MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

- Cài đặt được môi trường phát triển ứng dụng web với công nghệ Servlet/JSP
- Tạo và quản lý được dự án web động (Dynamic Web Project)
- Tạo được Servlet, JSP và tổ chức theo mô hình MVC
- Truyền được dữ liệu từ Servlet sang JSP để hiển thị
- Đọc được tham số form đơn giản

PHẦN I: CÀI ĐẶT VÀ TẠO DỰ ÁN

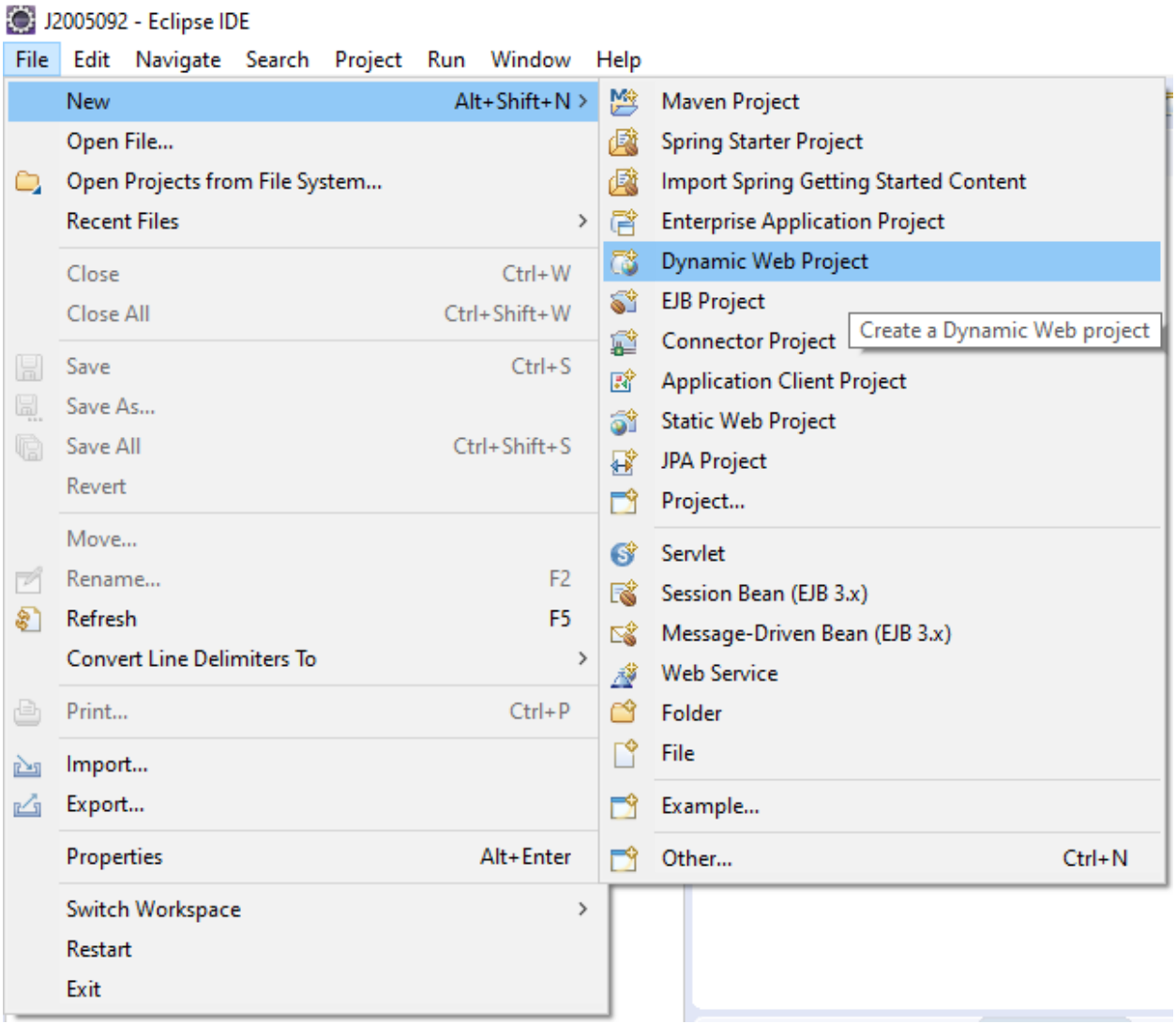
BÀI 1: CÀI ĐẶT MÔI TRƯỜNG PHÁT TRIỂN

Hãy tiến hành cài đặt môi trường theo hướng dẫn trên slide bài giảng. Cụ thể bạn cần download và cài đặt những phần mềm sau

