Course: Web Application Development
Lab Instructor: Dr.Nguyen Van Sinh

Email: nvsinh@hcmiu.edu.vn

Lab 3 - Introduction to Servlet Programming (14/3/2018)

Content:

- Introduction to Java Servlet Technology
- How to create a servlet and run on NetBeans IDE
- Practices and Exercises

Duration: 3 hours

Part 1: Introduction to Servlet and Servlet Life Cycle.

- **Recall: What is Servlet?**
  - A *servlet* is a Java programming language class that is used to extend the capabilities of servers that host applications access via a request-response programming model.
  - Although servlets can respond to any type of request, they are commonly used to extend the applications hosted by web servers. For such applications, Java Servlet technology defines HTTP-specific servlet classes.
- **How Java support Servlet Programming**
  - The javax.servlet and javax.servlet.http packages provide interfaces and classes for writing servlets. All servlets must implement the Servlet interface, which defines servlet life-cycle methods.
  - The **HttpServlet** class provides methods, such as **doGet** and **doPost**, for handling HTTP-specific services.
  - When implementing a generic service, you can use or extend the **GenericServlet** class provided with the Java Servlet API.
- **Tomcat Servlet/JSP container**
  - ***Tomcat*** can act as a stand-alone Web server and also as a servlet/JSP engine for other Web servers. When you download the Tomcat server, you really get a number of packages. **Catalina** and **Jasper** are the names of the servlet and JSP containers.
  - Tomcat by itself is a web server. This means that you can use Tomcat to service HTTP requests for servlets, as well as static files (HTML, image files, and so on). In practice, however, since it is faster for non-servlet, non-JSP requests, Tomcat normally is used as a module with another more robust web

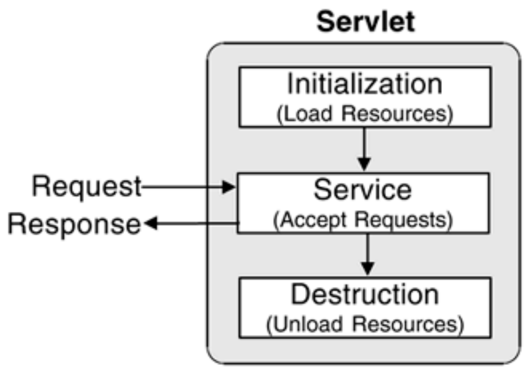
server, such as Apache web server or Microsoft Internet Information Server (IIS).

- Tomcat is not a J2EE application server. However, as J2EE app servers must themselves contain a servlet container to support the servlet/JSP APIs, J2EE app servers can embed Tomcat into their code to provide support for the Servlet and JSP APIs. One example of just such an application server is the popular open source JBoss J2EE app server (http://www.jboss.org/).

- **Servlet Life Cycle**
  - The javax.servlet.Servlet interface defines the methods that all servlets must implement and, among others, three methods that are known as **life-cycle methods**:

```
public void init(ServletConfig config) throws ServletException

public void service(ServletRequest req, ServletResponse res) throws ServletException, IOException

public void destroy()
```

  - These life-cycle methods are each called at separate times during the life span of a servlet, from the initial creation to the moment it's removed from service and destroyed. These methods are called in the following order:



1. When the servlet is constructed, it is initialized with the init() method.

2. Any requests from clients are handled initially by the service() method before delegating to the doXxx() methods in the case of an HttpServlet. The service() method is responsible for processing the request and returning the response.

3. When the servlet needs to be removed from service, it's destroyed with the destroy() method, then garbage collected and finalized. When the container decides to take a servlet out of service, it first ensures that any service() method calls have been completed.

  - The init method is called by the servlet container after the servlet class has been instantiated. The servlet container calls this method exactly once to indicate to the servlet that the servlet is being placed into service. The init method is important also because the servlet container passes a *ServletConfig* object, which contains the configuration values stated in the web.xml file for this application.

