**Asutosh Dash-Daily Assignment**

**Sprint 1 Day 6**

1. Create a Servlet that displays “Welcome to Servlets!” in the browser.

Solution:

package com.nisum;  
  
import javax.servlet.\*;  
import javax.servlet.annotation.WebServlet;  
import javax.servlet.http.\*;  
import java.io.IOException;  
  
@WebServlet("/welcome")  
public class WelcomeServlet extends HttpServlet {  
@Override  
protected void doGet(HttpServletRequest req, HttpServletResponse resp)  
throws ServletException, IOException {  
resp.setContentType("text/html");  
resp.getWriter().write("<h1>Welcome to Servlets!</h1>");  
}  
}

1. Develop a Servlet that takes user input (name, email) from an HTML form and displays it.

Solution:

<!DOCTYPE html>  
<html>  
<head>  
 <title>Create User</title>  
</head>  
<body>  
<h2>Create User</h2>  
<form action="create-user" method="post">  
 <label>Username:</label><br/>  
 <input type="text" name="username" required/><br/>  
 <label>Email:</label><br/>  
 <input type="email" name="email" required/><br/>  
 <input type="submit" value="Create User"/>  
</form>  
</body>  
</html>

1. Write a Servlet to demonstrate the doGet() and doPost() methods.

Solution:

package com.nisum;  
  
import javax.servlet.\*;  
import javax.servlet.annotation.WebServlet;  
import javax.servlet.http.\*;  
import java.io.IOException;  
import java.util.HashMap;  
import java.util.HashSet;  
import java.util.Map;  
import java.util.Set;  
  
public class CreateUserServlet extends HttpServlet {  
 private Map<String, User> users = new HashMap<>();  
  
 String username;  
 String Password;  
 String url;  
  
 @Override  
 public void init(ServletConfig config) throws ServletException {  
 this.username= config.getInitParameter("username");  
 this.Password=config.getInitParameter("password");  
 }  
  
 @Override  
 public void init() throws ServletException {  
 ServletContext servletcontext=getServletContext();  
 username=(String) servletcontext.getAttribute("username");  
 }  
  
 @Override  
 protected void doPost(HttpServletRequest req, HttpServletResponse resp)  
 throws ServletException, IOException {  
  
 String username = req.getParameter("username");  
 String email = req.getParameter("email");  
  
 resp.setContentType("text/html");  
 resp.getWriter().write("<h2>User Created Successfully</h2>");  
 resp.getWriter().write("<p>Username: " + username + "</p>");  
 resp.getWriter().write("<p>Email: " + email + "</p>");  
 users.put(email, new User(username, email));  
 }  
  
 @Override  
 protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  
// String pathInfo = req.getPathInfo();  
//  
// resp.setContentType("application/json");  
//  
// String email = pathInfo.substring(1);  
  
 String email= req.getParameter("email");  
 User user= users.get(email);  
 resp.getWriter().write("<h1>"+ user.toString()+"</h1>");  
 }  
  
  
}

1. Create a Servlet that prints the current date and time.

Solution:

WEB.xml:

<!-- DateTimeServlet -->  
<servlet>  
 <servlet-name>DateTimeServlet</servlet-name>  
 <servlet-class>com.nisum.DateTimeServlet</servlet-class>  
</servlet>  
  
<servlet-mapping>  
 <servlet-name>DateTimeServlet</servlet-name>  
 <url-pattern>/currentDateTime</url-pattern>  
</servlet-mapping>

DateTimeServlet:

package com.nisum;  
  
import java.io.IOException;  
import java.io.PrintWriter;  
import java.util.Date;  
import javax.servlet.ServletException;  
import javax.servlet.annotation.WebServlet;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
  
  
public class DateTimeServlet extends HttpServlet {  
 private static final long *serialVersionUID* = 1L;  
  
 protected void doGet(HttpServletRequest request, HttpServletResponse response)  
 throws ServletException, IOException {  
  
 response.setContentType("text/html");  
 PrintWriter out = response.getWriter();  
  
 Date currentDate = new Date();  
  
 out.println("<html><body>");  
 out.println("<h2>Current Date and Time</h2>");  
 out.println("<p>" + currentDate.toString() + "</p>");  
 out.println("</body></html>");  
 }  
}

