**CGI vs. Servlets: Understanding the Differences**

Common Gateway Interface (CGI)

CGI (Common Gateway Interface) is a protocol that enables web servers to execute external programs (such as Python or Perl scripts) and return dynamic content. It is one of the oldest methods for creating dynamic web pages.

Disadvantages of CGI:

1. Process Creation Overhead: Every client request spawns a new process, consuming significant server resources.
2. Scalability Issues: High traffic leads to excessive process creation, slowing down the server.
3. Poor Memory Management: Each CGI process runs independently, preventing resource sharing.

Java Servlets

Servlets are Java programs that run on a web server and handle requests dynamically. They are executed within a servlet container (like Apache Tomcat) and use threads instead of processes.

Advantages of Servlets over CGI:

1. Efficient Processing: Servlets use a single JVM process with multiple threads, significantly reducing overhead.
2. Better Scalability: Unlike CGI, Servlets can handle multiple requests using lightweight threads instead of spawning new processes.
3. Platform Independence: Servlets are written in Java and run on any system with a JVM.
4. Enhanced Security: Servlets provide built-in security features like session management and authentication.

**Practical Implementation**

CGI Program (Python)

This script handles a form submission using CGI in Python.

**HTML Form (form.html)**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>CGI Form</title></head>

<body>

<form action="cgi-bin/process.py" method="post">

Name: <input type="text" name="name"><br>

Email: <input type="email" name="email"><br>

<input type="submit" value="Submit">

</form></body></html>

**CGI Script (process.py)**

#!/usr/bin/env python3

import cgi

# Get form data

form = cgi.FieldStorage()

name = form.getvalue("name", "Guest")

email = form.getvalue("email", "Not Provided")

# Print HTTP header

print("Content-Type: text/html\n")

print(f"<html><body>")

print(f"<h2>Form Submission Received</h2>")

print(f"<p>Name: {name}</p>")

print(f"<p>Email: {email}</p>")

print(f"</body></html>")

Java Servlet Implementation

To achieve the same functionality with Java Servlets:

**Servlet Configuration**

1. Create a Dynamic Web Project in Eclipse or IntelliJ.
2. Add Tomcat as the web server.
3. Place the HTML form in WebContent/index.html

**HTML Form (form.html)**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Servlet Form</title>

</head>

<body>

<form action="FormServlet" method="post">

Name: <input type="text" name="name"><br>

Email: <input type="email" name="email"><br>

<input type="submit" value="Submit">

</form>

</body>

</html>

**Java Servlet (FormServlet.java)**

import java.io.IOException;

import java.io.PrintWriter;

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("/FormServlet")

public class FormServlet extends HttpServlet {

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

String name = request.getParameter("name");

String email = request.getParameter("email");

out.println("<html><body>");

out.println("<h2>Form Submission Received</h2>");

out.println("<p>Name: " + name + "</p>");

out.println("<p>Email: " + email + "</p>");

out.println("</body></html>");

}

}

Performance Comparison

|  |  |  |
| --- | --- | --- |
| **Feature** | **CGI (Python)** | **Java Servlets** |
| Process Handling | New process for each request | Thread-based execution |
| Performance | Slower (higher overhead) | Faster (multi-threaded) |
| Scalability | Poor (multiple processes) | Excellent (shared memory, threading) |
| Security | Basic security | Advanced security features |

**Conclusion**

1. CGI is simple but inefficient for handling large-scale applications.
2. Servlets are significantly faster, more scalable, and secure, making them ideal for enterprise-level applications.

**Understanding Servlet API**

A Servlet is a Java class that runs on a web server (like Tomcat) and handles HTTP requests.

* HttpServletRequest → Represents client request (query parameters, headers, etc.).
* HttpServletResponse → Represents server response (HTML, JSON, etc.).
* doGet() → Handles HTTP GET requests (e.g., fetching data).
* doPost() → Handles HTTP POST requests (e.g., form submission).

