**Detailed Manual Steps to Execute Servlet:**

1. Set Up Environment:

Install Eclipse IDE.

Install Apache Tomcat.

2. Create Project:

Open Eclipse.

File > New > Dynamic Web Project.

Name the project Program8 > Next > Next > Check web.xml deployment descriptor > Finish

3. Create Servlet:

Java Resources > src/main/java -> Right-click -> New > Servlet.

Name the servlet GreetingServlet

Name the Java package & Class name > Next > Next > Check doPost & doGet (if required in the program) > Finish

Place the generated code in the provided GreetingServlet.java File.

\* Save the File.

4. Create HTML:

main > webapp -> Right-click -> New > HTML File.

Name the File Name > Next > Finish

Place the generated code in the provided html File.

\* Save the File.

5. Configure web.xml:

webapp > /WEB-INF/ > web.xml->

In web.xml File > source

Name the file web.xml.

Add the provided web.xml content.

\* Save the File.

6. Deploy Application:

Right-click the project -> Run As > Run on Server.

Choose Apache Tomcat > Next > add the project > Finish.

7. Access and Test:

Open a web browser.

Go to http://localhost:8080/GreetingApp/greet.

Interact with the form.

This will guide you through creating a servlet that displays a greeting message and accepts a username from the client.

**Reference:** https://www.javatpoint.com/creating-servlet-in-eclipse-ide

**8. A program to display greeting message on the browser “Hello User Name”, “How Are You?”, accept username from the client using servlet.**

**Program:**

**Servlet File:**

package javaprogram;

import jakarta.servlet.\*;

import jakarta.servlet.http.\*;

public class GreetingServlet extends HttpServlet {

public GreetingServlet() {

super();

// TODO Auto-generated constructor stub

}

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

// TODO Auto-generated method stub

//response.getWriter().append("Served at: ").append(request.getContextPath());

response.setContentType("text/html");

// Get username parameter from the request

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

// Generate greeting message

String greetingMessage = "Hello " + (username != null ? username : "User") + ", How Are You?";

// Write greeting message to the response

response.getWriter().println("<html><body>");

response.getWriter().println("<h1>" + greetingMessage + "</h1>");

response.getWriter().println("</body></html>");

}

}

**web.xml File:**

<servlet>

<servlet-name>GreetingServlet</servlet-name>

<servlet-class>javaprogram.GreetingServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>GreetingServlet</servlet-name>

<url-pattern>/greet</url-pattern>

</servlet-mapping>

**Note:** Replace above with the existing one.

His configuration maps the servlet to the URL pattern "/greet", so when a client sends a request to "/greet" with a parameter "username", the servlet will handle it.

To test the servlet, you can create a simple HTML form that submits the username to the servlet:

**HTML File:**

<!DOCTYPE html>

<html>

<head>

<title>Greeting Form</title>

</head>

<body>

<form action="greet" method="get">

Username: <input type="text" name="username">

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

</form>

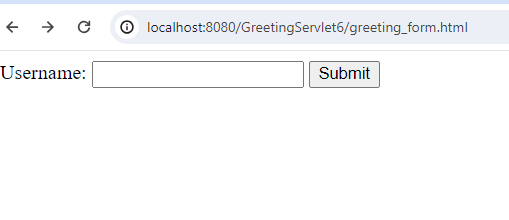
</body>

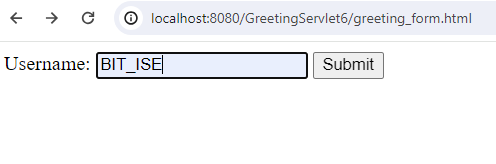
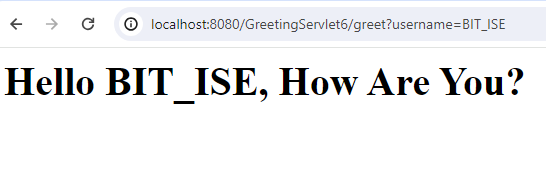
</html>

Save this HTML code as "greeting\_form.html" and deploy it along with the servlet. When you open the HTML file in a browser, it will display a form where you can enter a username. Upon submitting the form, it will send a GET request to the servlet, which will generate the greeting message and display it on the browser.

Remember to deploy your servlet and HTML file properly in a servlet container like Apache Tomcat.

