**SERVLET**

* Overview of client-server arch /environment
* What is web application
* Deployment descriptor
* web.xml

*LAB1- create a web application, write web.xml, welcome file; create a welcome.html and now run the application on server (tomcat). Requirement is that the welcome.html file comes on web browser on writing the url*

AIM of this assignment is to understand web application, URL (<http://localhost:8080/webAppName>)

* Brief on Tomcat (web server)
* URL details like – Localhost, 8080 etc

--------------------------------------------------------------------------------------------------------------------------------------

* Servlet overview
* HTTP protocol overview
* Servlet life cycle
* Servlet class hierarchy
* Mapping of request to servlet class
* Mapping in web.xml

*LAB 2 – now on the welcome.html page – create an input type field and submit button; map this request with java servlet, check if code inside doGet or doPost is executing;*

* Development environment
* Deployment environment

--------------------------------------------------------------------------------------------------------------------------------------

* Web/Servlet container
* HTTP servlet request, HTTP servlet response

*LAB 3 – now on the welcome.html page – create an input type field and submit button; now user will enter a name in input field and on submit, the system will receive this name and print it on console*

AIM of this assignment – is to understand how to map request with a servlet class; servlet-mapping in web.xml; how code flow from HTML page to servlet class and in which method (GET and POST); how to receive data from HTML page (use of request.getParameter)

* Get and Post

*LAB 4 – execute the LAB 3, for both get and post, observe the URL and try to understand difference between get and post*

--------------------------------------------------------------------------------------------------------------------------------------

* HTTP servlet response
* Response type –
* HTML code on Java

*LAB 5 – now modify the LAB3 and now system will check if name entered by user on welcome.html page is present in DB (create a DB with one table name user with 4 fields – user ID, name, technology, password; also put some dummy data) . Based on condition that username is valid or not – show the appropriate response on a HTML page (this HTML page will be java/servlet generated)*

response.setContentType("text/html");

PrintWriter out = response.getWriter();

out.println("<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.0 " +

"Transitional//EN\">\n" +

"<HTML>\n" +

"<HEAD><TITLE>Hello WWW</TITLE></HEAD>\n" +

"<BODY>\n" +

"<H1>Hello WWW</H1>\n" +

"</BODY></HTML>");

AIM of this assignment – is to understand how to use response, print-Writer etc. and how to create HTML page from Java?

--------------------------------------------------------------------------------------------------------------------------------------

* Send Redirect

*LAB 6 – modify lab 5, now instead of creating HTML code; just redirect the response to right.html or wrong.html page accordingly.*

AIM of this assignment – is to use send redirect

--------------------------------------------------------------------------------------------------------------------------------------

* Request dispatcher
* Forward

Difference between forward and sendRedirect

LAB 7 – from HTML page having only one field, call a servlet (say servlet 1)– in servlet print this name (using request.getParameter) on console and then with the help of sendRedirect call another servlet (say servlet 2), print the name (using request.getParameter) on console.

Now go from Servlet 1 to servlet 2 using forward

Now, add an attribute in request object in servlet 1 and then in servlet 2 retrieve it and print

*LAB 8 - Adding more feature to LAB 1 – now system will check if name entered by user on welcome.html page is present in DB (create a DB with one table name user with 3 fields – user ID, name, technology also put some dummy data) . If user found system will show a message “Hi <user name> U belongs to <TECH NAME> technology” on web browser. Here technology name will be an anchor tag and on click on this link, system will show all other users belonging to this technology on next web page*

----------------------------------------------------------------------------------------------------------------------------------

* Servlet config
* Init param

LAB 9 –in LAB 8 for database connection receive essential info (like URL, user Id and password) from servlet config as an init param

----------------------------------------------------------------------------------------------------------------------------------

* Servlet context
* Context param

LAB 10 –re do the LAB 9 by using context param - try to understand difference between context & config

------------------------------------------------------------------------------------------------------------------------

* **HTTP is stateless**
* **Session –**

Session manage by

* Cookies
* URL rewriting

Creating session using-

**HttpSession session = request.getSession();**

LAB 11 - From HTML page (welcome page) call a servlet – create a session and print session id on console and this servlet will redirect to second HTML. From second HTML call second servlet. Create session object in second servlet as well and print session id on console.