**Create a Basic Java Servlet**

WelcomeServlet.java

import java.io.IOException;

import java.io.PrintWriter;

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("/welcome") // Maps this servlet to /welcome URL

public class WelcomeServlet extends HttpServlet

// Handles GET requests

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

// Handles POST requests

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

// Common method for handling GET & POST requests

private void processRequest(HttpServletRequest request, HttpServletResponse response)

throws IOException {

response.setContentType("text/html"); // Set response type

PrintWriter out = response.getWriter(); // Writer to send response

// Retrieving query parameters from URL

String name = request.getParameter("name");

if (name == null || name.isEmpty()) {

name = "Guest";

}

// Retrieving client request details

String clientIP = request.getRemoteAddr(); // Client IP Address

String userAgent = request.getHeader("User-Agent"); // User-Agent info

// Sending response to client

out.println("<html><body>");

out.println("<h2>Welcome, " + name + "!</h2>");

out.println("<p>Your IP Address: " + clientIP + "</p>");

out.println("<p>Your User-Agent: " + userAgent + "</p>");

out.println("</body></html>");}}

**Deploying the Servlet**

Steps to Run the Servlet

1. Set up Apache Tomcat in Eclipse or IntelliJ.
2. Deploy the servlet and start the Tomcat server.
3. Access the servlet using a browser or Postman.

**Example URL:** http://localhost:8080/YourProjectName/welcome?name=John

**Expected Output**

Response in Browser

<html>

<body>

<h2>Welcome, John!</h2>

<p>Your IP Address: 192.168.1.100</p>

<p>Your User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)</p>

</body>

</html>

**Explanation of Features**

* Handles GET & POST Requests → doGet() and doPost() call processRequest().
* Retrieves Query Parameters → Extracts name from URL (request.getParameter("name")).
* Displays Client Info → Fetches IP Address and User-Agent from HttpServletRequest.
* Sends Dynamic HTML Response → Uses PrintWriter to generate output.

**Enhancements**

* Session Management: Store user data using HttpSession.
* Database Integration: Retrieve user info from a database.
* JSON Response: Use response.setContentType("application/json") to return JSON data.

**ServletConfig & ServletContext Implementation**

* ServletConfig: Used to retrieve initialization parameters specific to a servlet from web.xml (e.g., database credentials).
* ServletContext: Used to share data across multiple servlets within an application.

Steps:

1. Define init parameters (e.g., database config) in web.xml.
2. Use ServletConfig to fetch them in a servlet.

Location: src/main/webapp/WEB-INF/web.xml

**web.xml**

<web-app xmlns="http://java.sun.com/xml/ns/javaee"

version="3.1">

<!-- Servlet Configuration -->

<servlet>

<servlet-name>DBServlet</servlet-name>

<servlet-class>com.example.DBServlet</servlet-class>

<!-- Init Parameters -->

<init-param>

<param-name>dbUrl</param-name>

<param-value>jdbc:mysql://localhost:3306/mydb</param-value>

</init-param>

<init-param>

<param-name>dbUser</param-name>

<param-value>root</param-value>

</init-param>

<init-param>

<param-name>dbPassword</param-name>

<param-value>password123</param-value>

</init-param>

</servlet>

<servlet-mapping>

<servlet-name>DBServlet</servlet-name>

<url-pattern>/dbconfig</url-pattern>

</servlet-mapping>

</web-app>

**Servlet Code: DBServlet.java**

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletConfig;

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("/dbconfig")

public class DBServlet extends HttpServlet {

private String dbUrl, dbUser, dbPassword;

// Initialize parameters from web.xml

public void init(ServletConfig config) throws ServletException {

dbUrl = config.getInitParameter("dbUrl");

dbUser = config.getInitParameter("dbUser");

dbPassword = config.getInitParameter("dbPassword");

}

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>Database Configuration</h2>");

out.println("<p>URL: " + dbUrl + "</p>");

out.println("<p>User: " + dbUser + "</p>");

out.println("<p>Password: " + dbPassword + "</p>");

out.println("</body></html>");}}

**Expected Output**

When accessing:

http://localhost:8080/YourProjectName/dbconfig

You'll see:

Database Configuration

URL: jdbc:mysql://localhost:3306/mydb

User: root

Password: password123

**ServletContext Implementation**

Goal: Store and share application-wide data across multiple servlets.

