NAME: SATYAJIT GHANA REG NO: 17ETCS002159

Laboratory 6

Title of the Laboratory Exercise: To implement Server side web component (Servlet) to handle client request

1. Introduction and Purpose of Experiment

   Students will learn to implement servlet to handle client submitted request

2. Aim and Objectives

   Aim

   Objectives

2. Experimental Procedure
3. Calculations/Computations/Algorithms

LoginServlet.jsp

```
package webarch.servlets;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.*;
import javax.servlet.http.*;

public class LoginServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        RequestDispatcher rd = request.getRequestDispatcher("login.jsp");
        rd.include(request, response);
    }
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        String username = request.getParameter("username");
        String password = request.getParameter("password");

        response.setContentType("text/html");
        PrintWriter pw = response.getWriter();
        pw.println("<script type=\"text/javascript\">");
```

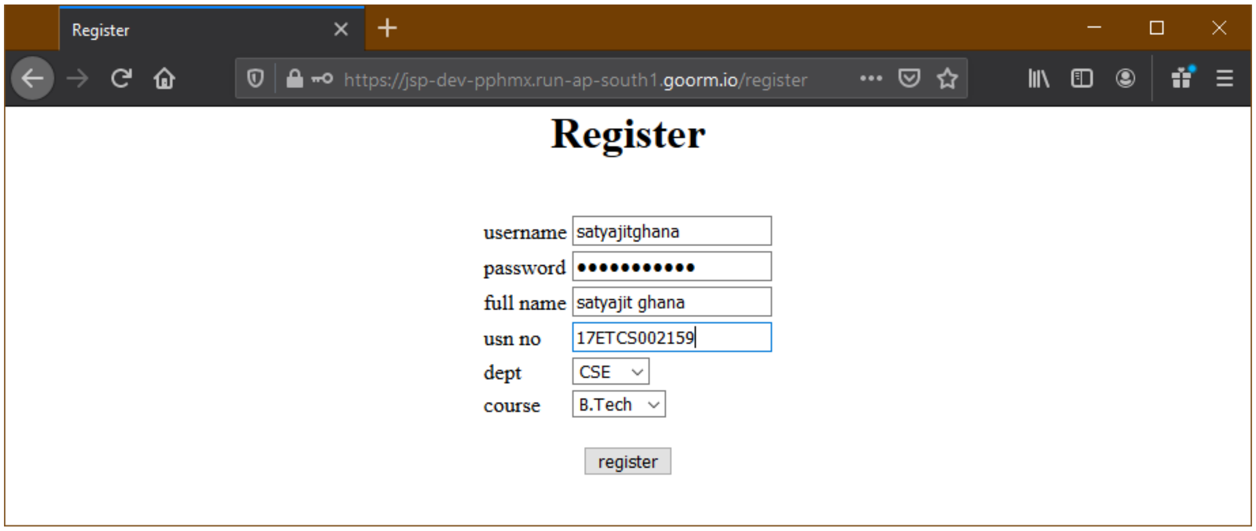
```java
        pw.println("alert('Entered Values: username [ "+username+ " ], password [ "+password+
" ] ');");
        pw.println("</script>");

        // if login success then redirect to projects page
        // RequestDispatcher rd = request.getRequestDispatcher("projects.jsp");

        RequestDispatcher rd = request.getRequestDispatcher("login.jsp");
        rd.include(request, response);
    }
}
```

RegisterServlet

```java
package webarch.servlets;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.*;
import javax.servlet.http.*;

public class RegisterServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        RequestDispatcher rd = request.getRequestDispatcher("register.jsp");
        rd.include(request, response);
    }
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        String username = request.getParameter("username");
        String password = request.getParameter("password");
        String fullname = request.getParameter("fullname");
        String usnno = request.getParameter("usnno");
        String dept = request.getParameter("dept");
        String course = request.getParameter("course");

        response.setContentType("text/html");
        PrintWriter pw = response.getWriter();
        pw.println("<script type=\"text/javascript\">");
        pw.println("alert('Entered Values: username [ "+username+ "], password [ "+password+"
], fullname ["+fullname+"], usnno ["+usnno+"], dept ["+dept+"], course ["+course+"] ');");
        pw.println("</script>");

        // if register success then send to login page
        // RequestDispatcher rd = request.getRequestDispatcher("login.jsp");
        RequestDispatcher rd = request.getRequestDispatcher("register.jsp");
        rd.include(request, response);
    }
}
```