1. Implement a Servlet that sets and reads cookies.

This was not covered in the sessions

1. Create a Servlet that reads initialization parameters using ServletConfig.

Solution:

package com.nisum;  
  
import java.io.\*;  
import javax.servlet.\*;  
import javax.servlet.http.\*;  
  
public class InitParamServlet extends HttpServlet {  
   
 @Override  
 public void init() throws ServletException {  
   
 String param1 = getServletConfig().getInitParameter("param1");  
 String param2 = getServletConfig().getInitParameter("param2");  
  
   
 System.*out*.println("Initialization Parameter 1: " + param1);  
 System.*out*.println("Initialization Parameter 2: " + param2);  
 }  
  
 @Override  
 protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
   
 response.setContentType("text/html");  
 PrintWriter out = response.getWriter();  
 out.println("<html><body>");  
 out.println("<h2>Initialization Parameters from ServletConfig</h2>");  
 out.println("<p>Param1: " + getServletConfig().getInitParameter("param1") + "</p>");  
 out.println("<p>Param2: " + getServletConfig().getInitParameter("param2") + "</p>");  
 out.println("</body></html>");  
 }  
}

Web.xml:

<web-app xmlns="http://java.sun.com/xml/ns/javaee"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://java.sun.com/xml/ns/javaee  
 http://java.sun.com/xml/ns/javaee/web-app\_3\_1.xsd"  
 version="3.1">  
  
 <!-- Define the Servlet -->  
 <servlet>  
 <servlet-name>InitParamServlet</servlet-name>  
 <servlet-class>InitParamServlet</servlet-class>  
 <init-param>  
 <param-name>param1</param-name>  
 <param-value>Value1</param-value>  
 </init-param>  
 <init-param>  
 <param-name>param2</param-name>  
 <param-value>Value2</param-value>  
 </init-param>  
 </servlet>  
  
 <!-- Map the Servlet to a URL -->  
 <servlet-mapping>  
 <servlet-name>InitParamServlet</servlet-name>  
 <url-pattern>/initparams</url-pattern>  
 </servlet-mapping>  
  
</web-app>

1. Create a Servlet that accesses context parameters using ServletContext.

Solution:

package com.nisum;  
  
import java.io.\*;  
import javax.servlet.\*;  
import javax.servlet.http.\*;  
  
public class ContextParamServlet extends HttpServlet {  
  
 @Override  
 public void doGet(HttpServletRequest request, HttpServletResponse response)  
 throws ServletException, IOException {  
  
   
 ServletContext context = getServletContext();  
  
   
 String appName = context.getInitParameter("appName");  
 String appVersion = context.getInitParameter("appVersion");  
  
   
 response.setContentType("text/html");  
 PrintWriter out = response.getWriter();  
 out.println("<html><body>");  
 out.println("<h2>Application Context Parameters</h2>");  
 out.println("<p>App Name: " + appName + "</p>");  
 out.println("<p>App Version: " + appVersion + "</p>");  
 out.println("</body></html>");  
 }  
}

Web.xml:

<web-app xmlns="http://java.sun.com/xml/ns/javaee"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://java.sun.com/xml/ns/javaee  
 http://java.sun.com/xml/ns/javaee/web-app\_3\_1.xsd"  
 version="3.1">  
  
 <!-- Context Parameters -->  
 <context-param>  
 <param-name>appName</param-name>  
 <param-value>MyWebApp</param-value>  
 </context-param>  
 <context-param>  
 <param-name>appVersion</param-name>  
 <param-value>1.0.0</param-value>  
 </context-param>  
  
 <!-- Servlet Definition -->  
 <servlet>  
 <servlet-name>ContextParamServlet</servlet-name>  
 <servlet-class>ContextParamServlet</servlet-class>  
 </servlet>  
  
 <!-- Servlet Mapping -->  
 <servlet-mapping>  
 <servlet-name>ContextParamServlet</servlet-name>  
 <url-pattern>/contextparams</url-pattern>  
 </servlet-mapping>  
  
</web-app>

1. Build a simple login system using Servlets and HTML. (Hardcode credentials for now)

Solution:

package com.nisum;  
  
import javax.servlet.\*;  
import javax.servlet.http.\*;  
import java.io.\*;  
  
public class LoginServlet extends HttpServlet {  
  
 @Override  
 protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
   
 response.setContentType("text/html");  
 PrintWriter out = response.getWriter();  
 out.println("<html><body>");  
 out.println("<form action='LoginServlet' method='POST'>");  
 out.println("Username: <input type='text' name='username'><br>");  
 out.println("Password: <input type='password' name='password'><br>");  
 out.println("<input type='submit' value='Login'>");  
 out.println("</form>");  
 out.println("</body></html>");  
 }  
  
