

INTERNSHIP TASK

Task1: What is Servlet and jsp.

❖ SERVLET:

Servlets are the Java programs that run on the Java-enabled web server or application server. They are used to handle the request obtained from the web server, process the request, produce the response, and then send response back to the web server.

- Properties of Servlets:
 - Servlets work on the server-side.
 - Servlets are capable of handling complex requests obtained from web server.
- Execution of Servlets involves six basic steps:
 1. The clients send the request to the web server.
 2. The web server receives the request.
 3. The web server passes the request to the corresponding servlet.
 4. The servlet processes the request and generates the response in the form of output.
 5. The servlet sends the response back to the web server.
 6. The web server sends the response back to the client and the client browser displays it on the screen.

❖ JSP

Jakarta Server Pages (JSP; formerly Java Server Pages) is a collection of technologies that helps software developers create dynamically generated web pages based on HTML, XML, SOAP, or other document types. Released in 1999 by Sun Microsystems JSP is similar to PHP and ASP, but uses the Java programming language. To deploy and run Jakarta Server Pages, a compatible web server with a servlet container, such as Apache Tomcat or Jetty, is required.

JSP allows Java code and certain predefined actions to be interleaved with static web markup content, such as HTML. The resulting page is compiled and executed on the server to deliver a document. The compiled pages, as well as any dependent Java libraries, contain Java bytecode rather than machine code. Like any other .jar or Java program, code must be executed within a Java virtual machine (JVM) that interacts with the server's host operating system to provide an abstract, platform-neutral environment. JSPs are usually used to deliver HTML and XML documents, but through the use of OutputStream, they can deliver other types of data as well.

JSP pages use several delimiters for scripting functions:

- The most basic is `<%.....%>` which encloses a JSP *scriptlet*. A scriptlet is a fragment of Java code that runs when the user requests the page.
- Other common delimiters include `<%=.....%>` for *expressions*, where the scriptlet and delimiters are replaced with the result of evaluating the expression, and
- *directives*, denoted with `<% @.....%>`

Task 2: What is Request Dispatcher and send redirect

❖ RequestDispatcher:

The RequestDispatcher interface provides the facility of dispatching the request to another resource it may be html, servlet or jsp. This interface can also be used to include the content of another resource also. It is one of the way of servlet collaboration.

The RequestDispatcher interface provides two methods. They are:

- *public void forward(ServletRequest request, ServletResponse response) throws ServletException, java.io.IOException:*
 - Forwards a request from a servlet to another resource (servlet, JSP file, or HTML file) on the server.
- *public void include(ServletRequest request, ServletResponse response) throws ServletException, java.io.IOException:*
 - Includes the content of a resource (servlet, JSP page, or HTML file) in the response.

The getRequestDispatcher() method of ServletRequest interface returns the object of RequestDispatcher. Syntax:

- *public RequestDispatcher getRequestDispatcher(String resource);*

❖ sendRedirect

The sendRedirect() method of HttpServletResponse interface can be used to redirect response to another resource, it may be servlet, jsp or html file. It accepts relative as well as absolute URL. It works at client side because it uses the url bar of the browser to make another request. So, it can work inside and outside the server.

Difference between forward() method of RequestDispatcher and sendRedirect() method

forward() method	sendRedirect() method
The forward() method works at server side.	The sendRedirect() method works at client side.
It sends the same request and response objects to another servlet.	It always sends a new request.
It can work within the server only.	It can be used within and outside the server.
Example: request.getRequestDispatcher("servlet2").forward(request, response);	Example: response.sendRedirect("servlet2"0;

Task: 3 Create demo of generic servlet and http servlet and when to use them in real life.

GenericServlet is protocol independent and can be used with any protocol such as HTTP, SMTP, FTP, and, CGI while HttpServlet is protocol dependent and only used with HTTP protocol.

GenericController

```
package controller;

import java.io.IOException;
import javax.servlet.GenericServlet;
import javax.servlet.ServletException;
import javax.servlet.ServletRequest;
import javax.servlet.ServletResponse;
import javax.servlet.annotation.WebServlet;

@WebServlet("/GenericController")
public class GenericController extends GenericServlet {
    private static final long serialVersionUID = 1L;
    public GenericController() {
        super();
    }
    public void service(ServletRequest request, ServletResponse response)
    throws ServletException, IOException {
        response.getWriter().append("<h1>This is Genric servlet</h1>");
    }
}
```

ServletController

```
package controller;

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;

@WebServlet("/ServletController")
public class ServletController extends HttpServlet {
    private static final long serialVersionUID = 1L;
    public ServletController() {
        super();
    }
    protected void doGet(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
        response.getWriter().append("<h1>This is HTTP Servlet</h1>");
    }
    protected void doPost(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
        doGet(request, response);
    }
}
```