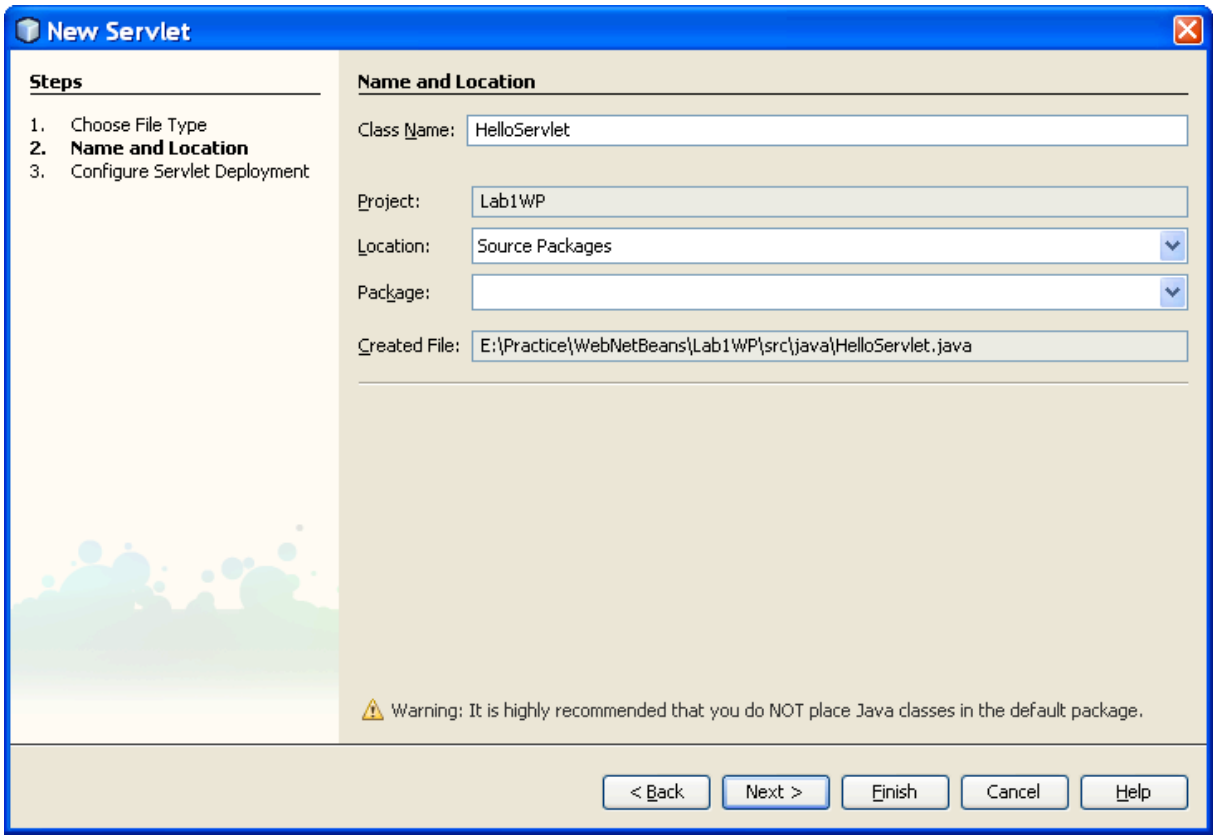
- The service method is called by the servlet container after the servlet's init method to allow the servlet to respond to a request. The servlet container passes a ServletRequest object and the ServletResponse object. The ServletRequest object contains the client's request and the ServletResponse contains the servlet's response.
- The servlet container calls the destroy method before removing a servlet instance from service. This normally happens when the servlet container is shut down or the servlet container needs some free memory.

*(More information: refer from textbook: **Core Servlets and Java Server Pages**)*

Part 2: How to create a Servlet on NetBeans IDE.