Modify **web.xml** to Add 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</param-value>

</context-param>

**Servlet to Fetch Application Context (AppInfoServlet.java)**

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.ServletContext;

@WebServlet("/appinfo")

public class AppInfoServlet extends HttpServlet {

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

// Get ServletContext

ServletContext context = getServletContext();

String appName = context.getInitParameter("appName");

String appVersion = context.getInitParameter("appVersion");

out.println("<html><body>");

out.println("<h2>Application Info</h2>");

out.println("<p>App Name: " + appName + "</p>");

out.println("<p>App Version: " + appVersion + "</p>");

out.println("</body></html>");}}

**Expected Output**

When accessing:

http://localhost:8080/YourProjectName/appinfo

You'll see:

Application Info

App Name: MyWebApp

App Version: 1.0

**Using ServletContext to Share Data Across Servlets**

Goal: Store a visitor counter and share it across multiple servlets.

Servlet to Track Visitors (VisitorServlet.java)

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.ServletContext;

@WebServlet("/visitor")

public class VisitorServlet extends HttpServlet {

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

// Get ServletContext

ServletContext context = getServletContext();

// Retrieve visitor count, increment it, and store it back

Integer visitorCount = (Integer) context.getAttribute("visitorCount");

if (visitorCount == null) {

visitorCount = 1;}

else {

visitorCount++;}

context.setAttribute("visitorCount", visitorCount);

out.println("<html><body>");

out.println("<h2>Visitor Counter</h2>");

out.println("<p>Total Visitors: " + visitorCount + "</p>");

out.println("</body></html>");}}

**Expected Output**

When accessing:

http://localhost:8080/YourProjectName/visitor

Each refresh will increment the count:

Visitor Counter

Total Visitors: 1

Visitor Counter

Total Visitors: 2

Visitor Counter

Total Visitors: 3

**Session Management in Servlets**

Session management is essential for maintaining user-specific data across multiple requests. Below are three ways to track sessions in Servlets:

* Cookies → Store and retrieve session data using browser cookies.
* URL Rewriting → Pass session data through the URL.
* HttpSession API → Use built-in session management with HttpSession.

**Implementing Session Tracking using Cookies**

Servlet to Set Cookie (LoginServlet.java)

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.Cookie;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

@WebServlet("/login")

public class LoginServlet extends HttpServlet {

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

// Retrieve username from form input

String username = request.getParameter("username");

// Create and set cookie

Cookie userCookie = new Cookie("username", username);

userCookie.setMaxAge(60 \* 60); // 1 hour

response.addCookie(userCookie);

out.println("<html><body>");

out.println("<h2>Welcome, " + username + "!</h2>");

out.println("<a href='dashboard'>Go to Dashboard</a>");

out.println("</body></html>");}}

**Servlet to Retrieve Cookie (DashboardServlet.java)**

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.Cookie;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

@WebServlet("/dashboard")

public class DashboardServlet extends HttpServlet {

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

// Retrieve cookies

Cookie[] cookies = request.getCookies();

String username = "Guest";

if (cookies != null) {

for (Cookie cookie : cookies) {

if (cookie.getName().equals("username")) {

username = cookie.getValue();}}}

out.println("<html><body>");

out.println("<h2>Welcome Back, " + username + "!</h2>");

out.println("</body></html>");}}

Expected Output

1. User logs in via http://localhost:8080/YourProjectName/login

* Submits the form with the username.

1. User visits the dashboard http://localhost:8080/YourProjectName/dashboard

* Output: "Welcome Back, John!" (retrieved from the cookie).