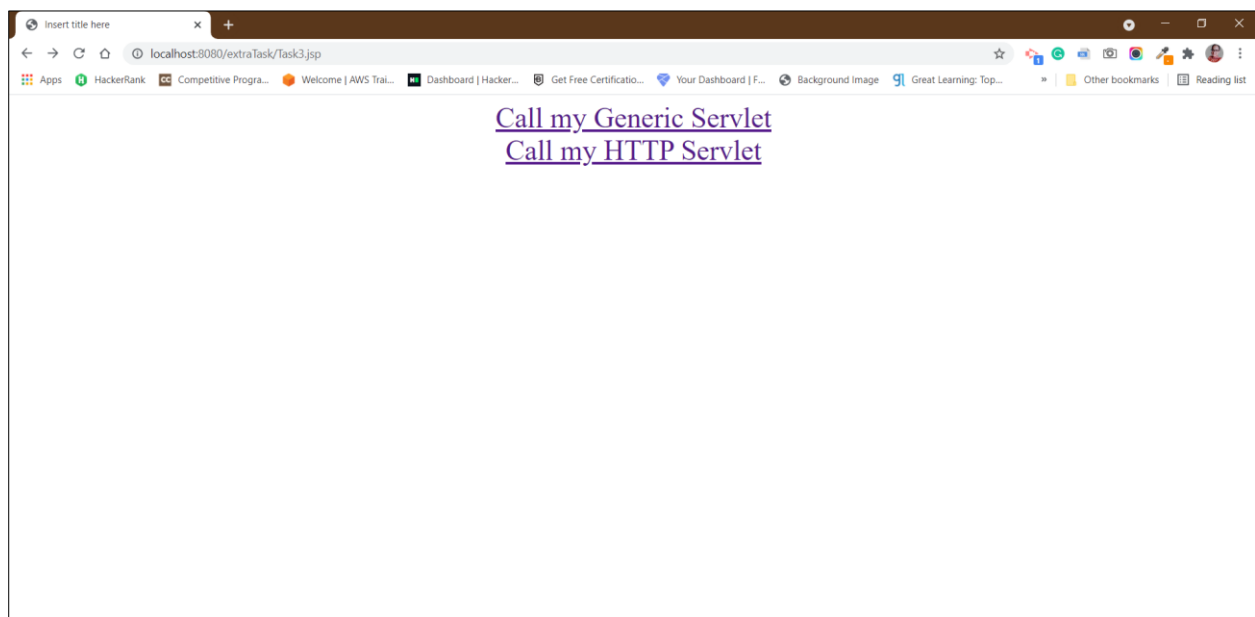
Task3.jsp

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//
//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/x
html1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-
1" />
<title>Task3</title>
</head>
<body style="font-size: 35px">
```

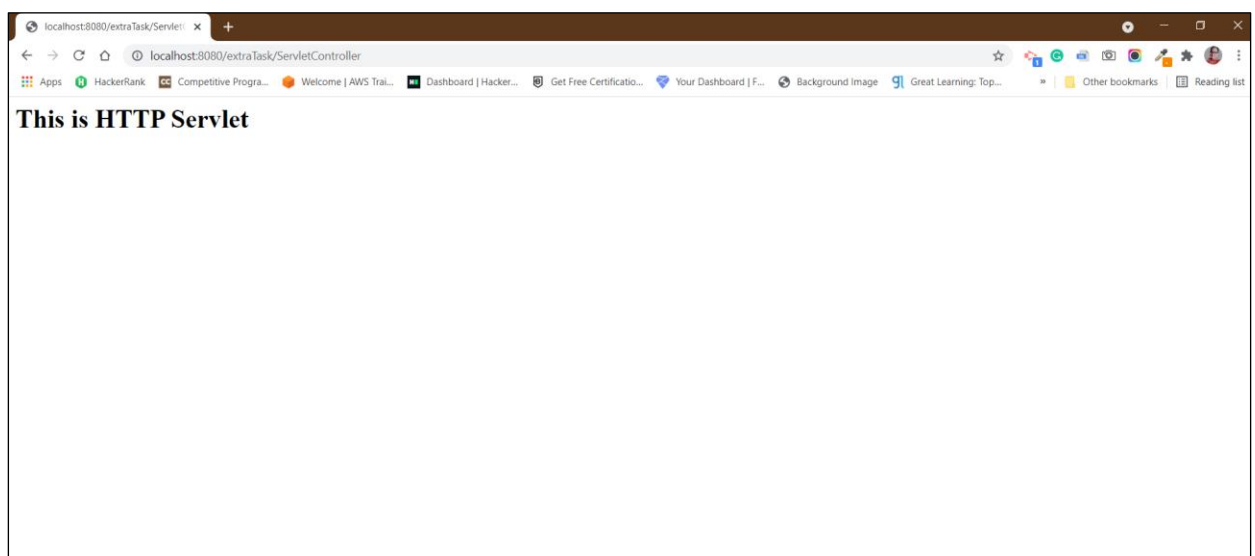
```
<center>
    <a href="GenericController">Call my Generic Servlet</a> </br>
    <a href="ServletController">Call my HTTP Servlet</a>
</center>
</body>
</html>
```

Output:

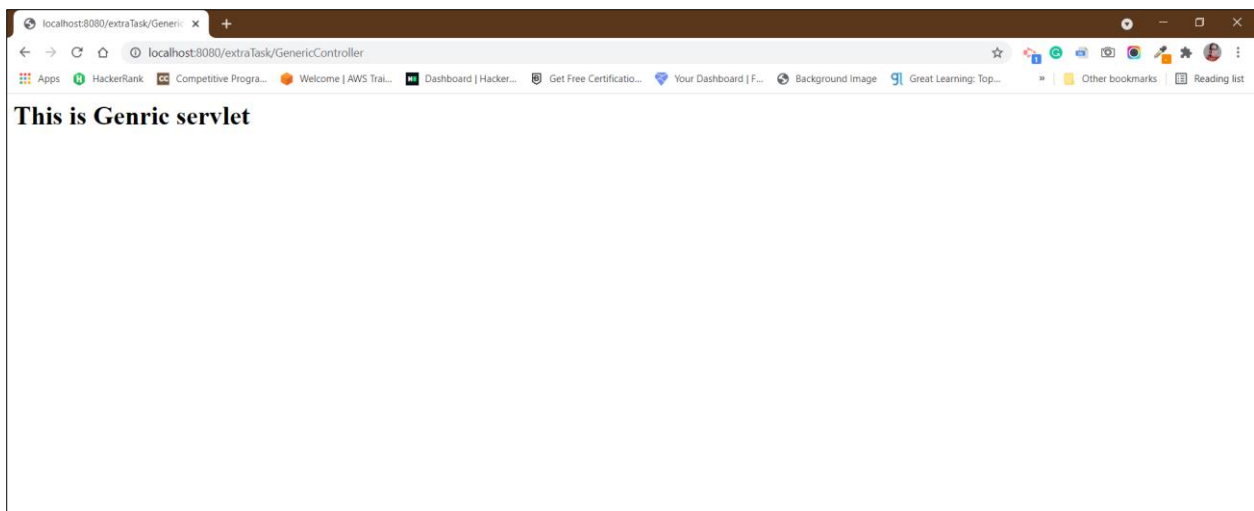
Task.jsp



ServletController



GenericController



Task:4 Create below pattern using servlet