- **Default servlet creation**

After create a website (lab 1) on NetBeans IDE, right_click to the name of website -> New -> Servlet…


New Servlet

Steps
1. Choose File Type
2. **Name and Location**
3. Configure Servlet Deployment

Name and Location

Class Name: HelloServlet

Project: Lab1WP

Location: Source Packages

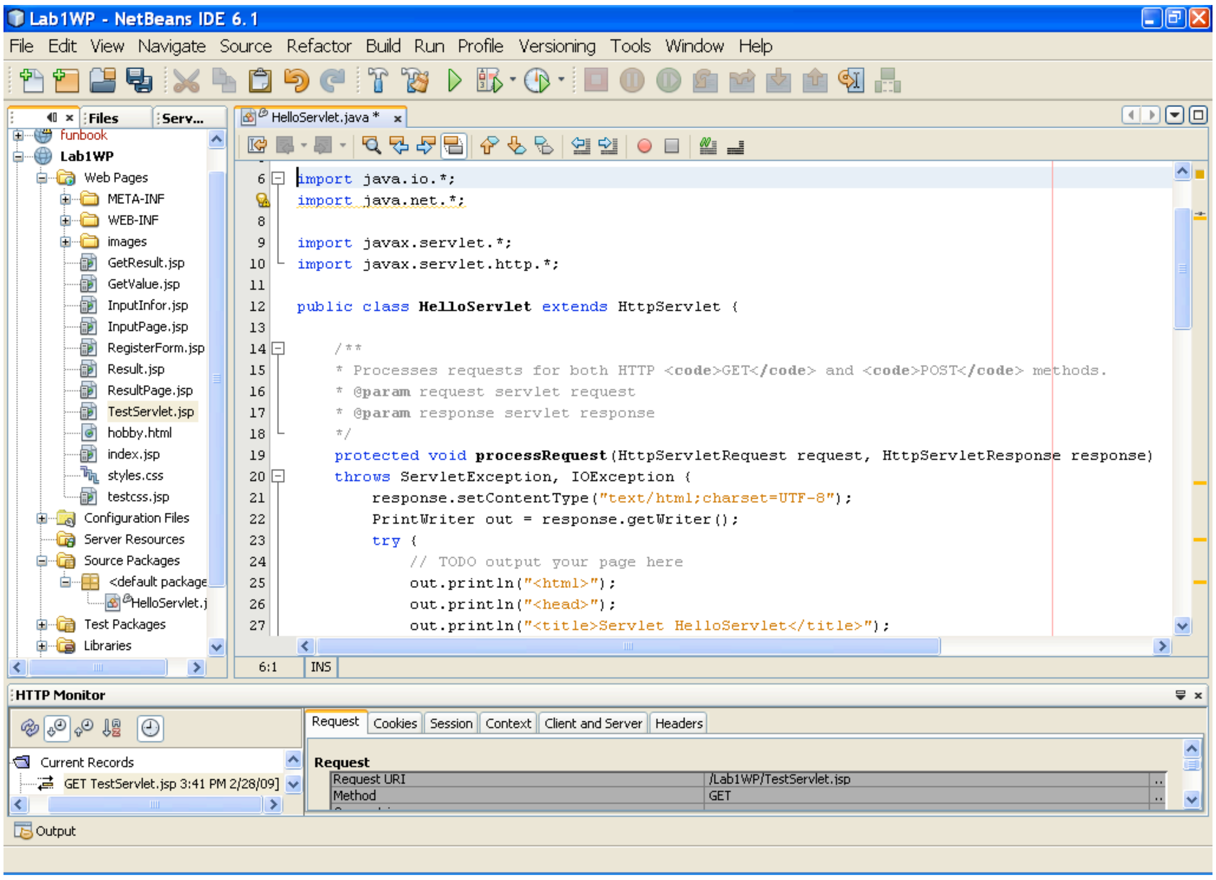
Package:

Created File: E:\Practice\WebNetBeans\Lab1WP\src\java\HelloServlet.java

Warning: It is highly recommended that you do NOT place Java classes in the default package.

Type a name and click next, then click Finish button, the source codes generate automatically in the file HelloServlet.java.