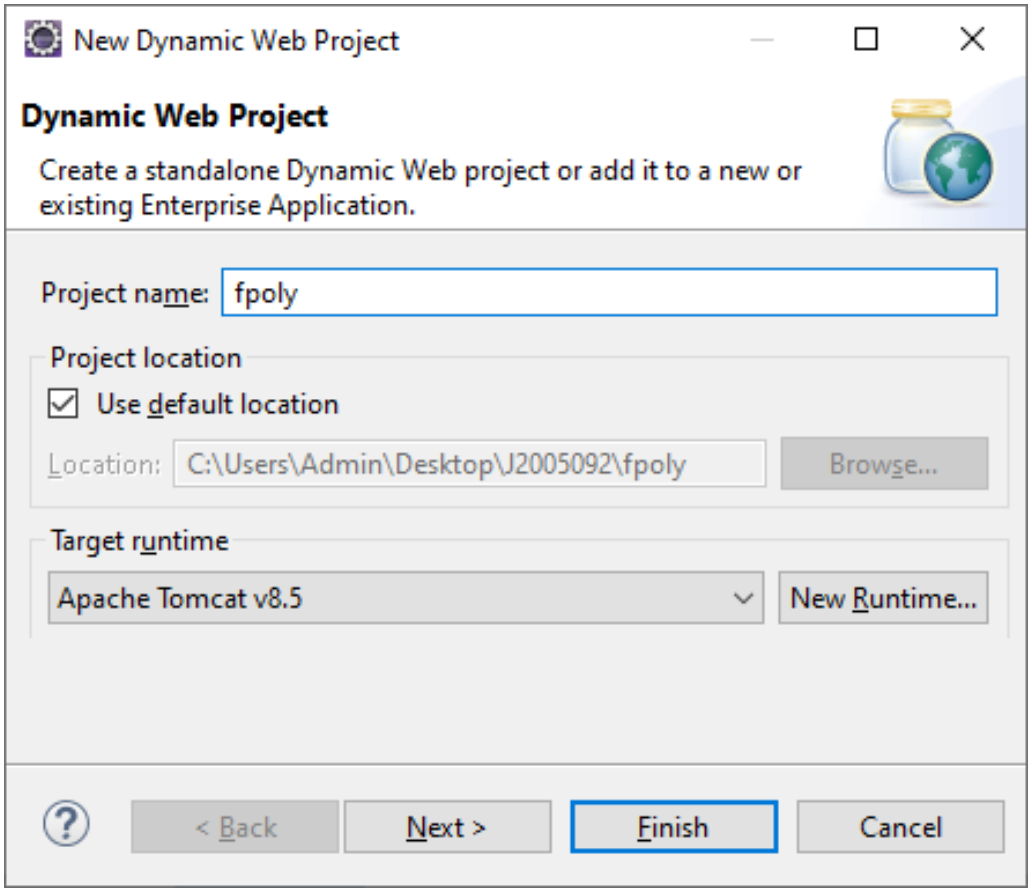
- Cài đặt JDK
- Eclipse for Java EE
- Tích hợp Tomcat vào eclipse
- Cấu hình chọn trình duyệt ngoài
- SQL Server 2008+ (có thể cài đặt sau)

BÀI 2: TẠO DỰ ÁN WEB ĐỘNG (DYNAMIC WEB PROJECT)