```
      *
    2*2
  #2*2#
4#2*2#4
  #2*2#
    2*2
      *
```

```
package controller;

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;

/**
 * Servlet implementation class Task4Sevlet
 */
@WebServlet("/Task4Sevlet")
public class Task4Sevlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public Task4Sevlet() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
```

```
    * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
onse
    *     response)
    */
    protected void doGet(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {

        int size = 7, mid = size / 2;
        for (int i = 0; i < size; i++) {
            int par = Math.abs(mid - i);
            for (int j = 0; j < size; j++) {
                if (j < mid) {
                    for (int k = 0; k < mid; k++, j++) {
                        if (k < par) {

                            response.getWriter().append(" ");
                        } else {
                            printCharacter(response, k, mid);
                        }
                    }
                }
                if (j == mid) {
                    response.getWriter().append("*");
                    j++;
                }
                if (j > mid) {
                    for (int k = mid - 1; k >= 0; k--, j++) {
                        if (k < par) {
                            response.getWriter().append(" ");
                        } else
                            printCharacter(response, j, mid);
                    }
                }
            }
        }
    }
}
```



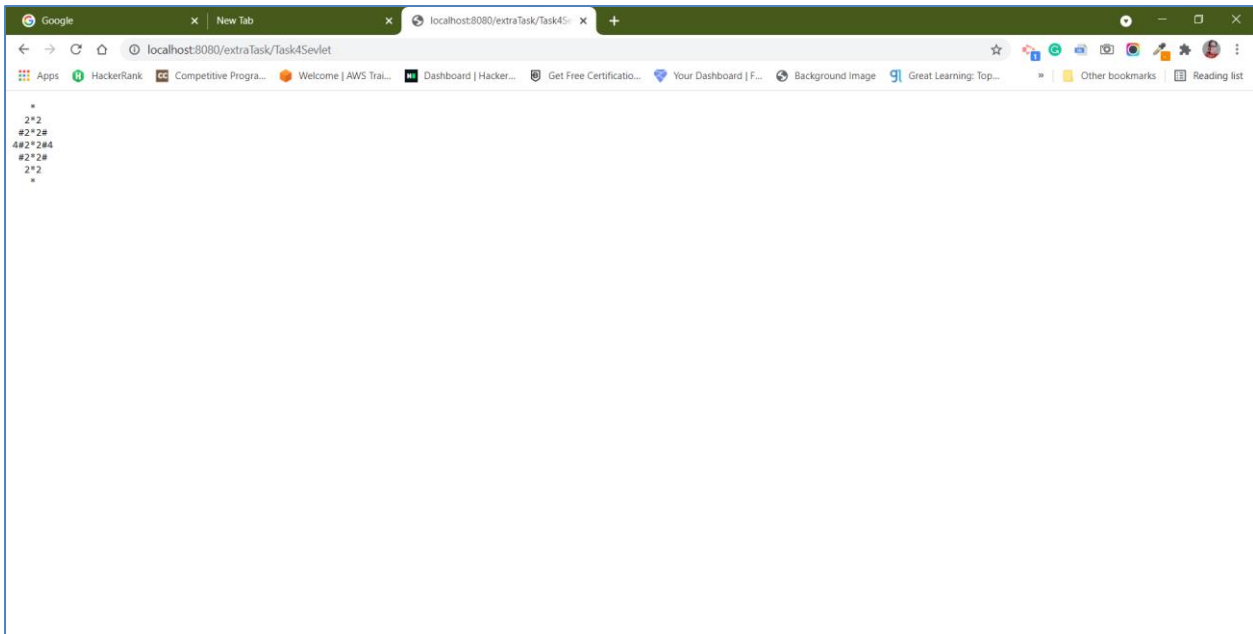
```
        }
    }
}
response.getWriter().append("\n");
}

}

private void printCharacter(HttpServletResponse response, int j, int
mid) throws IOException {
    if (j % 2 == 0) {
        response.getWriter().append("'" + (Math.abs(mid - j) + 1));
        return;
    } else if (j % 2 == 1) {
        response.getWriter().append("#");
        return;
    }

}

/**
 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletRes
ponse
 *     response)
 */
protected void doPost(HttpServletRequest request, HttpServletResponse
response)
    throws ServletException, IOException {
    // TODO Auto-generated method stub
    doGet(request, response);
}
}
```



Task-5: What is Cookie and perform a below demo using it.

❖ Cookies:

Cookies are text files stored on the client computer and they are kept for information tracking purpose. Java Servlets transparently supports HTTP cookies.

There are three steps involved in identifying returning users –

- Server script sends a set of cookies to the browser. For example name, age, or identification number etc.
- Browser stores this information on local machine for future use.
- When next time browser sends any request to web server then it sends those cookies information to the server and server uses that information to identify the user.

5.1 take below 6 input using text box

- * **5 input for number**
- * **1 for operation which will be done for that number like (add,sub,mul,div etc)**

5.2 display result of that on a second page and ask user to select any other operation to perform like (average of that no, min ,max or factorial)

Task5.jsp

```
<?xml version="1.0" encoding="ISO-8859-1" ?>  
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"  
    pageEncoding="ISO-8859-1"%>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
<title>Insert title here</title>
</head>
<body align="center">
    <h1>Task 5</h1>
    <table align="center">
        <form action="Task5Servlet" method="post">
            <h2>Enter the following details</h2>
            <tr>
                <td>Number 1 :</td>
                <td><input type="text" name="n1" placeholder="Enter a value"
                    required pattern="^[1-9]\d*$" /></td>
            </tr>
            <tr>
                <td>Number 2 :</td>
                <td><input type="text" name="n2" placeholder="Enter a value"
                    required pattern="^[1-9]\d*$" /></td>
            </tr>
            <tr>
                <td>Number 3 :</td>
                <td><input type="text" name="n3" placeholder="Enter a value"
                    required pattern="^[1-9]\d*$" /></td>
            </tr>
            <tr>
                <td>Number 4 :</td>
```