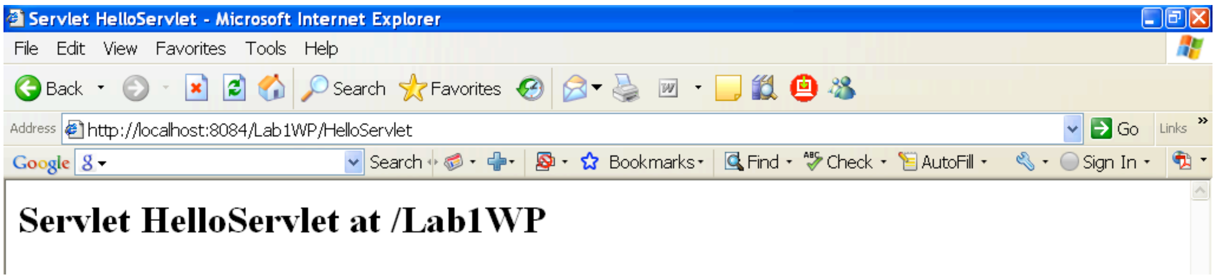
This is source code:



```
import java.io.*;
import java.net.*;

import javax.servlet.*;
import javax.servlet.http.*;

public class HelloServlet extends HttpServlet {

    /**
     * Processes requests for both HTTP <code>GET</code> and <code>POST</code> methods.
     * @param request servlet request
     * @param response servlet response
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try {
            // TODO output your page here
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet HelloServlet</title>");
```

Request	
Request URI	/Lab1WP/TestServlet.jsp
Method	GET

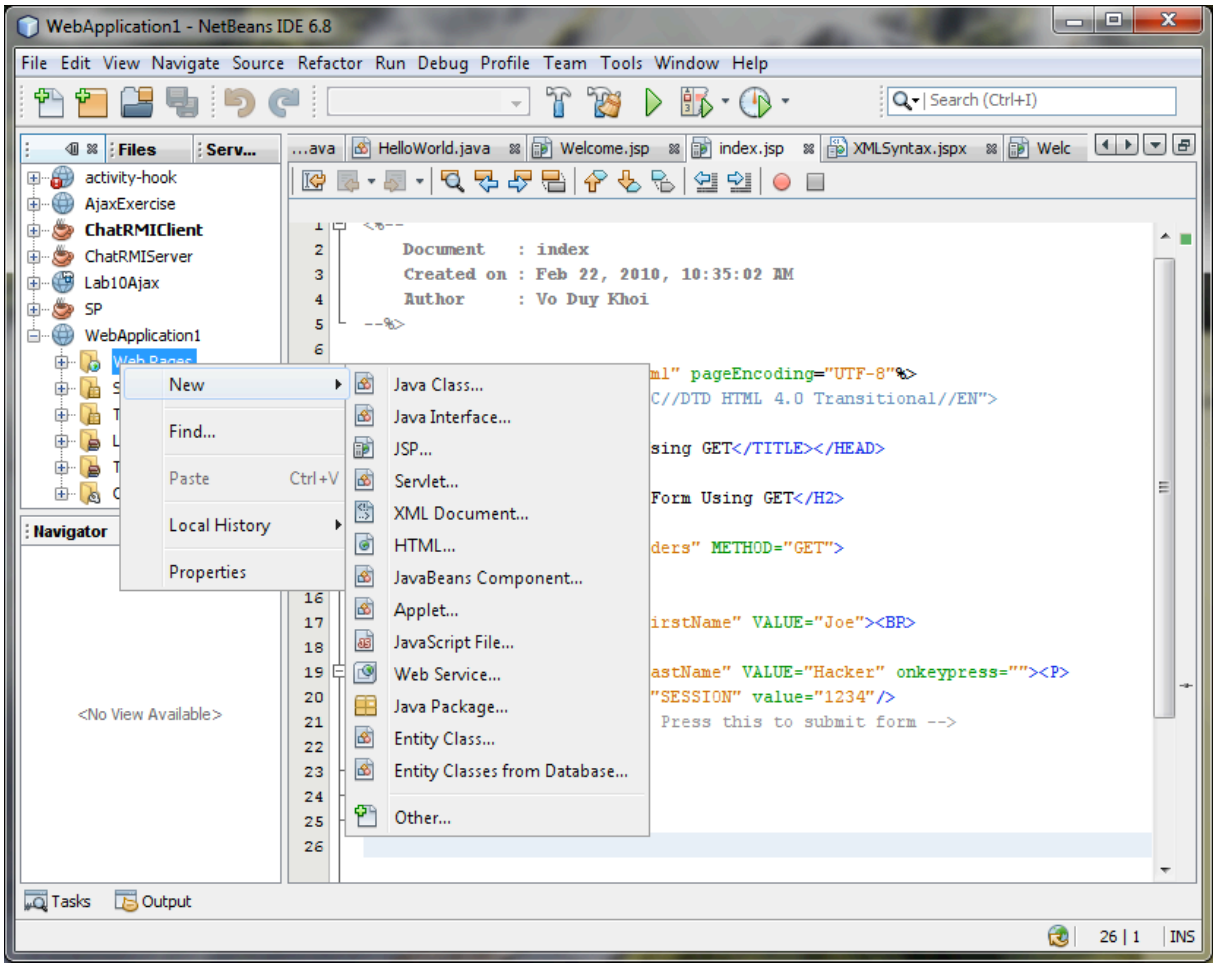
After building and running on the web server, this is the result page.