**Implementing Session Tracking using URL Rewriting**

Login Servlet with URL Rewriting (URLLoginServlet.java)

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

@WebServlet("/url-login")

public class URLLoginServlet extends HttpServlet {

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

String username = request.getParameter("username");

// Create session and store username

HttpSession session = request.getSession();

session.setAttribute("username", username);

// URL rewriting: Append session ID

String url = "url-dashboard?JSESSIONID=" + session.getId();

out.println("<html><body>");

out.println("<h2>Welcome, " + username + "!</h2>");

out.println("<a href='" + url + "'>Go to Dashboard</a>");

out.println("</body></html>");}}

**Dashboard Servlet using URL Rewriting (URLDashboardServlet.java)**

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

@WebServlet("/url-dashboard")

public class URLDashboardServlet extends HttpServlet {

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

// Retrieve session

HttpSession session = request.getSession(false);

String username = (session != null) ? (String) session.getAttribute("username") : "Guest";

out.println("<html><body>");

out.println("<h2>Welcome Back, " + username + "!</h2>");

out.println("</body></html>");}}

**Expected Output**

* User logs in via http://localhost:8080/YourProjectName/url-login
* Session ID is appended to the URL:

http://localhost:8080/YourProjectName/url-dashboard?JSESSIONID=123456

* Dashboard retrieves session ID and username.

**Implementing Session Tracking using HttpSession**

**Servlet to Set User Preferences (PreferenceServlet.java)**

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

@WebServlet("/set-preference")

public class PreferenceServlet extends HttpServlet {

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

// Retrieve preferences

String theme = request.getParameter("theme");

String language = request.getParameter("language");

// Store in session

HttpSession session = request.getSession();

session.setAttribute("theme", theme);

session.setAttribute("language", language);

out.println("<html><body>");

out.println("<h2>Preferences Saved!</h2>");

out.println("<a href='get-preference'>View Preferences</a>");

out.println("</body></html>");}}

**Servlet to Retrieve Preferences (GetPreferenceServlet.java)**

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

@WebServlet("/get-preference")

public class GetPreferenceServlet extends HttpServlet {

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

// Retrieve session

HttpSession session = request.getSession(false);

String theme = (session != null) ? (String) session.getAttribute("theme") : "Default";

String language = (session != null) ? (String) session.getAttribute("language") : "English";

out.println("<html><body>");

out.println("<h2>Saved Preferences</h2>");

out.println("<p>Theme: " + theme + "</p>");

out.println("<p>Language: " + language + "</p>");

out.println("</body></html>");}}

**Servlet Communication using RequestDispatcher**

**Login Page (index.html)**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Login</title>

</head>

<body>

<h2>Login</h2>

<!-- Display error message if present -->

<p style="color:red;">${errorMessage}</p>

<form action="login" method="post">

<label>Username:</label>

<input type="text" name="username" required>

<br><br>

<label>Password:</label>

<input type="password" name="password" required>

<br><br>

<button type="submit">Login</button>

</form>

</body>

</html>

**LoginServlet (Validates User Credentials)**

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.RequestDispatcher;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse

@WebServlet("/login")

public class LoginServlet extends HttpServlet {

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

String username = request.getParameter("username");

String password = request.getParameter("password");

// Hardcoded credentials for demonstration

if ("admin".equals(username) && "password123".equals(password)) {

// Forward to WelcomeServlet

RequestDispatcher dispatcher = request.getRequestDispatcher("welcome");

dispatcher.forward(request, response);

} else {

// Redirect back to login page with error message

response.sendRedirect("index.html?error=Invalid username or password");

}}}

**WelcomeServlet (Displays Success Message)**

import java.io.IOException;

import java.io.PrintWriter;