```
        <td><input type="text" name="n4" placeholder="Enter a value"
        required pattern="^[1-9]\d*$" /></td>
    </tr>
    <tr>
        <td>Number 5 :</td>
        <td><input type="text" name="n5" placeholder="Enter a value"
        required pattern="^[1-9]\d*$" /></td>
    </tr>
    <tr>
        <td>Operation name </br> (choice: add, sub, mul, div) :
        </td>
        <td><input type="text" name="operation"
        placeholder="Enter a value" required /></td>
    </tr>
    <tr>
        <td><input type="reset" name="action" value="Reset" /></td>
        <td><button type="submit" name="action" value="Calculate"
        >Find
        the Answer</button></td>
    </tr>
</form>
</table>
</body>
</html>
```

Task_1.jsp

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//
//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-
1" />
<title>Insert title here</title>
</head>
<body align="center">
    <%
        Cookie cookies[] = request.getCookies();
        String result = null, operation = null, numbers = null, oper = nu
ll, ans = null;
        for (int i = 0; i < cookies.length; i++) {
            if (cookies[i].getName().equals("result")) {
                result = cookies[i].getValue();
            }
            if (cookies[i].getName().equals("operation")) {
                operation = cookies[i].getValue();
            }
            if (cookies[i].getName().equals("numbers")) {
                numbers = cookies[i].getValue();
            }
            if (cookies[i].getName().equals("ans")) {
                ans = cookies[i].getValue();
            }
            if (cookies[i].getName().equals("oper")) {
                oper = cookies[i].getValue();
            }
        }
    %>
    <h3>
        Numbers are :
        <%=numbers%></h3>
    <h3>
        Result of
        <%=operation%>
```

```
is
<%=result%></h3>
<table align="center" style="padding: 5px;">
  <form action="Task5Servlet" method="post">
    <tr>
      <td>Select an operation:</td>
      <td><select name="operation" required>
        <option selected disabled hidden>Select one operation
to
        perform</option>
        <option value="min">Minimum</option>
        <option value="max">Maximum</option>
        <option value="avg">Average</option>
        <option value="fact">Factorial</option>
      </select></td>
    </tr>
    <tr>
      <td><input type="reset" name="action" value="Reset" /></td>
      <td><input type="submit" name="action" value="Find the Answer
" /></td>
    </tr>
  </form>
</table>
</body>
</html>
```

Task5Servlet

```
package controller;

import java.io.IOException;

import javax.servlet.RequestDispatcher;
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;
import javax.servlet.http.HttpSession;

import model.Numbers;

/**
 * Servlet implementation class Task4Servlet
 */
@WebServlet("/Task5Servlet")
public class Task5Servlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public Task5Servlet() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
onse
     *      response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
        // TODO Auto-generated method stub
        // response.getWriter().append("Served at: ").append(request.getConte
ontextPath());
        // deleteSessionCookies(request, response);
        response.sendRedirect("Task5.jsp");
    }
}
```

```
}

/**
 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
 *      response)
 */
protected void doPost(HttpServletRequest request, HttpServletResponse
response)
    throws ServletException, IOException {
    String action = request.getParameter("action");
    System.out.println("Action : " + action);
    if (action.equals("Calculate")) {

        Numbers numbers = new Numbers(request.getParameter("n1"), request.getParameter("n2"),
            request.getParameter("n3"), request.getParameter("n4"), request.getParameter("n5"));

        float result = 0;

        String operation = request.getParameter("operation").toLowerCase();
        System.out.println("Operation:" + operation);

        if (operation.equals("addition") || operation.equals("add")) {
            result = numbers.addAll();
        } else if (operation.equals("subtraction") || operation.equals("sub")) {
            result = numbers.subAll();
        } else if (operation.equals("multiplication") || operation.equals("mul")) {
            result = numbers.mulAll();
        }
    }
}
```



```
        } else if (operation.equals("division") || operation.equals("div")) {
            result = numbers.divAll();
        } else {
            deleteSessionCookies(request, response);
            response.getWriter().append("<h1>Something went wrong try again!!!</h1>");
        }

        System.out.println("result:" + result);

        Cookie resultCookie = new Cookie("result", Float.toString(result));

        Cookie operationCookie = new Cookie("operation", operation);
        System.out.println("numbers.getstring=" + numbers.getString());

        Cookie numbersCookie = new Cookie("numbers", numbers.getString());

        response.addCookie(resultCookie);
        response.addCookie(operationCookie);
        response.addCookie(numbersCookie);

        HttpSession session = request.getSession();
        session.setAttribute("numbers", numbers);
        RequestDispatcher dispatcher = request.getRequestDispatcher("Task5_1.jsp");
        dispatcher.forward(request, response);

    } else if (action.equals("Find the Answer")) {
        HttpSession session = request.getSession(false);
        Numbers numbers = (Numbers) session.getAttribute("numbers");

        if (numbers != null) {
            String operation = request.getParameter("operation");
```

```
        System.out.println("Operation:" + operation);

        switch (operation) {
            case "max":
                ans = String.valueOf(numbers.findMax());

                break;
            case "min":
                ans = String.valueOf(numbers.findMin());
                break;
            case "avg":
                ans = String.valueOf(numbers.findAvg());
                break;

            case "fact":
                ans = numbers.findFactorial();
                break;
        }

        response.getWriter().append(
            "<h1>" + numbers.getString() + "<h1><h2>Answer of
" + operation + " is : " + ans + "</h2>");

    } else {
        deleteSessionCookies(request, response);
        response.getWriter().append("<h1>Something went wrong try
again!!!</h1>");
    }
}

else {
    deleteSessionCookies(request, response);
    response.getWriter().append("<h1>Something went wrong try aga
in!!!</h1>");
}
```

```
    }  
}  
  
private void deleteSessionCookies(HttpServletRequest request, HttpServletResponse response) {  
    HttpSession session = request.getSession();  
  
    session.setAttribute("numbers", null);  
    session.invalidate();  
  
    Cookie[] cookies = request.getCookies();  
  
    for (Cookie cookie : cookies) {  
        cookie.setValue("null");  
        cookie.setMaxAge(0);  
        response.addCookie(cookie);  
    }  
}  
}
```