**OUTPUT:**



**9. A servlet program to display the name, USN, and total marks by accepting student detail.**

**Servlet File:**

package javaprogram;

import jakarta.servlet.\*;

import jakarta.servlet.http.\*;

public class StudentDetailsServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

public StudentDetailsServlet() {

super();

// TODO Auto-generated constructor stub

}

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

// TODO Auto-generated method stub

response.setContentType("text/html");

// Get student details from the request

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

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

int totalMarks = Integer.parseInt(request.getParameter("totalMarks"));

// Generate student details message

String detailsMessage = "Student Name: " + name + "<br>"

+ "USN: " + usn + "<br>"

+ "Total Marks: " + totalMarks;

// Write student details message to the response

response.getWriter().println("<html><body>");

response.getWriter().println("<h1>Student Details</h1>");

response.getWriter().println("<p>" + detailsMessage + "</p>");

response.getWriter().println("</body></html>");

}

}

**web.xml File:**

<servlet>

<description></description>

<display-name>StudentDetailsServlet</display-name>

<servlet-name>StudentDetailsServlet</servlet-name>

<servlet-class>javaprogram.StudentDetailsServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>StudentDetailsServlet</servlet-name>

<url-pattern>/studentDetails</url-pattern>

</servlet-mapping>

**Note:** Replace above with the existing one.

This configuration maps the servlet to the URL pattern "/studentDetails", so when a client sends a request to "/studentDetails" with parameters "name", "usn", and "totalMarks", the servlet will handle it.

To test the servlet, you can create a simple HTML form that submits the student details to the servlet:

**HTML File:**

<!DOCTYPE html>

<html>

<head>

<title>Student Details Form</title>

</head>

<body>

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

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

USN: <input type="text" name="usn"><br>

Total Marks: <input type="text" name="totalMarks"><br>

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

</form>

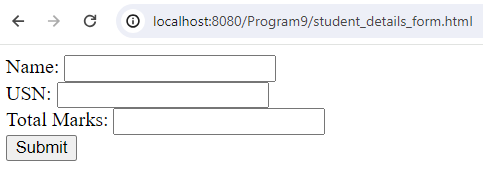
</body>

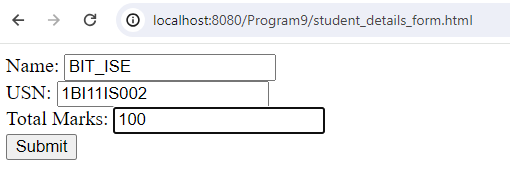
</html>

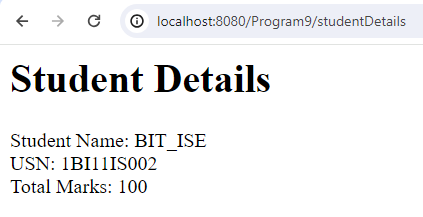
Save this HTML code as "student\_details\_form.html" and deploy it along with the servlet. When you open the HTML file in a browser, it will display a form where you can enter student details. Upon submitting the form, it will send a POST request to the servlet, which will generate the student details message and display it on the browser.

Remember to deploy your servlet and HTML file properly in a servlet container like Apache Tomcat.

**OUTPUT:**

****

****

****

**10. A Java program to create and read the cookie for the given cookie name as “EMPID” and its value as “AN2356”.**

**Program:**

**\\ Set Cookie Servlet File:**

package javaprogram;

import jakarta.servlet.\*;

import jakarta.servlet.http.Cookie;

import jakarta.servlet.http.\*;

@WebServlet("/SetCookie")

public class SetCookieServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

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

// Create a cookie

Cookie cookie = new Cookie("EMPID", "AN2356");

// Set cookie properties

cookie.setMaxAge(3600); // Cookie expires in 1 hour

cookie.setPath("/"); // Cookie is accessible from the root path

// Add cookie to the response

response.addCookie(cookie);

// Inform the user that the cookie is set

response.setContentType("text/html");

response.getWriter().println("Cookie 'EMPID' with value 'AN2356' is set.");

}

}

**\\ Read Cookie Servlet File:**

package javaprogram;

import jakarta.servlet.\*;

import jakarta.servlet.http.Cookie;

import jakarta.servlet.http.\*;

@WebServlet("/ReadCookie")

public class ReadCookieServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

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

// Get all cookies from the request

Cookie[] cookies = request.getCookies();

// Find the cookie with name "EMPID"