4. Presentation of Results

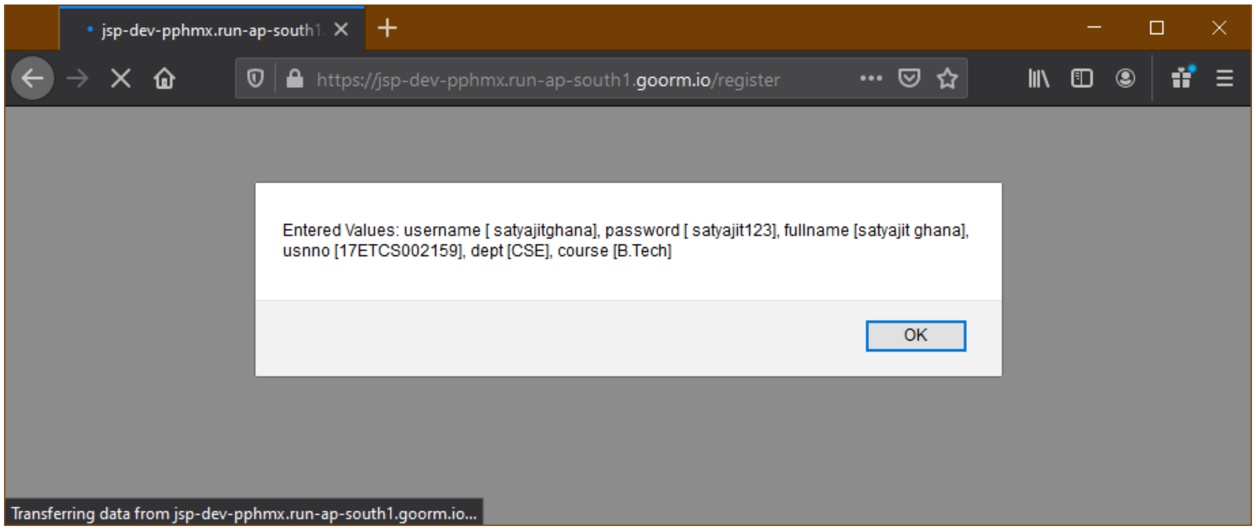
Filling up the Register Form



Register

username	satyajitghana
password	●●●●●●●●●●●
full name	satyajit ghana
usn no	17ETCS002159
dept	CSE
course	B.Tech

Entered details

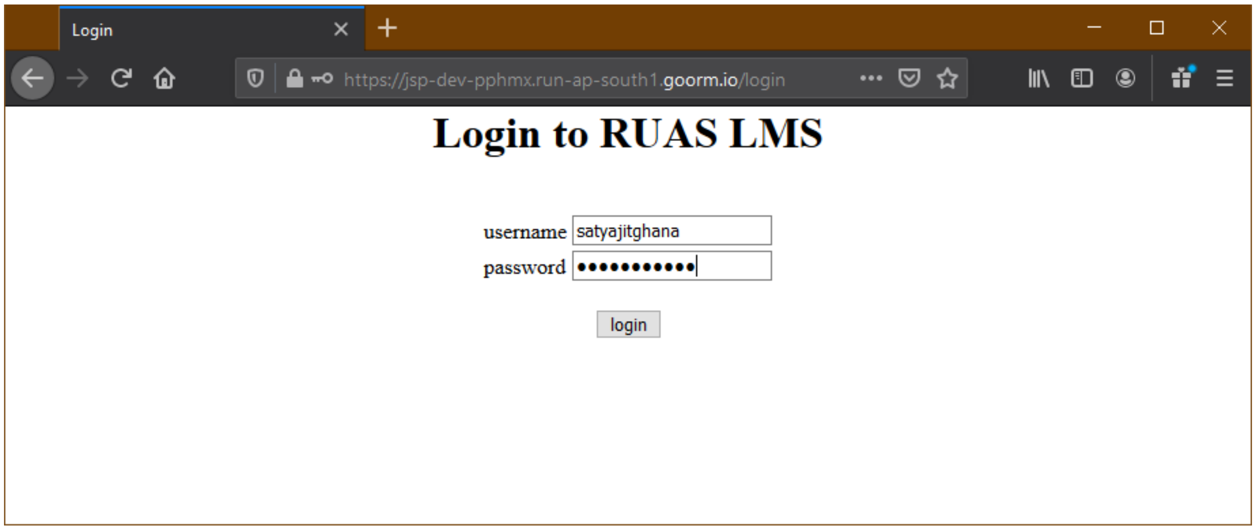


jsp-dev-pphmx.run-ap-south1
https://jsp-dev-pphmx.run-ap-south1.goorm.io/register
OK

Entered Values: username [satyajitghana], password [satyajit123], fullname [satyajit ghana], usnno [17ETCS002159], dept [CSE], course [B.Tech]

Transferring data from jsp-dev-pphmx.run-ap-south1.goorm.io...