Numbers.jsp

```
package model;  
  
import java.util.ArrayList;  
import java.util.Collection;  
import java.util.HashMap;  
  
public class Numbers {  
    int n1, n2, n3, n4, n5;  
  
    ArrayList<Integer> numbers;
```

```
    public Numbers(String n1, String n2, String n3, String
n4, String n5) {
        numbers = new ArrayList<>();

        numbers.add(Integer.parseInt(n1));
        numbers.add(Integer.parseInt(n2));
        numbers.add(Integer.parseInt(n3));
        numbers.add(Integer.parseInt(n4));
        numbers.add(Integer.parseInt(n5));

        System.out.println("numbers:" + numbers.toString())
;
    }

    public int addAll() {

        int result = 0;
        for (int number : numbers) {
            result += number;
        }
        System.out.println("Addition:" + result);
        return result;
    }

    public int mulAll() {

        int result = 1;
        for (int number : numbers) {
            result *= number;
        }
        System.out.println("multiplication:" + result);
        return result;
    }

    public int subAll() {
```

```
    int result = 0;
    result = numbers.get(0) - numbers.get(1);
    for (int i = 2; i < numbers.size(); i++) {
        result -= numbers.get(i);
    }
    System.out.println("Substraction:" + result);
    return result;
}

public float divAll() {
    float result = 0;
    result = numbers.get(0) / numbers.get(1);
    for (int i = 2; i < numbers.size(); i++) {
        result /= (float) numbers.get(i);
    }
    System.out.println("Division:" + result);
    return result;
}

public int findMax() {

    int max = 0;

    for (int int1 : numbers) {
        if (int1 > max) {
            max = int1;
        }
    }
    System.out.println("Maximum:" + max);
    return max;
}

public int findMin() {

    int min = numbers.get(0);
```

```
        for (int int1 : numbers) {
            if (int1 < min) {
                min = int1;
            }
        }
        System.out.println("minimum:" + min);
        return min;
    }

    public float findAvg() {
        int sum = 0;

        for (int int1 : numbers) {
            sum += int1;
        }
        System.out.println(("Average:" + (float) sum / (float) numbers.size()));
        return (float) sum / (float) numbers.size();
    }

    public String findFactorial() {

        int fact = 0;
        HashMap<Integer, Integer> factorials = new HashMap<>();

        for (int int1 : numbers) {
            fact = fact(int1);

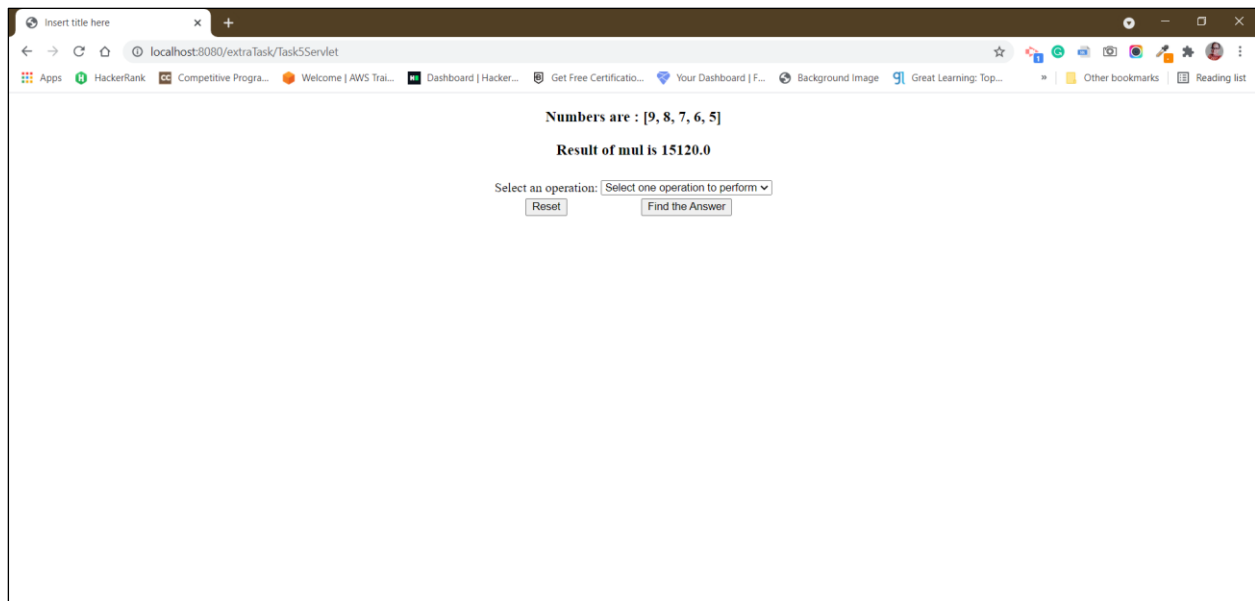
            factorials.put(int1, fact);
        }
        System.out.println("factorial : " + factorials.toString());
        return factorials.toString();
    }
}
```

```
}

private int fact(int no) {
    int ans = 1;
    for (int i = no; i > 0; i--) {
        ans *= i;
    }
    System.out.println("factorial of " + no + " is " +
ans);
    return ans;
}

public String getString() {
    return numbers.toString();
}
}
```

The screenshot shows a web browser window with the address bar displaying 'localhost:8080/extraTask/Task5.jsp'. The browser's bookmark bar includes links to 'Apps', 'HackerRank', 'Competitive Progra...', 'Welcome | AWS Trail...', 'Dashboard | Hacker...', 'Get Free Certificatio...', 'Your Dashboard | F...', 'Background Image', 'Great Learning: Top...', 'Other bookmarks', and 'Reading list'. The main content area of the browser displays a form titled 'Task 5' with the instruction 'Enter the following details'. The form contains five input fields labeled 'Number 1 :', 'Number 2 :', 'Number 3 :', 'Number 4 :', and 'Number 5 :', each containing the values 9, 8, 7, 6, and 5 respectively. Below these is a label 'Operation name (choice: add, sub, mul, div):' followed by a dropdown menu currently showing 'mul'. At the bottom of the form are two buttons: 'Reset' and 'Find the Answer'.



