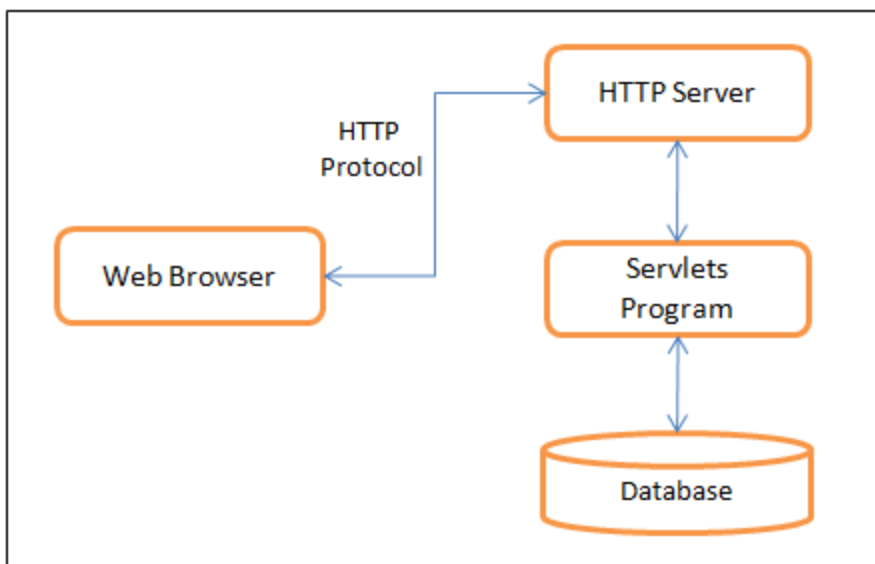
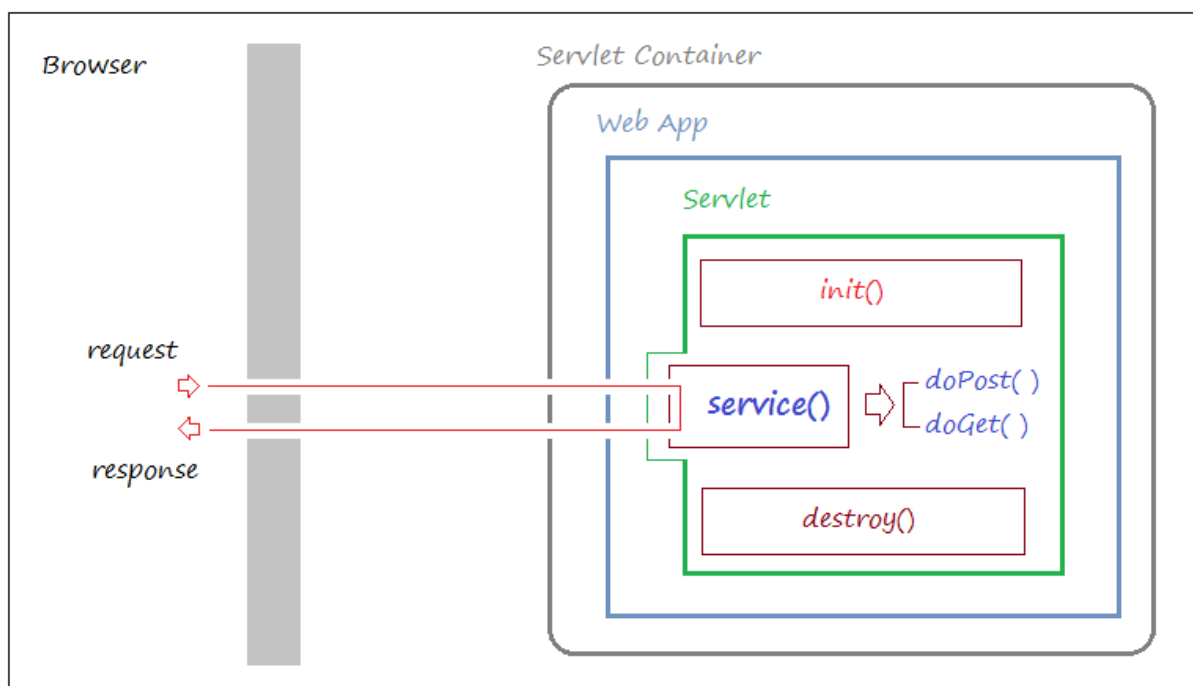