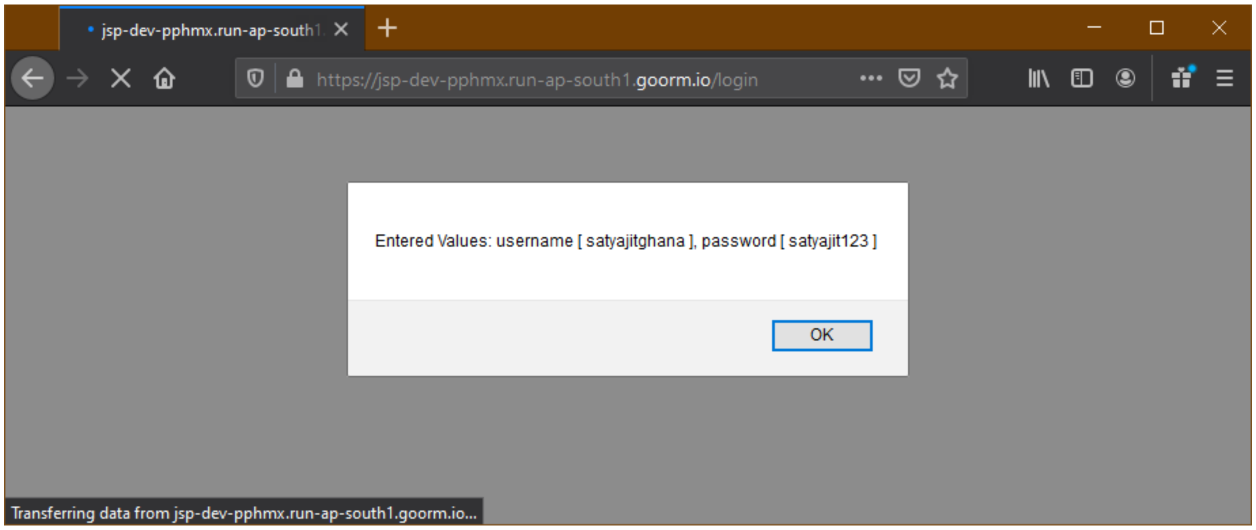
Filling up the Login Form



Login to RUAS LMS

username	satyajitghana
password	•••••••••••

Entered Details

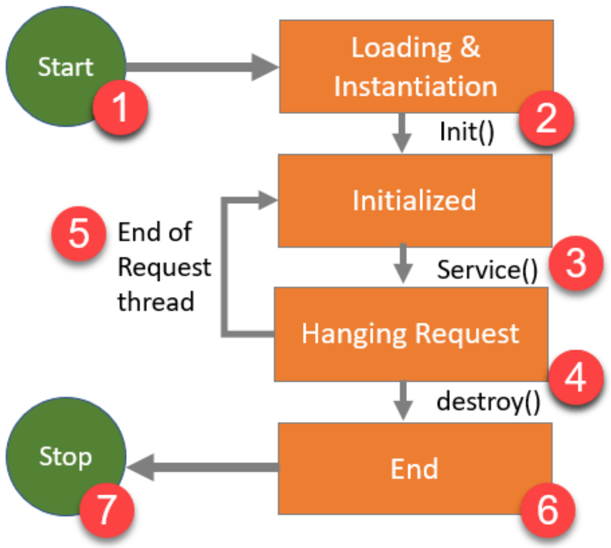


jsp-dev-pphmx.run-ap-south1.
https://jsp-dev-pphmx.run-ap-south1.goorm.io/login

Entered Values: username [satyajitghana], password [satyajit123]

Transferring data from jsp-dev-pphmx.run-ap-south1.goorm.io...

5. Analysis and Discussions



1) Start: Execution of servlet begins.

2) Loading & instantiation void init(): It is called when servlet is first loaded. This method lets you initialize servlet.

3) Initialized void service(): The purpose of this method is to serve a request. You can call it as many times as you like.

4) Handling request and destroying servlet: Java application must be first determined what code is needed to execute the request URL to provide a response. To destroy servlet Void destroy method is used at the end of servlet life cycle.

5) End of Request Thread: When service() finishes its task, either the thread ends or returns to the thread pool that is managed by servlet container.

6) End: Servlet lifecycle finishes.

7): Stop: Servlet stop executing.

6. Conclusions

Servlet Advantage

1. Servlets provide a way to generate dynamic documents that is both easier to write and faster to run.
2. provide all the powerfull features of JAVA, such as Exception handling and garbage collection.
3. Servlet enables easy portability across Web Servers.
4. Servlet can communicate with different servlet and servers.
5. Since all web applications are stateless protocol, servlet uses its own API to maintain session

Servlet Disadvantage

1. Designing in servlet is difficult and slows down the application.
2. Writing complex business logic makes the application difficult to understand.
3. You need a Java Runtime Environment on the server to run servlets. CGI is a completely language independent protocol, so you can write CGIs in whatever languages you have available (including Java if you want to).

7. Comments

a. Limitations of Experiments

It is hard to trace JSP pages error because JSP pages are translated to servlet. As JSP output is HTML, it is not rich in features. It is very hard to debug or trace errors because JSP pages are first translated into servlets before the compilation process. JSP pages require more disk space and time to hold JSP pages as they are compiled on the server.

b. Limitations of Results

None

c. Learning happened

We learnt how to connect a servlet to a JSP page, and use the request response model

d. Recommendations

None

Component	Max Marks	Marks Obtained
Viva	6	
Results	7	
Documentation	7	
Total	20	