Task-6: What is HTTP Session and perform a demo of it using login and logout.

HttpSession object is used to store entire session with a specific client. We can store, retrieve and remove attribute from HttpSession object. Any servlet can have access to HttpSession object throughout the getSession() method of the HttpServletRequest object

6.1 User will first login after that enter the marks of 7 subject and student name submit it.

6.2 use send redirect in to redirect user to result page.

6.3 on result page show the average of that marks and whether he cleared or not along with an college logo and name of the college and student as well.

6.4 store the student marks in database as well.

Task6.jsp

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
<title>Task 6</title>
</head>
<body align="center">
    <h1>Task 6</h1>
    <h3>Login</h3>
    <table align="center">
        <form action="Task6Servlet" method="post">
            <tr>
                <td>Email :</td>
                <td><input type="text" name="email" required /></td>
            </tr>
            <tr>
                <td>Password :</td>
                <td><input type="password" name="password" required /></td>
            </tr>
            <tr colspan="2">
                <input type="submit" value="Login" />
            </tr>
        </form>
    </table>
</body>
</html>
```

```
                <td><input type="submit" name="action" value="login" required /></td>
            </tr>
        </form>
    </table>
</body>
</html>
```

Task6_1.jsp

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
<title>Insert title here</title>
</head>
<body align="center">
    <h1>Task 6 :</h1>
    <h4>Enter the follwong details:</h4>
    <table align="center">
        <form action="Task6Servlet" method="post">
            <tr>
                <td>Name</td>
                <td><input type="text" name="name" placeholder="enter your marks" /></td>
            </tr>
            <tr>
                <td>Subject1</td>
                <td><input type="text" name="s1" placeholder="enter your marks" /></td>
```

```
        </tr>
        <tr>
            <td>Subject2</td>
            <td><input type="text" name="s2" placeholder="enter your mark
s" /></td>
        </tr>
        <tr>
            <td>Subject3</td>
            <td><input type="text" name="s3" placeholder="enter your mark
s" /></td>
        </tr>
        <tr>
            <td>Subject4</td>
            <td><input t4ype="text" name="s4" placeholder="enter your mar
ks" /></td>
        </tr>
        <tr>
            <td>Subject5</td>
            <td><input type="text" name="s5" placeholder="enter your mark
s" /></td>
        </tr>
        <tr>
            <td>Subject6</td>
            <td><input type="text" name="s6" placeholder="enter your mark
s" /></td>
        </tr>
        <tr>
            <td>Subject7</td>
            <td><input type="text" name="s7" placeholder="enter your mark
s" /></td>
        </tr>
        <tr>
            <td></td>
            <td><input type="submit" name="action" value="Calculate Resul
t" /></td>
```

```
        </tr>
    </form>
</table>
</body>
</html>
```

Task6_3.jsp

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<%@page import="model.Marks"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
<title>Insert title here</title>
</head>
<body align="center">
    <%
        Marks marks = null;
        HttpSession httpSession = request.getSession(false);
        marks = (Marks) httpSession.getAttribute("marks");
        String status = (String) httpSession.getAttribute("status");
        if (marks == null) {
            response.sendRedirect("Task6Servlet?action=logout");
        }
    %>
    <table align="center">
        <tr>
            <td></td>
            <td>Aditya Silver Oak Institute Of technology</td>
        </tr>
        <tr>
```

```
        <td colspan="1">Result</td>
    </tr>
    <tr>
        <td>Name:</td>
        <td><%=marks.getName()%></td>
    </tr>
    <tr>
        <td>Subject 1 :</td>
        <td><%=marks.getS1()%></td>
    </tr>
    <tr>
        <td>Subject 2 :</td>
        <td><%=marks.getS2()%></td>
    </tr>
    <tr>
        <td>Subject 3 :</td>
        <td><%=marks.getS3()%></td>
    </tr>
    <tr>
        <td>Subject 3 :</td>
        <td><%=marks.getS3()%></td>
    </tr>
    <tr>
        <td>Subject 4 :</td>
        <td><%=marks.getS4()%></td>
    </tr>
    <tr>
        <td>Subject 5 :</td>
        <td><%=marks.getS5()%></td>
    </tr>
    <tr>
        <td>Subject 6 :</td>
        <td><%=marks.getS6()%></td>
    </tr>
    <tr>
```

```
        <td>Subject 7 :</td>
        <td><%=marks.getS7()%></td>
    </tr>
    <tr>
        <td>Average Marks</td>
        <td><%=marks.getAvg()%></td>
    </tr>
    <tr>
        <td>Percentage</td>
        <td><%=marks.getPercentage()%></td>
    </tr>
    <tr>
        <td>Pass or Fail!!</td>
        <td><%=status%></td>
    </tr>
    <tr>
        <td colspan="1"><a href="Task6Servlet?action=logout">Logout</a></td>
    </tr>
</table>
</body>
</html>
```

Marks.java

```
package model;

public class Marks {
    private int s1;
    private int s2;
    private int s3;
    private int s4;
    private int s5;
    private int s6;
    private int s7;
    private float avg;
```