Servlet HelloServlet - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
http://localhost:8084/Lab1WP/HelloServlet
Go
Links
Google
Search
Bookmarks
Find
Check
AutoFill
Sign In

Servlet HelloServlet at /Lab1WP

- **Custom serlvet creation**

During application development, we might need to create more servlet. The steps to create new servlet in project are illustrated as figures:

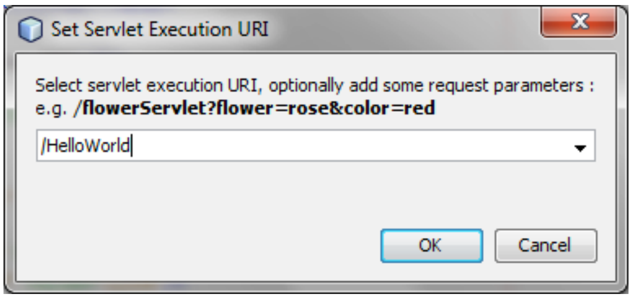


```
<%--
    Document   : index
    Created on : Feb 22, 2010, 10:35:02 AM
    Author     : Vo Duy Khoi
--%>

ml" pageEncoding="UTF-8"%>
C//DTD HTML 4.0 Transitional//EN">

sing GET</TITLE></HEAD>

Form Using GET</H2>

ders" METHOD="GET">

irstName" VALUE="Joe"><BR>

astName" VALUE="Hacker" onkeypress=""><P>
"SESSION" value="1234"/>
 Press this to submit form -->
```

Choose menu "Serlvet" to create new servlet. You should not choose normal "Java Class" then write class as servlet. If you do so, you have to configure to activate servlet manually.

- **How to run a servlet**

A servlet can be called by name of class without extension. For example, if your create a servlet class named "HelloWorld.class" and run it under context /Lab3 then your URL to run the servlet will be "/Lab3/HelloWorld" and one message will display as follows:

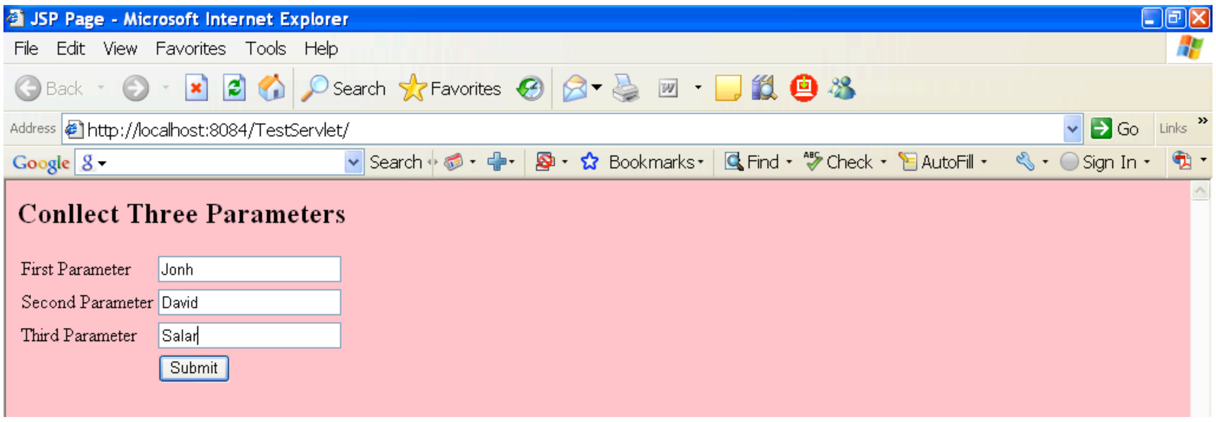


Select servlet execution URI, optionally add some request parameters :
e.g. **/flowerServlet?flower=rose&color=red**

/HelloWorld

Part 3: Practices and Exercises.