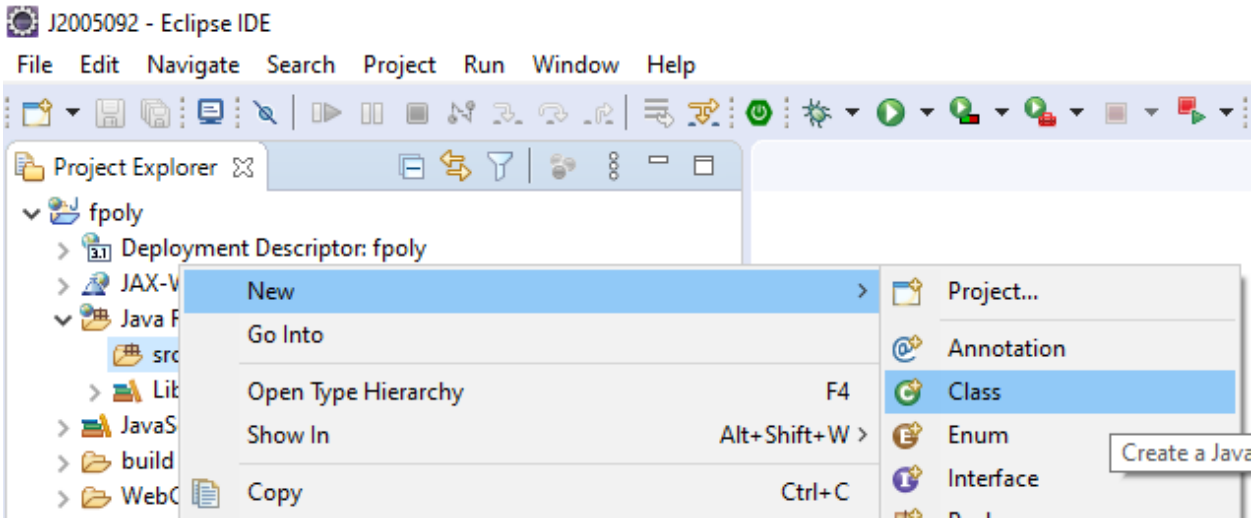
B1: Chọn loại dự án

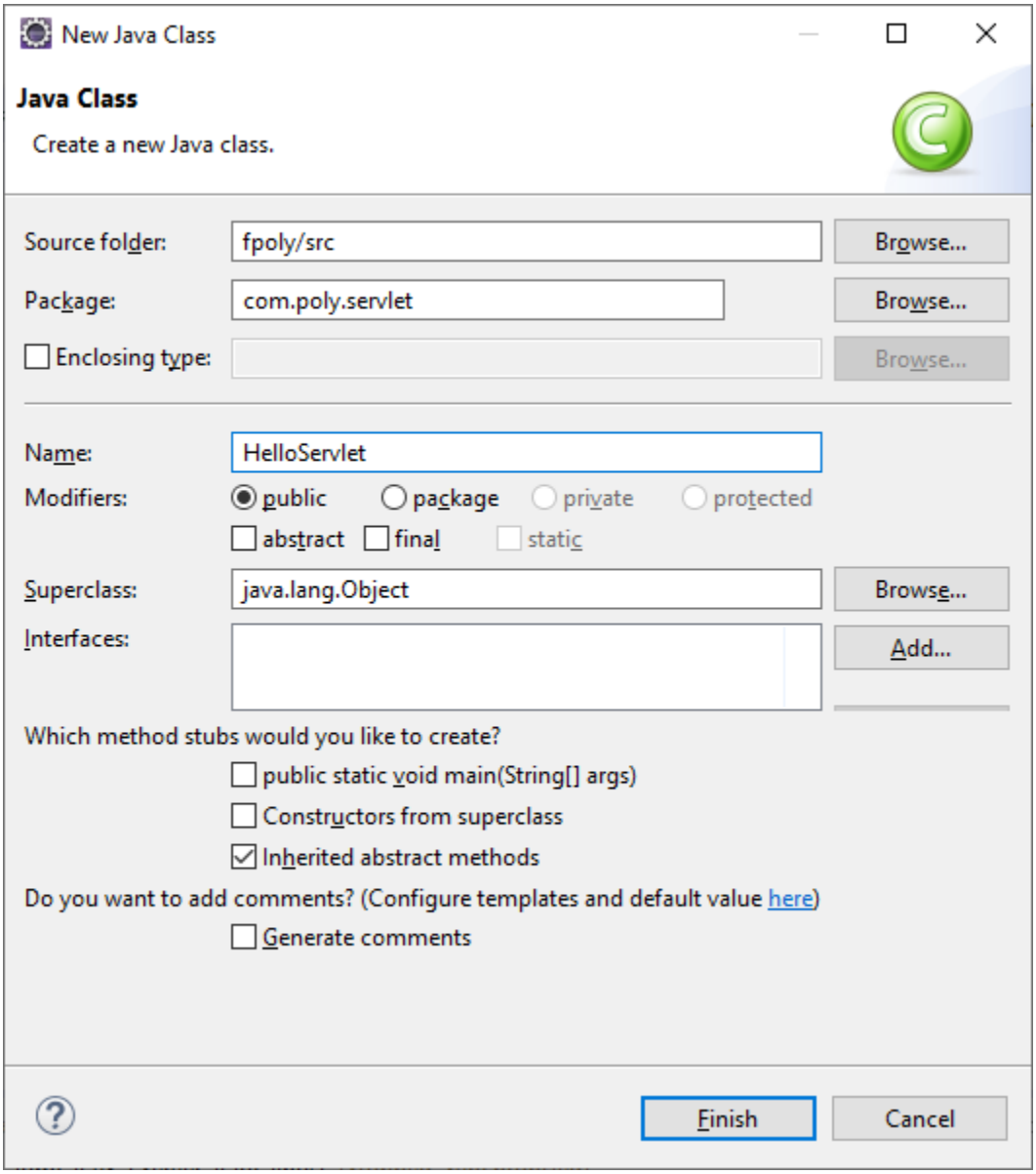


B2: Đặt tên cho dự án



B3: Tạo Servlet

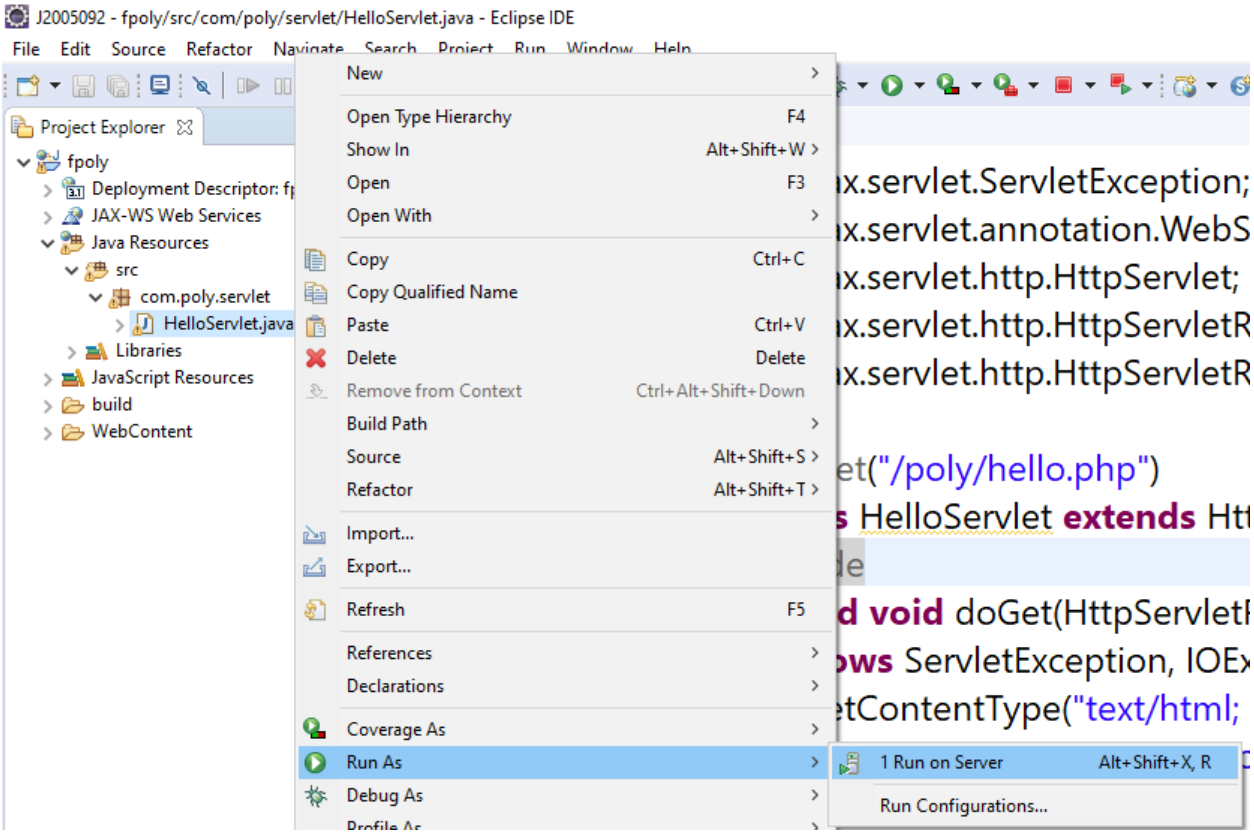




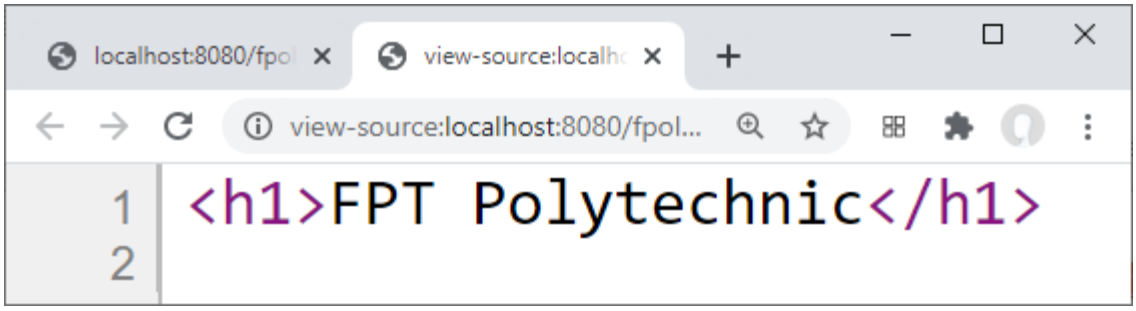
B4: Viết mã cho HelloServlet

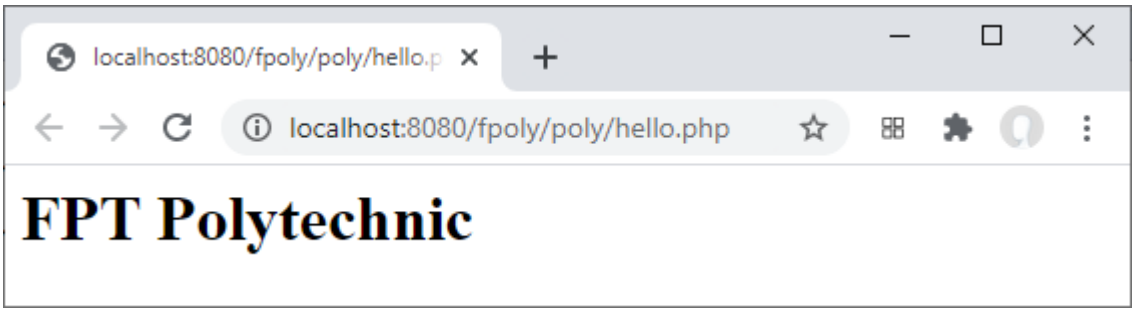
```
@WebServlet("/poly/hello.php")
public class HelloServlet extends HttpServlet{
	@Override
	protected void doGet(HttpServletRequest req, HttpServletResponse resp)
			throws ServletException, IOException {
		resp.getWriter().println("<h1>FPT Polytechnic</h1>");
	}
}
```