```
private float percentage;  
private float total;  
private String name;  
  
public int getS1() {  
    return s1;  
}  
  
public void setS1(int s1) {  
    this.s1 = s1;  
}  
  
public int getS2() {  
    return s2;  
}  
  
public void setS2(int s2) {  
    this.s2 = s2;  
}  
  
public int getS3() {  
    return s3;  
}  
  
public void setS3(int s3) {  
    this.s3 = s3;  
}  
  
public int getS4() {  
    return s4;  
}  
  
public void setS4(int s4) {  
    this.s4 = s4;  
}
```

```
public int getS5() {  
    return s5;  
}  
  
public void setS5(int s5) {  
    this.s5 = s5;  
}  
  
public int getS6() {  
    return s6;  
}  
  
public void setS6(int s6) {  
    this.s6 = s6;  
}  
  
public int getS7() {  
    return s7;  
}  
  
public void setS7(int s7) {  
    this.s7 = s7;  
}  
  
private void setTotal() {  
    this.total = this.s1 + this.s2 + this.s3 + this.s4 + this.s5 + this.s6 + this.s7;  
}  
  
public float getAvg() {  
    return avg;  
}
```



```
public void setAvg() {
    setTotal();
    this.avg = this.total / (float) 7;
}

public float getPercentage() {
    return percentage;
}

public void setPercentage() {
    this.percentage = this.total / 700 * 100;
}

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

@Override
public String toString() {
    return "Marks [s1=" + s1 + ", s2=" + s2 + ", s3=" + s3 + ", s4="
+ s4 + ", s5=" + s5 + ", s6=" + s6 + ", s7="
        + s7 + ", avg=" + avg + ", percentage=" + percentage + ",
name=" + name + " ]";
}
}
```

Student.java

```
package model;

public class Student {
    private int id;
```

```
private String email;
private String password;
private String name;

public int getId() {
    return id;
}

public void setId(int id) {
    this.id = id;
}

public String getEmail() {
    return email;
}

public void setEmail(String email) {
    this.email = email;
}

public String getPassword() {
    return password;
}

public void setPassword(String password) {
    this.password = password;
}

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}
```

```
@Override
public String toString() {
    return "Student [id=" + id + ", email=" + email + ", password=" +
password + ", name=" + name + "]";
}
}
```

Task6Manager.java

```
package dao;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

import model.Marks;
import model.Student;

public class Task6Manager {
    Connection conn;
    Statement statement;
    ResultSet resultSet;

    public Task6Manager() {
        // TODO Auto-generated constructor stub
        conn = DB_connection.getConnection();
    }

    public Student getUserByEmail(String email) {

        System.out.println("email:" + email);
        try {
```

```
        statement = conn.createStatement();
        resultSet = statement.executeQuery("Select * from student where email = '" + email + "'");
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    Student student = null;
    try {
        while (resultSet.next()) {

            student = new Student();
            student.setId(resultSet.getInt("id"));
            student.setName(resultSet.getString("name"));

            student.setEmail(resultSet.getString("email"));
            student.setPassword(resultSet.getString("password"));

            System.out.println("student object :" + student.toString());
        }
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return student;
}

public String checkPassOrNot(Marks marks) {
    if (marks.getPercentage() > 33) {
        return "pass";
    }
    return "fail";
}
```

```
public boolean insert(Marks marks) {
    try {
        statement = conn.createStatement();

        int result = statement.executeUpdate(
            "INSERT into marks (`s1`,`s2`,`s3`,`s4`,`s5`,`s6`,`s7`,"
            + "`avg`,`percent`,`name`) values('"
            + marks.getS1() + "','" + marks.getS2() + "','"
            + marks.getS3() + "','" + marks.getS4()
            + "','" + marks.getS5() + "','" + marks.getS6()
            + "','" + marks.getS7() + "','"
            + marks.getAvg() + "','" + marks.getPercentage()
            + "','" + marks.getName() + "');"
        );
        if (result > 0) {
            return true;
        }

    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return false;
}
```

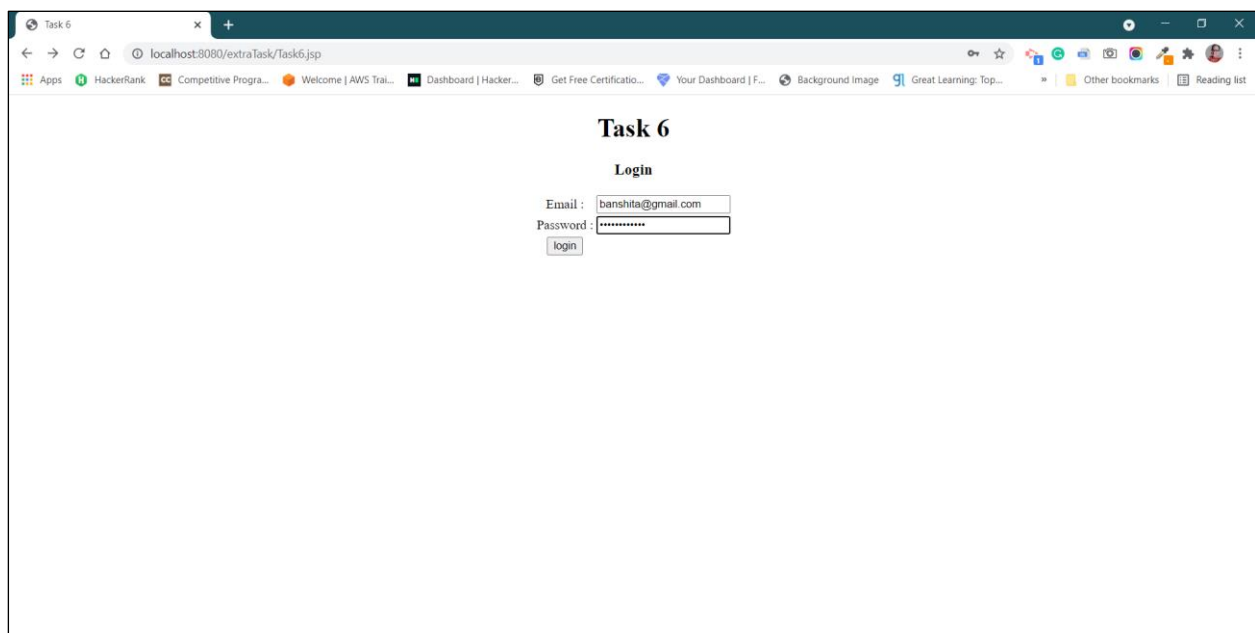
DB_connection.java