Exercise1: design a form (ThreeParams.JSP) as below:

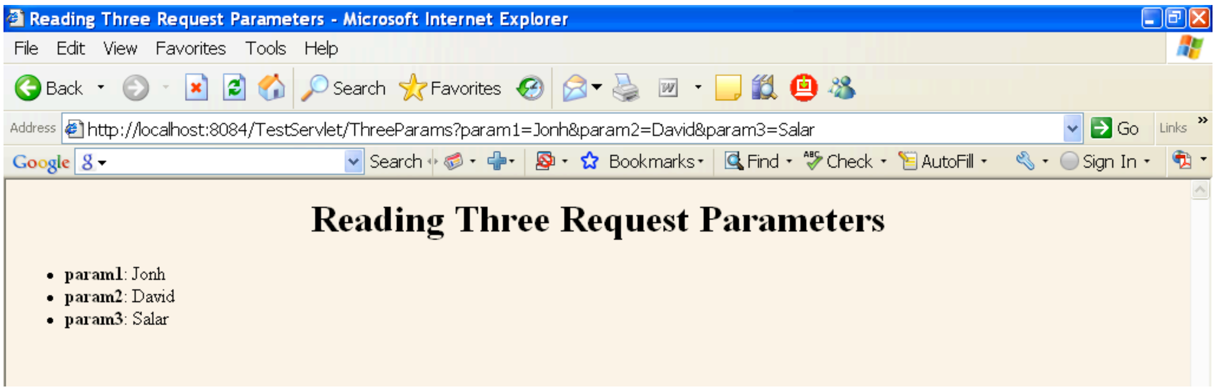


JSP Page - Microsoft Internet Explorer
http://localhost:8084/TestServlet/

Conllect Three Parameters

First Parameter	Jonh
Second Parameter	David
Third Parameter	Salar
	Submit

After input three values in the textbox -> click Submit button, it call result page GetThreeParam.class (this file is built from GetThreeParam.java)

