đăng xuất
  - Xem thông tin tài khoản
  - Đặt mua sản phẩm (đặt hàng ngay hoặc đặt theo số lượng)
  - Xem đơn hàng, xóa đơn hàng
  - Thanh toán hóa đơn
  - Xem lịch sử hóa đơn
  - Gửi phản hồi cho Website

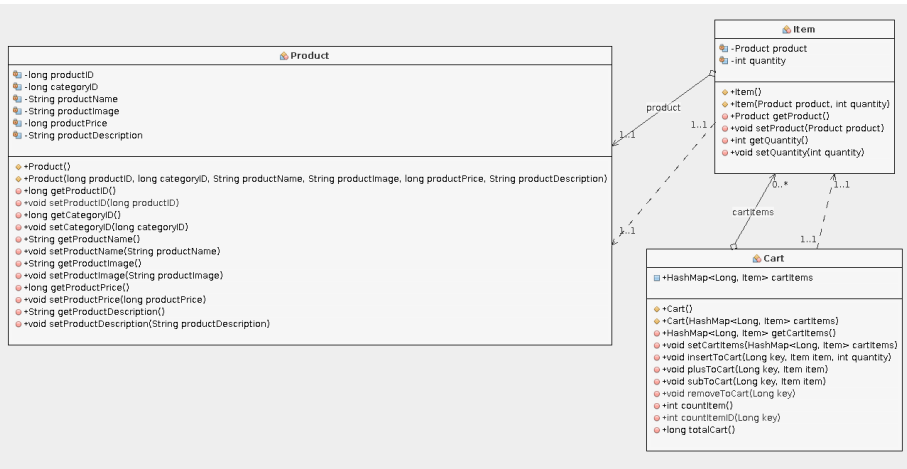
# Chức năng

- Đối với khách vãng lai (guest)
  - Xem danh sách sản phẩm, xem thông tin chi tiết sản phẩm
  - Tìm kiếm sản phẩm theo tên, danh mục
  - Xem thống kê sản phẩm theo danh mục
  - Đăng ký tài khoản
- Đối với khách hàng đã đăng nhập tài khoản (user)
  - Có tài khoản, có thể chỉnh sửa tài khoản, đăng nhập, đăng xuất
  - Xem thông tin tài khoản
  - Đặt mua sản phẩm (đặt hàng ngay hoặc đặt theo số lượng)
  - Xem đơn hàng, xóa đơn hàng
  - Thanh toán hóa đơn
  - Xem lịch sử hóa đơn
  - Gửi phản hồi cho Website
- Đối với admin/manager
  - Đăng nhập website
  - Xem, thêm, sửa xóa thông tin của sản phẩm, tài khoản, danh mục sản phẩm
  - Xem thông tin phản hồi, lịch sử hóa đơn

# Cơ sở dữ liệu



# UML Example





```
// JSP Scriptint Element in Cart.jsp
<%
Cart cart = (Cart) session.getAttribute("cart");
if (cart == null) {
    cart = new Cart();
    session.setAttribute("cart", cart);
}
%>
<%for (Map.Entry<Long, Item> list :
        cart.getCartItems().entrySet()) {%>
<tr>
    <td class="product-col">
        <a href="CartServlet?method=get&command=remove&productID=
            <%=list.getValue().getProduct().getProductID()%>">
            </a>
        <div class="pc-title">
            <h4><%=list.getValue().getProduct().getProductName()%></h4>
            <p>$<%=list.getValue().getProduct().getProductPrice()%></p>
        </div>
```

```

</td>
<td class="quy-col">
    <div class="quantity">
        <div class="pro-qty">
            <input type="text"
                value="<%=list.getValue().getQuantity()%>">
        </div>
    </div>
</td>
<td class="size-col">
    <h4><%=list.getValue().getProduct().getCategoryID()%></h4>
</td>
<td class="total-col">
    <h4><%=list.getValue().getProduct().getProductPrice() *
        list.getValue().getQuantity()%></h4>
</td>
</tr>
<%}%>

```

---

```

// Code Encryption Password

```

```

public static String encryption(String str) {

```

```
byte[] defaultBytes = str.getBytes();
try {
    MessageDigest algorithm =
        MessageDigest.getInstance("MD5");
    algorithm.reset();
    algorithm.update(defaultBytes);
    byte messageDigest[] = algorithm.digest();
    StringBuffer hexString = new StringBuffer();
    for (int i = 0; i < messageDigest.length; i++) {
        String hex = Integer.toHexString(0xFF &
            messageDigest[i]);
        if (hex.length() == 1) {
            hexString.append('0');
        }
        hexString.append(hex);
    }
    str = hexString + "";
} catch (NoSuchAlgorithmException e) {
    e.printStackTrace();
}

return str; }
```

---

```
// Code Splitting Name with Part File
private String getFileName(final Part part) {
    final String partHeader =
        part.getHeader("content-disposition");
    LOGGER.log(Level.INFO, "Part Header = {0}", partHeader);
    for (String content :
        part.getHeader("content-disposition").split(";")) {
        if (content.trim().startsWith("filename")) {
            return content.substring(
                content.indexOf('=') + 1).trim().replace("\\\"",
                "");
        }
    }
    return null;
}
```

---

The End