Run the application from STS web browser, on IE / chrome and other browere. Open multiple windows in respective browsers and run it, and check how it is working.

**IN FIRST SERVLET**

HttpSession session = request.getSession();

System.*out*.println("Session Id : " + session.getId());

response.sendRedirect("dusra.html");

**IN SECOND SERVLET**

HttpSession session = request.getSession();

System.*out*.println("Session Id in second : " + session.getId());

To check if session is new

HttpSession session = request.getSession();

**if** (session.isNew()){

System.*out*.println("new session");

} **else** {

System.*out*.println("old session ");

}

LAB 12 – Modify LAB 11, now check if session is new or not on both first and second servlet

If always wanted **to use existing session**

HttpSession session = request.getSession(**false**);

// give only existing session, if no session exist it return null

**if** (session == **null**){

response.sendRedirect("dusra.html");

}

In web-app

<session-config>

<session-timeout>5</session-timeout>

</session-config>

LAB 13 – First HTML will redirect to FirstServlet where it will create a session and sendredirect to second HTML. On second HTML there will be a link which takes to Second servlet – in second servlet use existing session to print session id.

After navigating from first HTML to second HTML, wait for little longer time then mentioned session time out and then click on link. If session is invalidated, then redirect to first page else print session id on console.

Here we are checking if session is invalidated automatically also we are creating a scenario of using existing session only

**LOGOUT**

session.invalidate()

LAB 14 – close a session

----------------------------------------------------------------------------------------------------------------------------------

* Listener

LAB 15 – use of listener (Servlet-Context-Listener) to create DB connection and store connection/statement in context attribute

**Web xml**

<listener>

<listener-class>

**In Listener class**

Public Class MyListner implements **ServletContextListener**

contextInitalized(ServletContextEvent ev){

ServletConetxt context = ev.getServletContext();

//

context.setAttribute(“key”, value);

}

contextDestroyed(ServletContextEvent ev){

}

**Attribute is not thread safe – so careful while setting anything in attribute**

----------------------------------------------------------------------------------------------------------------------------------

* **Difference between parameter and attribute**

|  |  |  |
| --- | --- | --- |
|  | ATTRIBUTE | PARAMETER |
|  | setAttribute | No method like setParamater |
| Return type | Object (so casting is must) | String |
| Available with | Application/context  Request  Session | Application/context  Request  Servlet init (nothing like session parameter) |

----------------------------------------------------------------------------------------------------------------------------------

* **Attribute scope - session, context, request**

Context – everyone in application

Session – to those with access to specific Http Session

Request – only those with access of specific Http Request

**Methods**

getAttribute(key)

setAttribute(key, value)

removeAttribute(key)

getAttributeNames()

LAB 16 – in LAB 7 – now set some attributes in request of servlet 1 then in servlet 2 (which is getting called from servlet 1 using request dispatcher (forward)) perform above mentioned 4 methods of attribute

----------------------------------------------------------------------------------------------------------------------------------

**Thread Safe servlet**

Never synchronize service method

SingleThreadModel

----------------------------------------------------------------------------------------------------------------------------------

**Deploying a web application**

WAR file

Deploying on tomcat 6

Setting up environment variable (JAVA\_HOME and CATELINA\_HOME)

Running tomcat

Revisiting deployed war file

Running the application

LAB 17 – create a WAR file and deploy it on tomcat 6 and run it

----------------------------------------------------------------------------------------------------------------------------------

* Problem in putting HTML code in Java
* JSP introduction
* How to write JSP
* Scriplet <% %> and <%= %> expression

*LAB 18 – modify the LAB 5 such that – now system will not only redirect to appropriate page but also show the customize massage (massage should contain name of the user – entered on welcome.html page). Also instead of two HTML page (right.html and wrong.html) now there should be only one JSP page result.jsp*

**In servlet page**

String name = request.getParameter(“hName”);

request.setAttribute(“jName, name);

request.getRequestDispatcher(“result.jsp”).forward(request, response);

OR

getServletContext().getRequestDispatcher(“**/**result.jsp”).forward(request, response);

**In JSP Page**

Hi <%= request.getAttribute(“jName”) %>

AIM of this assignment – is to know – how to create JSP page, use of request dispatcher, use of attribute; how to receive data from Java to HTML page (using JSP), etc

----------------------------------------------------------------------------------------------------------------------------------