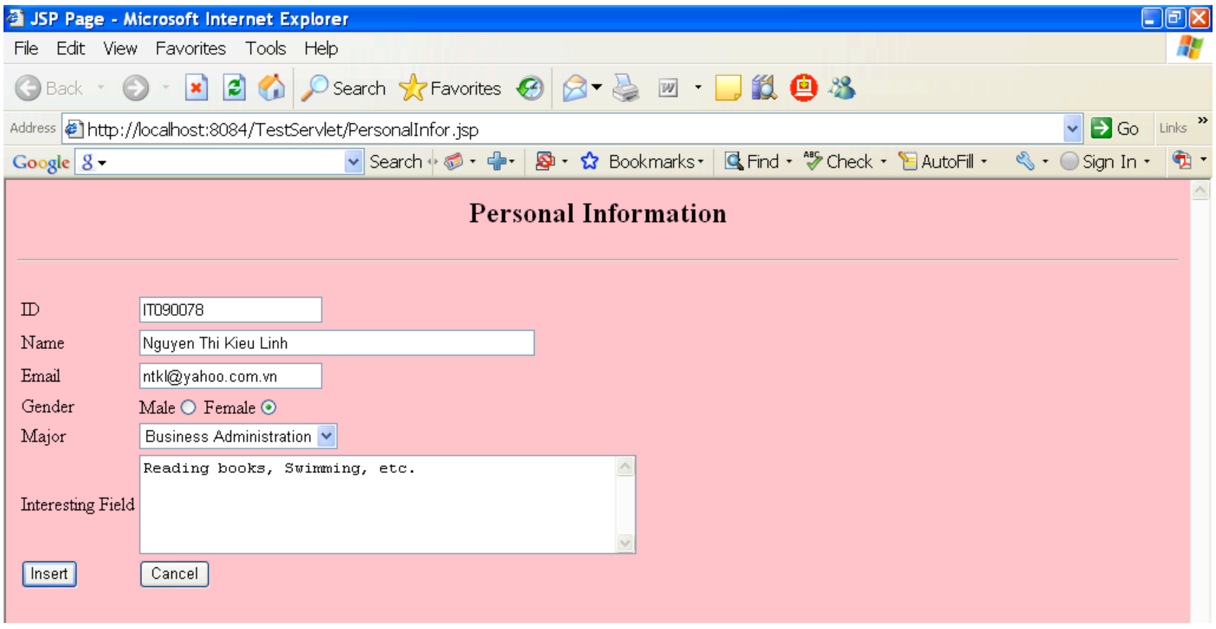


Reading Three Request Parameters - Microsoft Internet Explorer
http://localhost:8084/TestServlet/ThreeParams?param1=Jonh&param2=David&param3=Salar

Reading Three Request Parameters

- **param1**: Jonh
- **param2**: David
- **param3**: Salar

Exercise 2: Design form Personal Information (PersonalInfor.jsp)

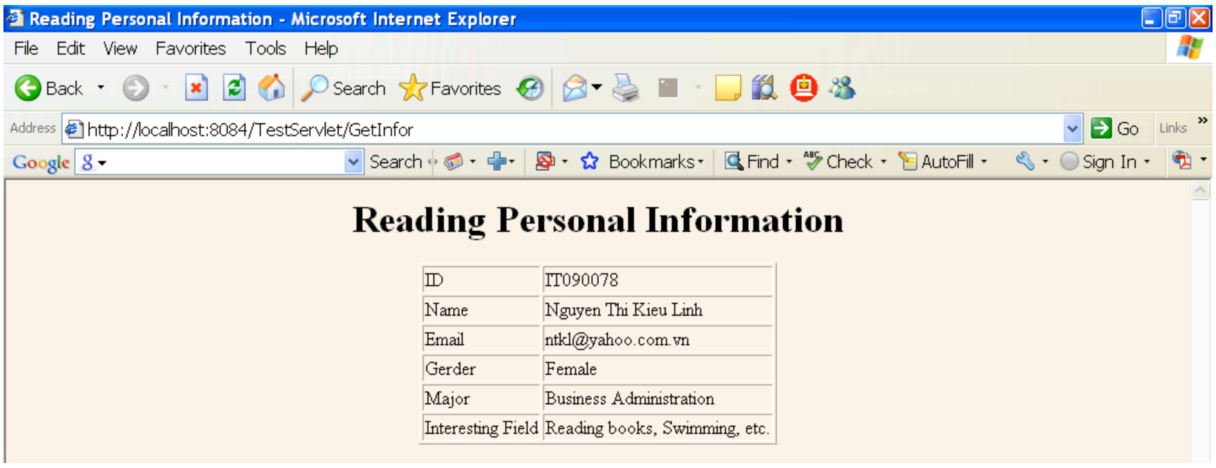


JSP Page - Microsoft Internet Explorer
http://localhost:8084/TestServlet/PersonalInfor.jsp

Personal Information

ID	IT090078
Name	Nguyen Thi Kieu Linh
Email	ntkl@yahoo.com.vn
Gender	Male ○ [✓ Female]
Major	Business Administration
Interesting Field	Reading books, Swimming, etc.
Insert	Cancel