String empId = null;

if (cookies != null) {

for (Cookie cookie : cookies) {

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

empId = cookie.getValue();

break;

}

}

}

// Respond with the value of the cookie

response.setContentType("text/html");

if (empId != null) {

response.getWriter().println("EMPID: " + empId);

} else {

response.getWriter().println("No cookie found with name EMPID");

}

}

}

**web.xml Set Cookie File:**

<servlet>

<description></description>

<servlet-name>SetCookieServlet</servlet-name>

<servlet-class>javaprogram.SetCookieServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>SetCookieServlet</servlet-name>

<url-pattern>/SetCookieServlet</url-pattern>

</servlet-mapping>

<servlet>

<description></description>

**Note:** Replace above with the existing one.

**web.xml Read Cookie File:**

<servlet-name>ReadCookieServlet</servlet-name>

<servlet-class>javaprogram.ReadCookieServlet</servlet-class>

</servlet>

<servlet-mapping>

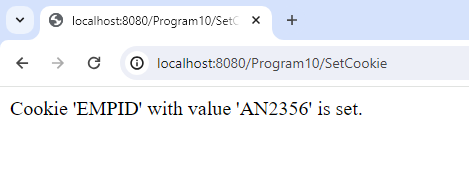
<servlet-name>ReadCookieServlet</servlet-name>

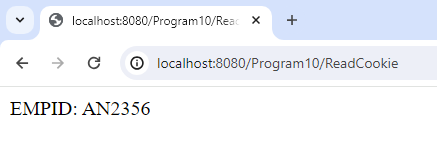
<url-pattern>/ReadCookieServlet</url-pattern>

</servlet-mapping>

**Note:** Replace above with the existing one.

**OUTPUT:**

****

****

**11. Write a JAVA Program to insert data into Student DATA BASE and retrieve info based on particular queries(For example update, delete, search etc…).**

**Program:**

package javaprogram;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

public class StudentDatabaseExample {

// JDBC URL, username, and password of MySQL server

private static final String JDBC\_URL = "jdbc:mysql://localhost:3306/studentdb";

private static final String USERNAME = "root";

private static final String PASSWORD = "ise123";

public static void main(String[] args) {

try {

// Register JDBC driver

Class.forName("com.mysql.cj.jdbc.Driver");

// Open a connection

Connection conn = DriverManager.getConnection(JDBC\_URL, USERNAME, PASSWORD);

// Insert data into Student table

insertData(conn, "John", 22, "CS");

insertData(conn, "Alice", 21, "Math");

insertData(conn, "Bob", 23, "Physics");

// Update data in Student table

updateData(conn, "John", 23);

// Delete data from Student table

deleteData(conn, "Alice");

// Search for data in Student table

searchStudent(conn, "Bob");

// Close connection

conn.close();

} catch (Exception e) {

e.printStackTrace();

}

}

public static void insertData(Connection conn, String name, int age, String major) throws SQLException {

String query = "INSERT INTO Student (name, age, major) VALUES (?, ?, ?)";

PreparedStatement pstmt = conn.prepareStatement(query);

pstmt.setString(1, name);

pstmt.setInt(2, age);

pstmt.setString(3, major);

pstmt.executeUpdate();

pstmt.close();

}

public static void updateData(Connection conn, String name, int newAge) throws SQLException {

String query = "UPDATE Student SET age = ? WHERE name = ?";

PreparedStatement pstmt = conn.prepareStatement(query);

pstmt.setInt(1, newAge);

pstmt.setString(2, name);

pstmt.executeUpdate();

pstmt.close();

}

public static void deleteData(Connection conn, String name) throws SQLException {

String query = "DELETE FROM Student WHERE name = ?";

PreparedStatement pstmt = conn.prepareStatement(query);

pstmt.setString(1, name);

pstmt.executeUpdate();

pstmt.close();

}

public static void searchStudent(Connection conn, String name) throws SQLException {

String query = "SELECT \* FROM Student WHERE name = ?";

PreparedStatement pstmt = conn.prepareStatement(query);

pstmt.setString(1, name);

ResultSet rs = pstmt.executeQuery();

while (rs.next()) {

int id = rs.getInt("id");

String studentName = rs.getString("name");

int age = rs.getInt("age");

String major = rs.getString("major");

System.out.println("ID: " + id + ", Name: " + studentName + ", Age: " + age + ", Major: " + major);

}

rs.close();

pstmt.close();