B5: Chạy Servlet



B6: Phân tích kết quả thực hiện

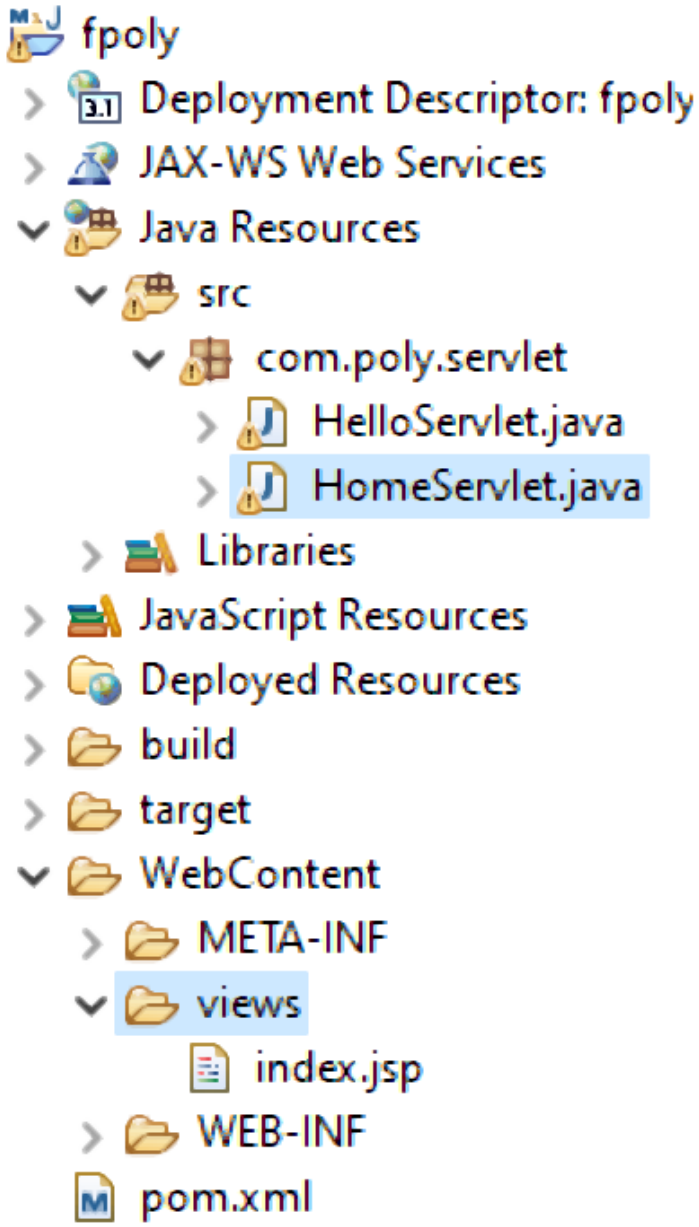




PHẦN II: TỔ CHỨC THEO MÔ HÌNH MVC

BÀI 3: TỔ CHỨC ỨNG DỤNG THEO MÔ HÌNH MVC

B1: Tạo HomeServlet.java và index.jsp đặt tại các thư mục như hình sau



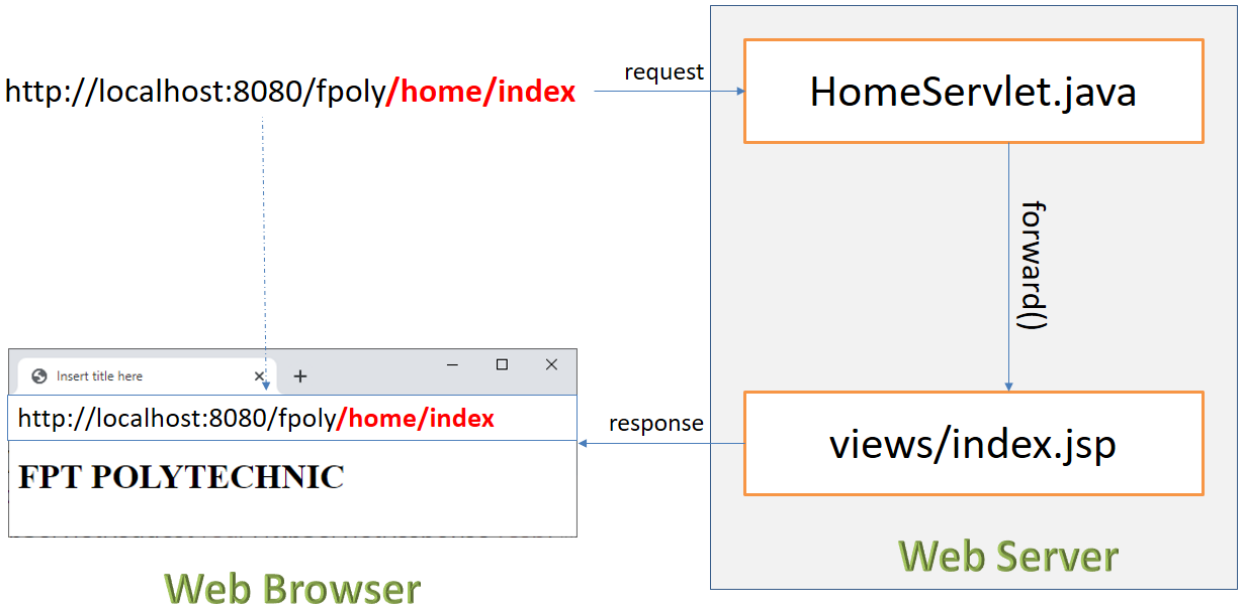
B2: Viết mã cho HomeServlet như sau

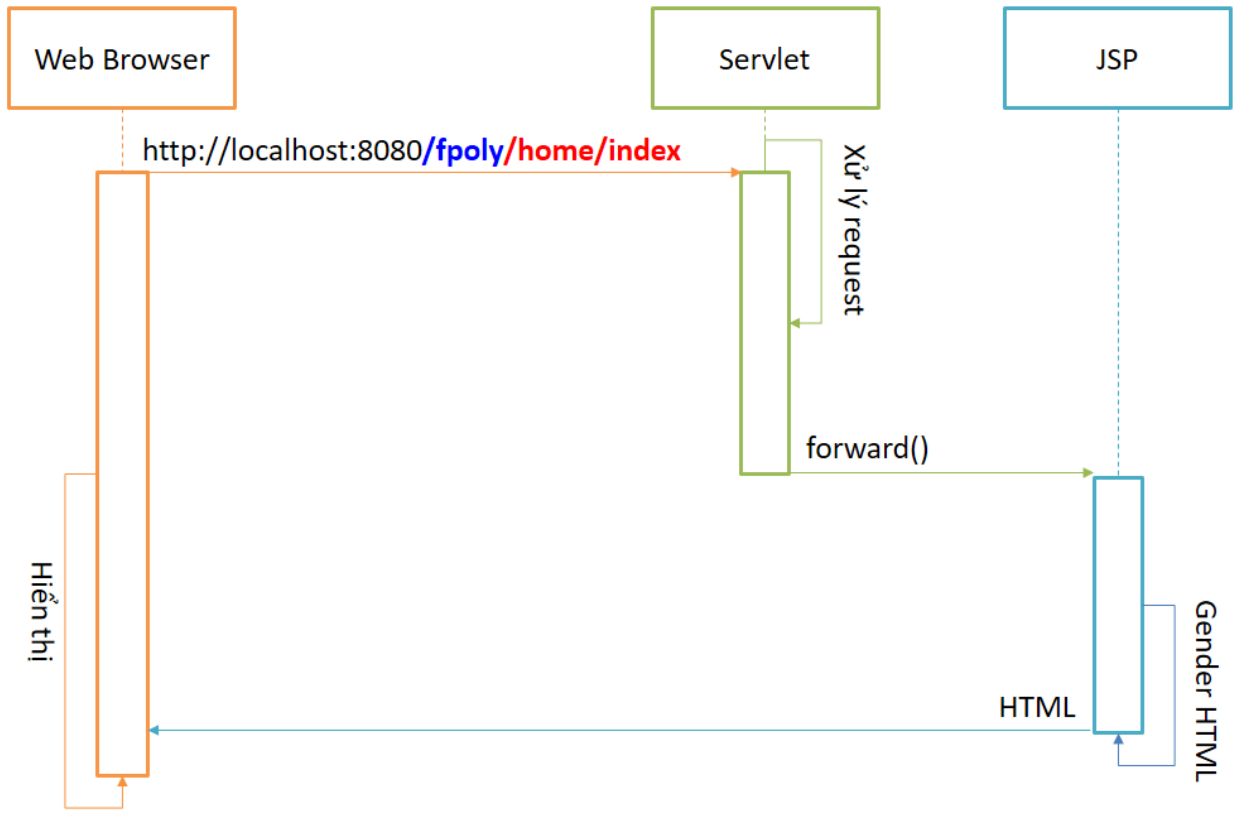
```java
package com.poly.servlet;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/home/index")
public class HomeServlet extends HttpServlet{
	@Override
	protected void doGet(HttpServletRequest req, HttpServletResponse resp)
			throws ServletException, IOException {
		req.getRequestDispatcher("/views/index.jsp").forward(req, resp);
	}
}
```