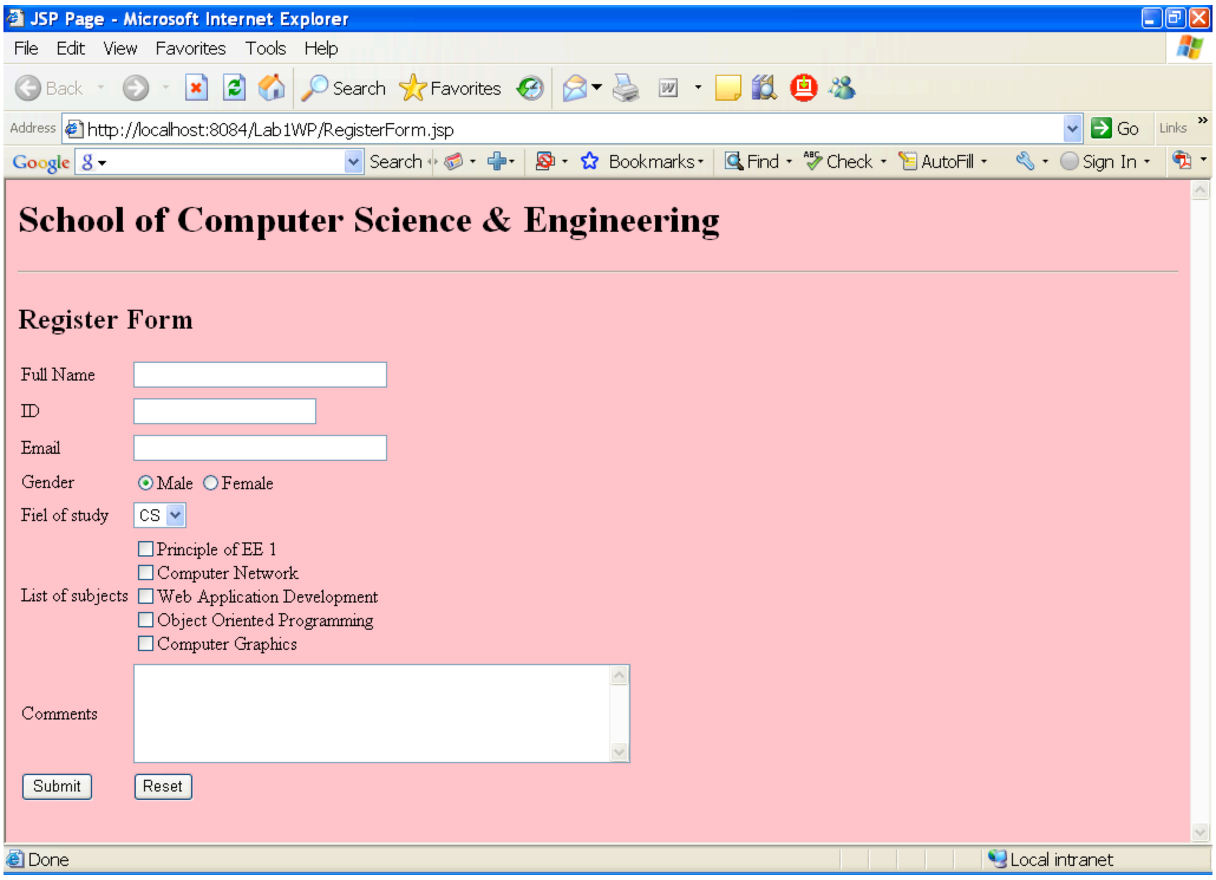
The result from GetInfor.java


Reading Personal Information - Microsoft Internet Explorer
http://localhost:8084/TestServlet/GetInfor

Reading Personal Information

ID	IT090078
Name	Nguyen Thi Kieu Linh
Email	ntkl@yahoo.com.vn
Gerder	Female
Major	Business Administration
Interesting Field	Reading books, Swimming, etc.

Exercise 3: Design a register form, use Servlet to get all information when user click the Submit button and put into a new page.


JSP Page - Microsoft Internet Explorer
http://localhost:8084/Lab1WP/RegisterForm.jsp

School of Computer Science & Engineering

Register Form

Full Name	
ID	
Email	
Gender	[✓ Male] Female
Fiel of study	CS
List of subjects	☐ Principle of EE 1 ☐ Computer Network ☐ Web Application Development ☐ Object Oriented Programming ☐ Computer Graphics
Comments	
Submit	Reset

The result show parameter Servlet (RegisterCourse.java)

Exercise 4 (bonus): You are required to modify the servlet in **Exercise 3**. When a user clicks on Submit button, all information will be sent to your email. You can use any technology, service or third party library to support the development of email function .