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("/welcome")

public class WelcomeServlet extends HttpServlet {

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

String username = request.getParameter("username");

out.println("<html><body>");

out.println("<h2>Welcome, " + username + "!</h2>");

out.println("</body></html>");}}

**Implementing Servlet Filters**

**Logging Filter (LoggingFilter.java)**

import java.io.IOException;

import java.io.PrintWriter;

import java.text.SimpleDateFormat;

import java.util.Date;

import javax.servlet.Filter;

import javax.servlet.FilterChain;

import javax.servlet.FilterConfig;

import javax.servlet.ServletException;

import javax.servlet.ServletRequest;

import javax.servlet.ServletResponse;

import javax.servlet.annotation.WebFilter;

import javax.servlet.http.HttpServletRequest;

@WebFilter("/\*") // Apply filter to all requests

public class LoggingFilter implements Filter {

public void init(FilterConfig filterConfig) throws ServletException {

System.out.println("Logging Filter Initialized");}

public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain)

throws IOException, ServletException {

HttpServletRequest req = (HttpServletRequest) request;

String ipAddress = request.getRemoteAddr();

String url = req.getRequestURI();

String timeStamp = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss").format(new Date());

System.out.println("[LOG] Request from IP: " + ipAddress + ", Time: " + timeStamp + ", URL: " + url);

// Continue request processing

chain.doFilter(request, response);}

public void destroy() {

System.out.println("Logging Filter Destroyed");}}

**Expected Log Output**

[LOG] Request from IP: 192.168.1.10, Time: 2025-03-30 10:30:45, URL: /protectedPage

[LOG] Request from IP: 192.168.1.12, Time: 2025-03-30 10:31:20, URL: /login

**Authentication Filter (AuthFilter.java)**

import java.io.IOException;

import javax.servlet.Filter;

import javax.servlet.FilterChain;

import javax.servlet.FilterConfig;

import javax.servlet.ServletException;

import javax.servlet.ServletRequest;

import javax.servlet.ServletResponse;

import javax.servlet.annotation.WebFilter;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

@WebFilter("/protectedPage") // Protects the "protectedPage" URL

public class AuthFilter implements Filter {

public void init(FilterConfig filterConfig) throws ServletException {

System.out.println("Authentication Filter Initialized");

}

public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain)

throws IOException, ServletException {

HttpServletRequest req = (HttpServletRequest) request;

HttpServletResponse res = (HttpServletResponse) response;

HttpSession session = req.getSession(false);

// Check if the user is logged in

if (session == null || session.getAttribute("username") == null) {

System.out.println("[AUTH] Unauthorized access attempt. Redirecting to login page.");

res.sendRedirect("login.html"); // Redirect to login page

} else {

chain.doFilter(request, response); // Continue to protected resource}}

public void destroy() {

System.out.println("Authentication Filter Destroyed");}}

**Login Servlet (LoginServlet.java)**

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

@WebServlet("/login")

public class LoginServlet extends HttpServlet {

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 out = response.getWriter();

i ("admin".equals(username) && "password123".equals(password)) {

// Create a session

HttpSession session = request.getSession();

session.setAttribute("username", username);

response.sendRedirect("protectedPage");

} else {

out.println("<html><body>");

out.println("<h3 style='color:red;'>Invalid username or password.</h3>");

out.println("<a href='login.html'>Try Again</a>");

out.println("</body></html>");}}}

**Protected Servlet (ProtectedServlet.java)**

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

@WebServlet("/protectedPage")

public class ProtectedServlet extends HttpServlet {

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

HttpSession session = request.getSession(false);

String username = (session != null) ? (String) session.getAttribute("username") : null;

out.println("<html><body>");

if (username != null) {

out.println("<h2>Welcome, " + username + "! You are in the protected area.</h2>");

out.println("<a href='logout'>Logout</a>");

} else {

out.println("<h3>Access Denied. Please <a href='login.html'>Login</a></h3>");}

out.println("</body></html>");}]

**Logout Servlet (LogoutServlet.java)**

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;

import javax.servlet.http.HttpSession;

@WebServlet("/logout")

public class LogoutServlet extends HttpServlet {

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

HttpSession session = request.getSession(false);

if (session != null) {

session.invalidate(); // Destroy session}

response.sendRedirect("login.html");}}