B3: Viết mã cho index.jsp

```jsp
<%@ page pageEncoding="utf-8"%>

<!DOCTYPE html>
<html>
<head>
	<meta charset="utf-8">
	<title>Insert title here</title>
</head>
<body>
	<h1>FPT POLYTECHNIC</h1>
</body>
</html>
```

B4: Chạy HomeServlet và giải thích quy trình thực hiện theo các sơ đồ sau





BÀI 4: TRUYỀN DỮ LIỆU TỪ SERVLET SANG JSP

4.1. Viết mã bổ sung vào phương thức doGet() của HomeServlet theo hướng dẫn sau để truyền attribute có tên là message cho trang index.jsp:

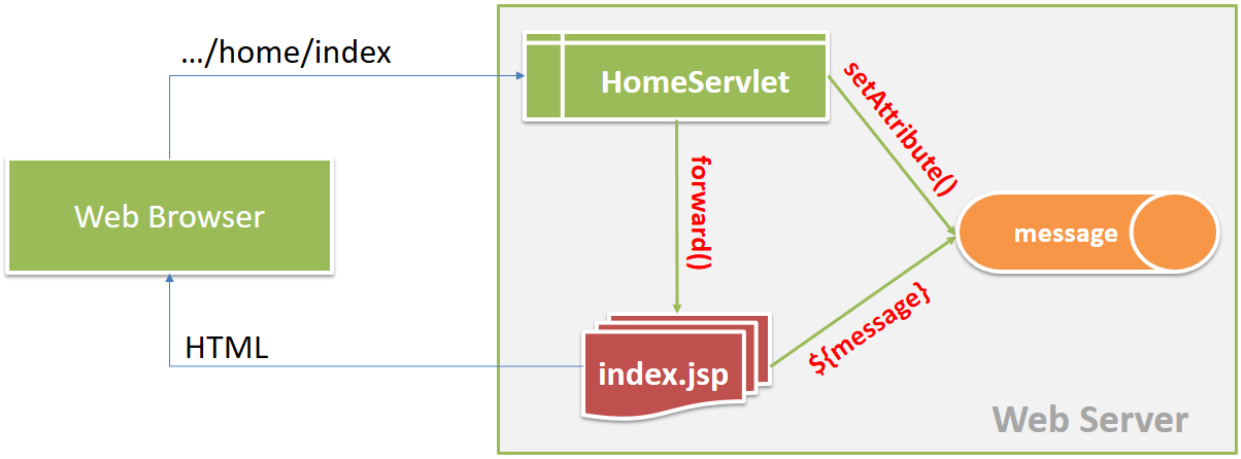
Thêm dòng lệnh ***req.setAttribute("message", "Chào thế giới Servlet/JSP!")*** vào trước dòng lệnh req.getRequestDispatcher(...).forward(...)

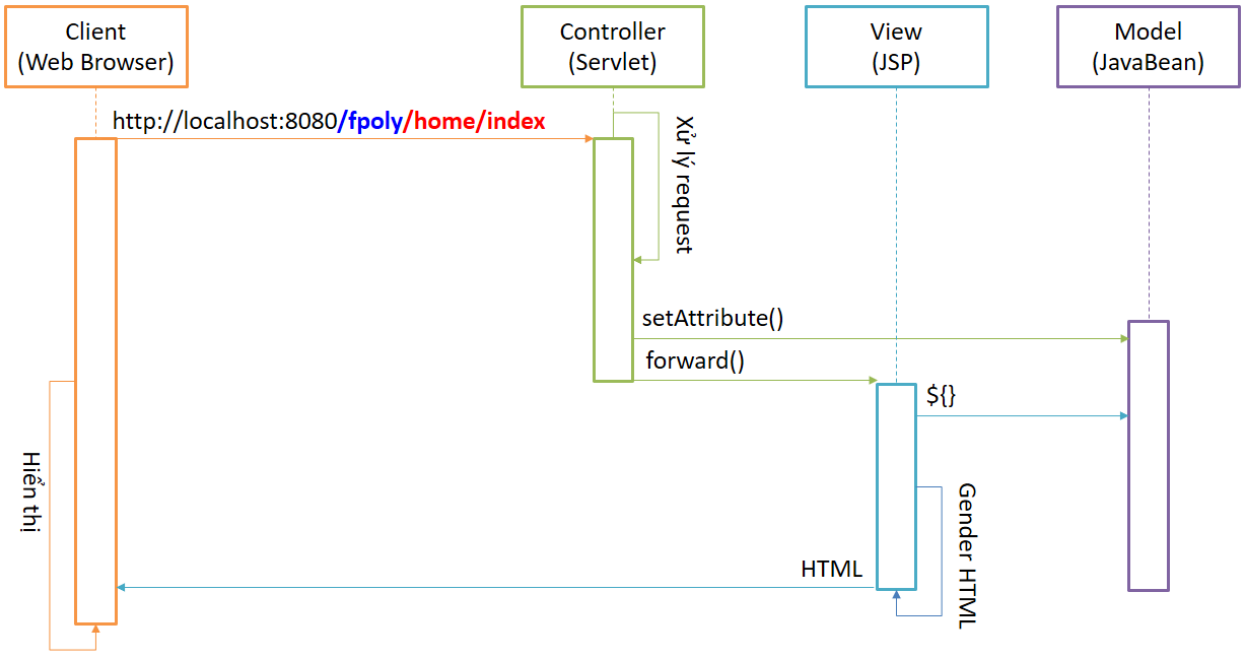
4.2. Viết mã bổ sung vào index.jsp theo hướng dẫn sau để lấy attribute message gửi từ servlet và hiển thị lên trang index.jsp