 @Override  
 protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
 // Hardcoded credentials  
 String username = "admin";  
 String password = "admin123";  
  
 String enteredUsername = request.getParameter("username");  
 String enteredPassword = request.getParameter("password");  
  
 response.setContentType("text/html");  
 PrintWriter out = response.getWriter();  
  
 if (username.equals(enteredUsername) && password.equals(enteredPassword)) {  
 out.println("<html><body>");  
 out.println("<h3>Login Successful</h3>");  
 out.println("<a href='WelcomeServlet'>Go to Welcome Page</a>");  
 out.println("</body></html>");  
 } else {  
 out.println("<html><body>");  
 out.println("<h3>Login Failed</h3>");  
 out.println("<a href='LoginServlet'>Try Again</a>");  
 out.println("</body></html>");  
 }  
 }  
}

1. Write a Servlet to redirect the user to another website using response.sendRedirect().

Solution:

package com.nisum;  
  
import javax.servlet.\*;  
import javax.servlet.http.\*;  
import java.io.\*;  
  
public class RedirectServlet extends HttpServlet {  
  
 @Override  
 protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
   
 response.sendRedirect("https://www.example.com");  
 }  
}

1. Create a Servlet that displays request headers sent by the client.

Implement session tracking using HttpSession in a shopping cart example.

Solution:

package com.nisum;  
  
import javax.servlet.\*;  
import javax.servlet.http.\*;  
import java.io.\*;  
import java.util.\*;  
  
public class DisplayHeadersServlet extends HttpServlet {  
  
 @Override  
 protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
 // Display request headers  
 response.setContentType("text/html");  
 PrintWriter out = response.getWriter();  
  
 out.println("<html><body>");  
 out.println("<h3>Request Headers</h3>");  
 Enumeration<String> headerNames = request.getHeaderNames();  
  
 while (headerNames.hasMoreElements()) {  
 String headerName = headerNames.nextElement();  
 out.println(headerName + ": " + request.getHeader(headerName) + "<br>");  
 }  
  
 out.println("</body></html>");  
 }  
}

1. Create a Servlet that sets session attributes and another that reads them.

Solution:

package com.nisum;  
  
import javax.servlet.\*;  
import javax.servlet.http.\*;  
import java.io.\*;  
  
public class SetSessionAttributesServlet extends HttpServlet {  
  
 @Override  
 protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
  
 HttpSession session = request.getSession();  
  
  
 session.setAttribute("cartItem", "Laptop");  
 session.setAttribute("cartQuantity", 1);  
  
  
 response.setContentType("text/html");  
 PrintWriter out = response.getWriter();  
 out.println("<html><body>");  
 out.println("<h3>Item added to cart</h3>");  
 out.println("<a href='GetSessionAttributesServlet'>View Cart</a>");  
 out.println("</body></html>");  
 }  
}

1. Write a Servlet that accepts user feedback via a form and stores it in memory (using a list).

Solution:

package com.nisum;  
  
import javax.servlet.\*;  
import javax.servlet.http.\*;  
import java.io.\*;  
  
public class GetSessionAttributesServlet extends HttpServlet {  
  
 @Override  
 protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
   
 HttpSession session = request.getSession();  
 String cartItem = (String) session.getAttribute("cartItem");  
 Integer cartQuantity = (Integer) session.getAttribute("cartQuantity");  
  
   
 response.setContentType("text/html");  
 PrintWriter out = response.getWriter();  
 out.println("<html><body>");  
 if (cartItem != null) {  
 out.println("<h3>Item in Cart: " + cartItem + "</h3>");  
 out.println("<h3>Quantity: " + cartQuantity + "</h3>");  
 } else {  
 out.println("<h3>No items in cart.</h3>");  
 }  
 out.println("</body></html>");  
 }  
}

1. Create a Servlet filter that logs every request URL.

Solution:

package com.nisum;  
  
import javax.servlet.\*;  
import javax.servlet.http.\*;  
import java.io.\*;  
import java.util.\*;  
  
public class RequestLoggingFilter implements Filter {  
  
 @Override  
 public void init(FilterConfig filterConfig) throws ServletException {  
   