I. MỤC TIÊU:

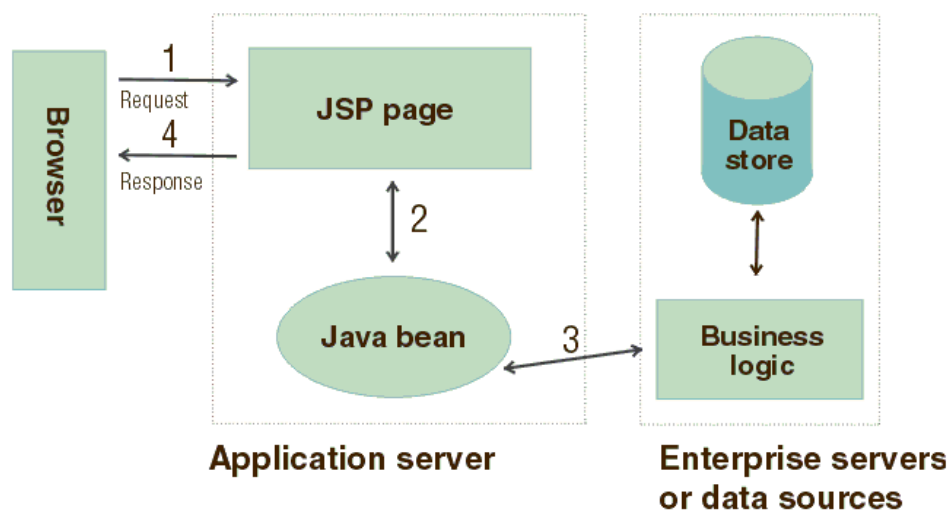
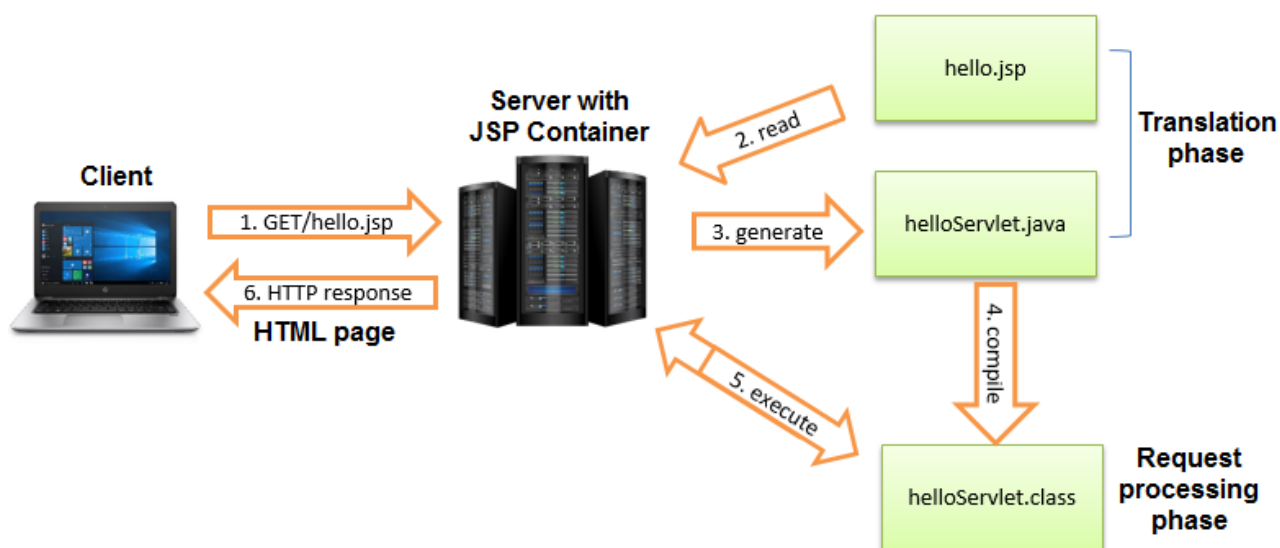
- Xây dựng ứng dụng web đơn giản dùng JSP/Servlet