Thêm thẻ ***<h4>${message}</h4>*** vào sau thẻ <h1>FPT POLYTECHNIC</h1>

4.3 Chạy HomeServlet

4.4 Phân tích kết quả dựa trên 2 hình sau

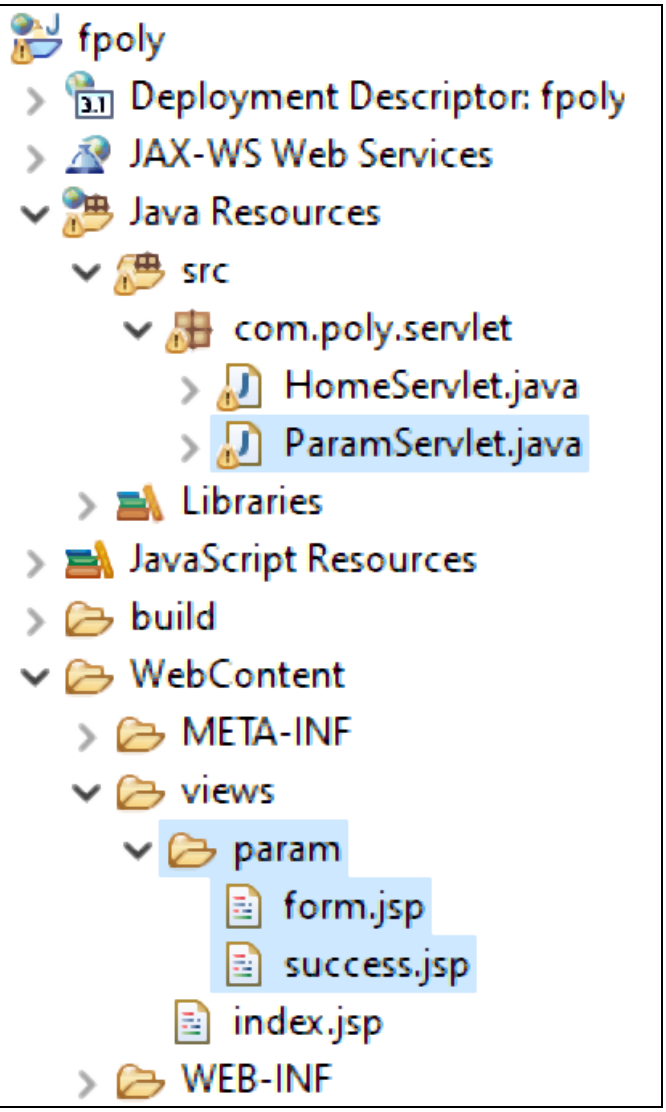



Client
(Web Browser)
Controller
(Servlet)
View
(JSP)
Model
(JavaBean)
http://localhost:8080/fpoly/home/index
Xử lý request
setAttribute()
forward()
${}
Hiển thị
Gender HTML
HTML

BÀI 5: XỬ LÝ THAM SỐ FORM

B1: Tạo các file ở các vị trí như hình bên:

- ParamServlet.java
- param/form.jsp
- param/success.jsp



B2: Viết mã cho ParamServlet.java như sau

```
package com.poly.servlet;

import java.io.IOException;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/param.php")
public class ParamServlet extends HttpServlet{
	@Override
	protected void doGet(HttpServletRequest req, HttpServletResponse resp)
```

```
                throws ServletException, IOException {
            req.getRequestDispatcher("/views/param/form.jsp").forward(req, resp);
        }

        @Override
        protected void doPost(HttpServletRequest req, HttpServletResponse resp)
                throws ServletException, IOException {
            String hoten = req.getParameter("hoten");
            req.setAttribute("message", "Xin chào " + hoten);
            req.getRequestDispatcher("/views/param/success.jsp").forward(req, resp);
        }
}
```

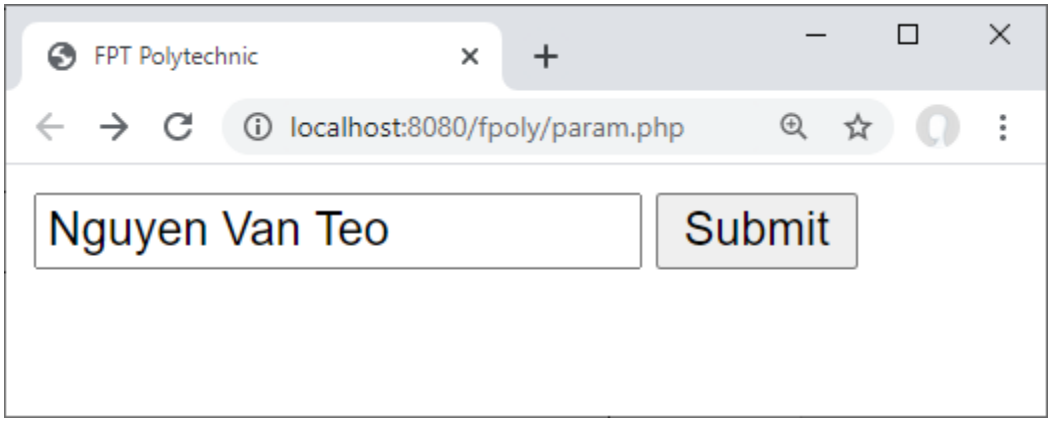
B3: Viết mã cho form.jsp

```
<%@ page pageEncoding="utf-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>FPT Polytechnic</title>
</head>
<body>
    <form action="/fpoly/param.php" method="post">
        <input name="hoten">
        <button>Submit</button>
    </form>
</body>
</html>
```