 }  
  
 @Override  
 public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain) throws IOException, ServletException {  
 HttpServletRequest httpRequest = (HttpServletRequest) request;  
 String url = httpRequest.getRequestURL().toString();  
  
   
 System.*out*.println("Request URL: " + url);  
  
   
 chain.doFilter(request, response);  
 }  
  
 @Override  
 public void destroy() {  
   
 }  
}

1. Develop a multi-page form with Servlets using sessions to persist data between pages.

Solution:

package com.nisum;  
  
import javax.servlet.\*;  
import javax.servlet.http.\*;  
import java.io.\*;  
  
public class Page1Servlet extends HttpServlet {  
  
 @Override  
 protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
  
 response.setContentType("text/html");  
 PrintWriter out = response.getWriter();  
 out.println("<html><body>");  
 out.println("<form action='Page2Servlet' method='POST'>");  
 out.println("Name: <input type='text' name='name'><br>");  
 out.println("<input type='submit' value='Next'>");  
 out.println("</form>");  
 out.println("</body></html>");  
 }  
}

package com.nisum;  
  
import javax.servlet.\*;  
import javax.servlet.http.\*;  
import java.io.\*;  
  
public class Page2Servlet extends HttpServlet {  
  
 @Override  
 protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
   
 HttpSession session = request.getSession();  
 String name = request.getParameter("name");  
 session.setAttribute("name", name);  
  
   
 response.setContentType("text/html");  
 PrintWriter out = response.getWriter();  
 out.println("<html><body>");  
 out.println("Hello, " + name + "<br>");  
 out.println("<a href='SubmitServlet'>Submit</a>");  
 out.println("</body></html>");  
 }  
}

package com.nisum;  
  
import javax.servlet.\*;  
import javax.servlet.http.\*;  
import java.io.\*;  
  
public class SubmitServlet extends HttpServlet {  
  
 @Override  
 protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
  
 HttpSession session = request.getSession();  
 String name = (String) session.getAttribute("name");  
  
 response.setContentType("text/html");  
 PrintWriter out = response.getWriter();  
 out.println("<html><body>");  
 out.println("Name submitted: " + name);  
 out.println("</body></html>");  
 }  
}

1. Use a RequestDispatcher to forward a request from one Servlet to another.

Solution:

package com.nisum;  
  
import javax.servlet.\*;  
import javax.servlet.http.\*;  
import java.io.\*;  
  
public class ForwardServlet extends HttpServlet {  
  
 @Override  
 protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
 RequestDispatcher dispatcher = request.getRequestDispatcher("TargetServlet");  
 dispatcher.forward(request, response);  
 }  
}

1. Create a Servlet that validates input data from an HTML form and shows success/error.

Solution:

package com.nisum;  
  
import javax.servlet.\*;  
import javax.servlet.http.\*;  
import java.io.\*;  
  
public class ValidateServlet extends HttpServlet {  
  
 @Override  
 protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
 String username = request.getParameter("username");  
  
 response.setContentType("text/html");  
 PrintWriter out = response.getWriter();  
 if (username == null || username.isEmpty()) {  
 out.println("<html><body>");  
 out.println("Error: Username is required.");  
 out.println("</body></html>");  
 } else {  
 out.println("<html><body>");  
 out.println("Username: " + username + " is valid.");  
 out.println("</body></html>");  
 }  
 }  
}

1. Create a Servlet that stores feedback/comments in a list and displays all submitted feedbacks.

Implement a mini project with Servlets that includes login, session tracking, and logout functionality.

Create a Servlet-based application that integrates with JDBC to display student data from a database.

Solution:

package com.nisum;  
  
import javax.servlet.\*;  
import javax.servlet.http.\*;  
import java.io.\*;  
import java.util.\*;  
  
public class FeedbackServlet extends HttpServlet {  
  
 private List<String> feedbackList = new ArrayList<>();  
  
 @Override  
 protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
 String feedback = request.getParameter("feedback");  
 if (feedback != null && !feedback.isEmpty()) {  
 feedbackList.add(feedback);  
 }  
  
 response.setContentType("text/html");  
 PrintWriter out = response.getWriter();  
 out.println("<html><body>");  
 out.println("<h3>All Feedbacks:</h3>");  
 for (String fb : feedbackList) {  
 out.println("<p>" + fb + "</p>");  
 }  
 out.println("</body></html>");  
 }  
}