}

}

**//**

CREATE DATABASE studentdb;

USE studentdb;

CREATE TABLE Student (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(50) NOT NULL,

age INT NOT NULL,

major VARCHAR(50) NOT NULL

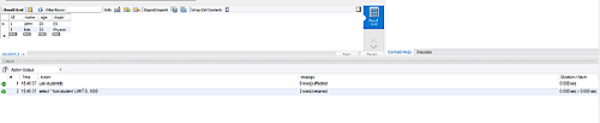
);

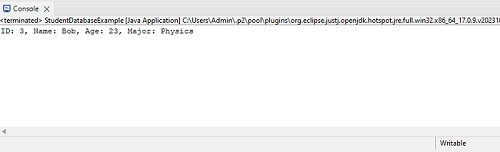
**//**

use studentdb;

select \* from student;

**OUTPUT:**

****

****

**12. A program to design the Login page and validating the USER\_ID and PASSWORD using JSP and DataBase.**

**Program:**

**Step 1:** Set Up the MySQL Database

Create a Database and Table:

Open your MySQL command line or a GUI tool like MySQL Workbench.

Create a database and a users table:

CREATE DATABASE userdb;

USE userdb;

CREATE TABLE users (

user\_id VARCHAR(50) PRIMARY KEY,

password VARCHAR(50)

);

**Insert a sample user:**

INSERT INTO users (user\_id, password) VALUES ('testuser', 'testpassword');

**Step 2:** Set Up Eclipse Project

Open Eclipse and select your workspace.

Create a Dynamic Web Project:

Go to File > New > Dynamic Web Project.

Name your project (e.g., JSPLoginExample).

Click Next and configure the project facets. Ensure that the version for Dynamic Web Module is set to the latest supported version and select your Apache Tomcat server.

Click Finish.

**Step 3:** Create JSP Pages

Create login.jsp:

Right-click the WebContent folder in your project (src/main/java).

Go to New > JSP File and name it login.jsp.

**Add the following code:**

<!DOCTYPE html>

<html>

<head>

<title>Login Page</title>

</head>

<body>

<h2>Login Page</h2>

<form action="validate.jsp" method="post">

User ID: <input type="text" name="user\_id"><br>

Password: <input type="password" name="password"><br>

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

</form>

</body>

</html>

Create validate.jsp:

Right-click the WebContent folder in your project (src/main/java).

Go to New > JSP File and name it validate.jsp.

**Add the following code:**

<%@ page import="java.sql.\*" %>

<!DOCTYPE html>

<html>

<head>

<title>Login Validation</title>

</head>

<body>

<h2>Login Validation</h2>

<%@ page language="java" %>

<%@ page contentType="text/html; charset=ISO-8859-1" %>

<%

String user\_id = request.getParameter("user\_id");

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

// JDBC URL, username, and password of your database

String JDBC\_URL = "jdbc:mysql://localhost:3306/userdb";

String USERNAME = "root"; // change to your MySQL username

String PASSWORD = "rootpassword"; // change to your MySQL password

try {

// Register JDBC driver

Class.forName("com.mysql.cj.jdbc.Driver");

// Open a connection

Connection conn = DriverManager.getConnection(JDBC\_URL, USERNAME, PASSWORD);

// Create a prepared statement to prevent SQL injection

String sql = "SELECT \* FROM users WHERE user\_id=? AND password=?";

PreparedStatement stmt = conn.prepareStatement(sql);

stmt.setString(1, user\_id);

stmt.setString(2, password);

// Execute the query

ResultSet rs = stmt.executeQuery();

// Check if user\_id and password match

if (rs.next()) {

out.println("<p>Login successful! Welcome, " + user\_id + "!</p>");

} else {

out.println("<p>Invalid user\_id or password.</p>");

}

// Close connections

rs.close();

stmt.close();

conn.close();

} catch (Exception e) {

out.println("<p>Error: " + e.getMessage() + "</p>");

}

%>

</body>

</html>

**Step 4:** Deploy and Run the Project

Start the Server:

In the Servers view, right-click your Tomcat server and select Start.

Access the Application:

Open a web browser and go to http://localhost:8080/JSPLoginExample/login.jsp.

Step 5: Test the Application

Login:

Enter the user ID and password you inserted into the database (testuser / testpassword).

Click Login.

You should see a message indicating whether the login was successful or not.

**OUTPUT:**