II. TÓM TẮT LÝ THUYẾT:**1. Servlet:****a. Kiến trúc:****b. Vòng đời của servlet:**

- **Phương thức init():** được gọi chỉ một lần để khởi tạo servlet, dùng để tạo hoặc tải một số dữ liệu sẽ được sử dụng trong suốt vòng đời của servlet.
- **Phương thức service():** để xử lý các yêu cầu đến từ khách hàng (trình duyệt) và trả về kết quả. Tùy thể yêu cầu máy khách là GET hay POST mà phương thức service() sẽ gọi phương thức xử lý tương ứng.
- **Phương thức doPost():** dùng để xử lý yêu cầu POST.
- **Phương thức doGet():** dùng để xử lý yêu cầu GET.
- **Phương thức destroy():** chỉ được gọi một lần vào cuối chu kỳ sống của một servlet, dùng để ngắt kết nối CSDL, giải phóng tài nguyên cho hệ thống

c. Quá trình xử lý yêu cầu của servlet:**d. Một số đối tượng thường dùng:**

- **HttpRequest:** nhận tham số từ form hoặc từ trang khác chuyển tới
 - **Phương thức `getParameter()`** – Gọi phương thức `request.getParameter()` để lấy giá trị của một tham số của form
 - **Phương thức `getParameterValues()`** – Gọi phương thức này nếu tham số xuất hiện nhiều lần và trả về nhiều giá trị, ví dụ checkbox.
 - **Phương thức `getParameterNames()`** – Gọi phương thức này nếu bạn muốn có một danh sách đầy đủ của tất cả các tham số trong yêu cầu hiện tại.
- **HttpResponse:** dùng để trả kết quả về cho máy khách.
`response.setContentType("text/html");`
`PrintWriter out = response.getWriter();`
- **RequestDispatcher:** chuyển hướng trang

2. JSP:**- Kiến trúc JSP:****- Quá trình xử lý trang JSP:****- Cú pháp JSP:**

- **JSP Scriptlet:** `<% java code %>`
- **JSP Declaration:** `<%! declaration; [declaration;]+ ... %>`
- **JSP Expression:** `<%= expression %>`

- **JSP Directive:** `<%@ directive attribute="value" %>`

STT	Chỉ thị và Mô tả
1	<%@ page ...%> Định nghĩa thuộc tính page-dependent, chẳng hạn như import package, trang lỗi và yêu cầu bộ đệm.
2	<%@ include ...%> Bao gồm tập tin trong giai đoạn biên dịch.
3	<%@ taglib ...%> Khai báo một tag library, chứa các hành động tùy chỉnh, được sử dụng trong trang.

- **JSP Action:** `<jsp:action_name attribute="value" />`

No.	Cú pháp & Mục đích
1	jsp:include Thêm nội dung được include tại thời điểm trang được request.
2	jsp:useBean Tìm hoặc khởi tạo một JavaBean.
3	jsp:setProperty Thiết lập thuộc tính của một JavaBean.
4	jsp:getProperty Chèn thuộc tính của một JavaBean vào đầu ra.
5	jsp:forward Chuyển tiếp người yêu cầu tới một trang mới.
6	jsp:plugin Tạo mã trình duyệt cụ thể mà làm cho một thẻ OBJECT hoặc EMBED cho Java plugin.
7	jsp:element Định nghĩa các phần tử XML động.
8	jsp:attribute Định nghĩa thuộc tính của phần tử XML được tự động định nghĩa.
9	jsp:body Định nghĩa phần thân của phần tử XML được tự động định nghĩa.
10	jsp:text Được sử dụng để viết văn bản mẫu trong các trang JSP và tài liệu.

III. NỘI DUNG THỰC HÀNH:

Trong Lab này sinh viên sẽ được hướng dẫn chi tiết các bước xây dựng trang web đơn giản bằng JSP và Servlet

1. Lab6_1:

- Mở file **Lab6_1.jsp**, đoạn script lấy dữ liệu từ form.

```
<% //Doc gia tri tu form gui len
2   String submit = request.getParameter("submit");
3   String strA = request.getParameter("txtA");
4   String strB = request.getParameter("txtB");
5   String kq = "";
6   if (submit!=null) {
7       double a = Double.parseDouble(strA);
8       double b = Double.parseDouble(strB);
9       switch(submit) {
10          case "+":
11              kq = a + " + " + b + " = " + (a + b);
12              break;
13          }
14   }
15 %>
```

- Chạy ứng dụng, chọn **Lab6_1.jsp**, nhập vào 2 số A, B, kích nút + quan sát kết quả.
- Thiết kế thêm nút - * /
- Viết code xử lý cho các nút vừa tạo.