B4: Viết mã cho success.jsp

```
<%@ page pageEncoding="utf-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>FPT Polytechnic</title>
</head>
<body>
    <h1>Kết quả: ${message}</h1>
</body>
</html>
```

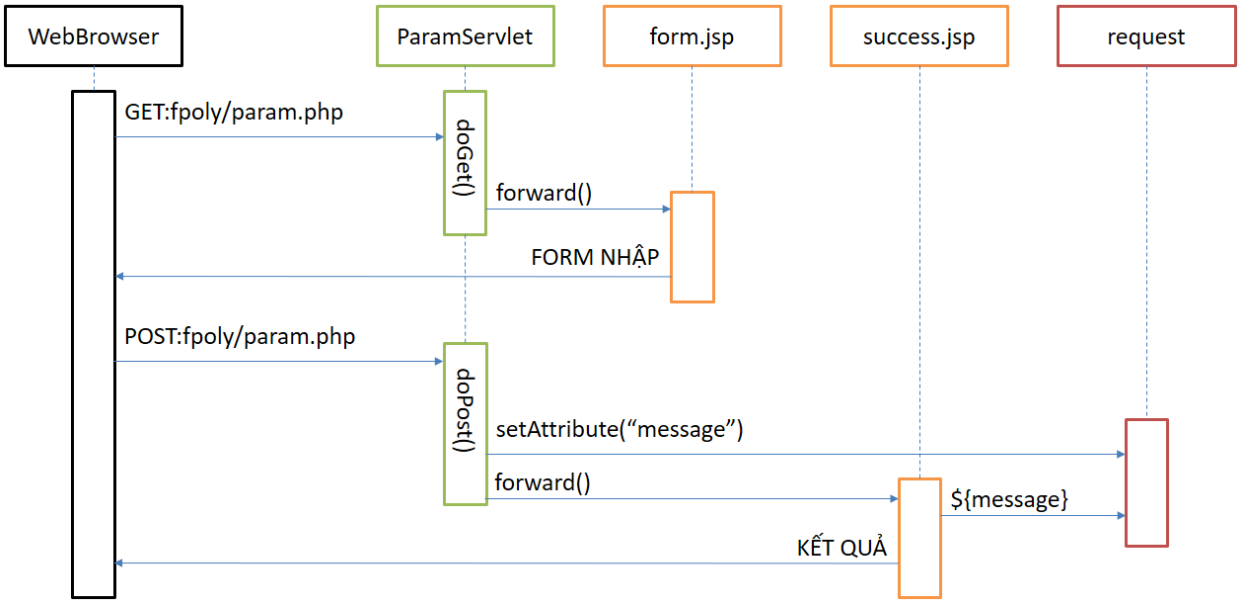
B5: Chạy ParamServlet



Hình 1: Mới chạy (doGet() -> form.jsp)

Hình 2: Nhấp nút Submit (doPost() -> success.jsp)

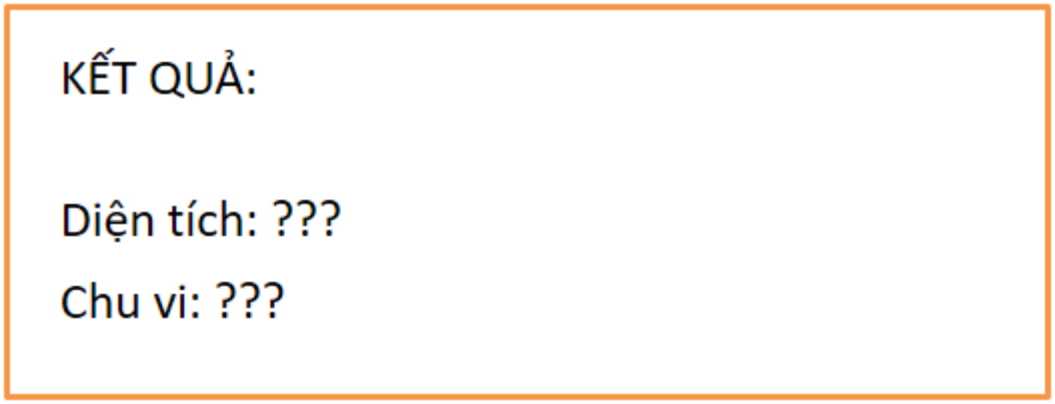
B6: Phân tích quy trình xử lý



BÀI 6: TÍNH DIỆN TÍCH VÀ CHU VI CỦA HÌNH CHỮ NHẬT

Xây dựng trang web cho phép nhập chiều rộng, chiều dài của một hình chữ nhật sau đó tính và xuất diện tích và chu vi ra trang kết quả.

Hình 1: Form nhập



Hình 2: Trang hiển thị kết quả

Hướng dẫn:

- Tạo Servlet tên là ChuNhatServlet, form-chu-nhat.jsp như (H1) và ket-qua.jsp như (H2) viết mã tương tự Bài 5
- Đặt tên cho 2 ô nhập của form là "dai" và "rong"
- Trong doPost() đọc và chuyển đổi giá trị các tham số "dai" và "rong" sang số thực:
  - String x = req.getParameter("dai");
  - double dai = Double.parseDouble(x)
- Áp dụng các công thức sau để tính diện tích và chu vi
  - diện tích = chiều dài x chiều rộng
  - chi vi = (chiều dài + chiều rộng) x 2
- Truyền diện tích và chu vi tính được sang ket-qua.jsp để hiển thị
  - req.setAttribute("s", diện tích);
- Hiển thị chu vi và diện tích trong ket-qua.jsp
  - Diện tích: ${s}