Bài tập 1: Tạo trang **baitap1.jsp** như hình, viết code xử lý cho các nút trong form.

MONEY CONVERTER	
Amount:	<input type="text" value="20"/>
VND:	<input type="text" value="44000"/>
<input type="button" value="USD"/>	<input type="button" value="JPY"/> <input type="button" value="EUR"/>
1 USD = 22000 VND	
1 JPY = 200 VND	
1 EUR = 28000 VND	

2. Lab6_2:

- Mở Source Package/Lab6_2.java

```
8      public class Lab6_2 extends HttpServlet {
9          protected void processRequest(HttpServletRequest request,
10              HttpServletResponse response)
11              throws ServletException, IOException {
12              response.setContentType("text/html;charset=UTF-8");
13              try (PrintWriter out = response.getWriter()) {
14                  out.println("<!DOCTYPE html>");
15                  out.println("<html>");
16                  out.println("<head>");
17                  out.println("<title>Servlet Lab6_2</title>");
18                  out.println("</head>");
19                  out.println("<body>");
20                  out.println("<form action='Lab6_2'>"
21                      + "Type a name: <input name='name'> "
22                      + "<input type='submit' value='Say Hello'>"
23                      + "</form>");
24                  //Xu ly du lieu tu form
25                  String name = request.getParameter("name");
26                  if (name!=null)
27                      out.println("<h3>Hello, " + name + "</h3>");
28                  out.println("</body>");
29                  out.println("</html>");
30              }
31      }
```

- Chạy ứng dụng, chọn [Lab6_2](#), nhập họ tên, kích nút “Say Hello” quan sát kết quả.

Bài tập 2:

- Tạo Servlet **Baitap2** cho nhập chiều dài, chiều rộng hình chữ nhật, khi nhấn nút Xuly hiển thị diện tích, chu vi.

Dai:	<input type="text"/>
Rong:	<input type="text"/>
<input type="button" value="Xuly"/>	
Diện tích: 20	
Chu vi: 18	

3. Lab6_3:

- Mở file **Lab6_3.jsp**: trang jsp hiển thị form cho user nhập liệu, hiển thị kết quả trả về từ servlet. Hình dưới là đoạn mã nhận dữ liệu từ servlet chuyển qua.

```
<%
2      String kq = (String) request.getAttribute("kq");
3      if (kq==null)
4          kq = "";
5  %>
```

- Mở file **Source Package/servlet/Lab6_3.java**: servlet nhận dữ liệu từ trang jsp gửi lên, xử lý, trả kết quả về cho trang jsp hiển thị.

```
31      protected void processRequest(HttpServletRequest request,
32      HttpServletResponse response) throws ServletException, IOException {
33          String strA = request.getParameter("txtA");
34          String submit = request.getParameter("submit");
35          String strB = request.getParameter("txtB");
36          String kq = "";
37          if (submit!=null) {
38              double a = Double.parseDouble(strA);
39              double b = Double.parseDouble(strB);
40              switch(submit) {
41                  case "+":
42                      kq = a + " + " + b + " = " + (a + b);
43                      break;
44              }
45          }
46          request.setAttribute("kq", kq);
47          RequestDispatcher rd = request.getRequestDispatcher("Lab6_3.jsp");
48          rd.forward(request, response);
49      }
```

- Chạy ứng dụng, chọn **Lab6_3.jsp**, nhập vào 2 số A, B, kích nút +
- Thiết kết thêm nút - * / . Viết code xử lý cho các nút vừa tạo.

Bài tập 3:

- Làm lại bài tập 1, 2 theo mẫu Lab6_3

IV. BÀI TẬP LÀM THÊM:

- Thực hiện trang login cho phép người dùng đăng nhập với user: “khaho” password: “123”.
- Nếu đăng nhập đúng chuyển qua trang success, ngược lại báo lỗi, quay lại trang login.