```
package dao;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DB_connection {
    public static Connection conn;
```

```
public static Connection getConnection() {  
  
    try {  
        Class.forName("com.mysql.cj.jdbc.Driver");  
        conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/xtratask ", "root", "root");  
    } catch (ClassNotFoundException e) {  
        // TODO Auto-generated catch block  
        e.printStackTrace();  
    } catch (SQLException e) {  
        // TODO Auto-generated catch block  
        e.printStackTrace();  
    }  
    return conn;  
}  
}
```



The screenshot shows a web browser window with the title 'Task 6'. The address bar displays 'localhost:8080/extraTask/Task6.jsp'. The browser's bookmark bar includes 'Apps', 'HackerRank', 'Competitive Progra...', 'Welcome | AWS Tral...', 'Dashboard | Hacker...', 'Get Free Certificatio...', 'Your Dashboard | F...', 'Background Image', 'Great Learning: Top...', 'Other bookmarks', and 'Reading list'. The main content area of the browser displays a login form titled 'Task 6' with a sub-header 'Login'. The form contains two input fields: 'Email :' with the value 'banshita@gmail.com' and 'Password :' with masked characters '*****'. Below these fields is a 'login' button.

Insert title here

localhost:8080/extraTask/Task6_1.jsp

AppsHackerRankCompetitive Progra...Welcome | AWS Tra...Dashboard | Hacker...Get Free Certificatio...Your Dashboard | F...Background ImageGreat Learning: Top...Other bookmarksReading list

Task 6 :

Enter the follwong details:

NameBanshita

Subject178

Subject289

Subject345

Subject478

Subject565

Subject689


Subject745

Calculate Result

Insert title here

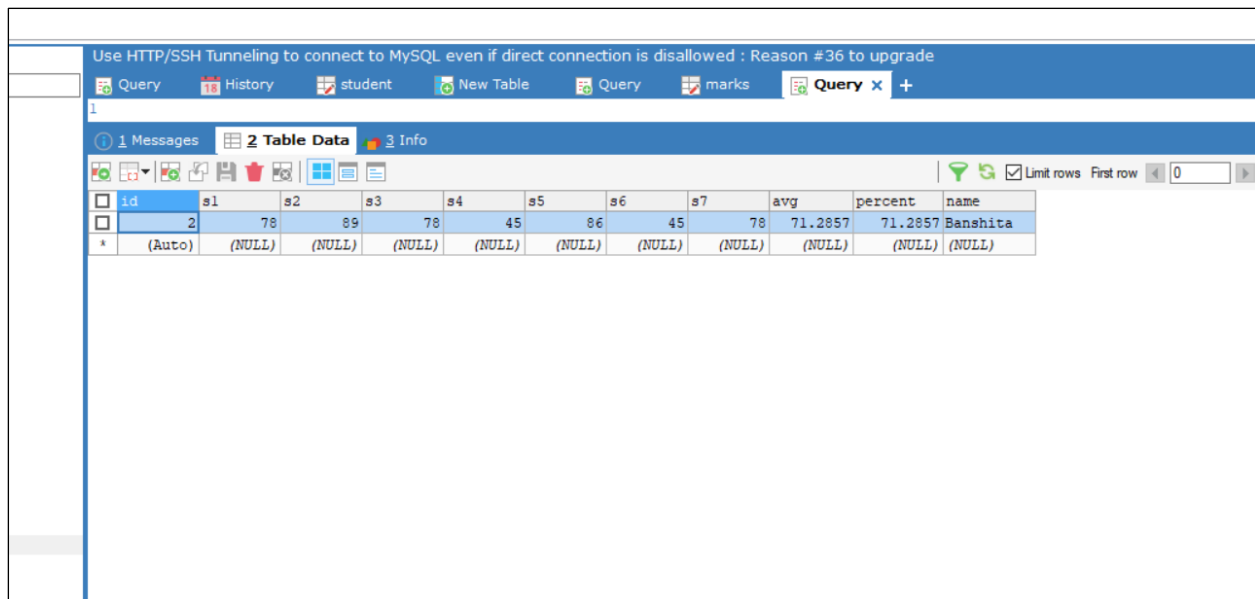
localhost:8080/extraTask/Task6_3.jsp

AppsHackerRankCompetitive Progra...Welcome | AWS Tra...Dashboard | Hacker...Get Free Certificatio...Your Dashboard | F...Background ImageGreat Learning: Top...Other bookmarksReading list



Aditya Silver Oak Institute Of technology

Result	
Name:	Banshita
Subject 1 :	78
Subject 2 :	89
Subject 3 :	45
Subject 3 :	45
Subject 4 :	78
Subject 5 :	65
Subject 6 :	89
Subject 7 :	45
Average Marks	69.85714
Percentage	69.85715
Pass or Fail!!	pass
Logout	



id	s1	s2	s3	s4	s5	s6	s7	avg	percent	name
2	78	89	78	45	86	45	78	71.2857	71.2857	Banshita
*	(Auto)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

Task-7: Create a forgot password and generate random highly secure password for that user based on their email-id.

7.1 on forgot password ask user to enter email-id and answer of 4 different security questions.

7.2 if email id matches with existing records then display newly generator password on jsp page.

Task7.jsp

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
<title>Insert title here</title>
</head>
<body align="center">
    <h1>Task 7</h1>
    <h3>Login</h3>
    <table align="center">
```



```
<form action="Task7Servlet" method="post">
  <tr>
    <td>Email :</td>
    <td><input type="text" name="email" required /></td>
  </tr>
  <tr>
    <td>Password :</td>
    <td><input type="password" name="password" required /></td>
  </tr>
  <tr>
    <td><a href="Task7_2.jsp">Forgot password</a></td>
    <td><input type="submit" name="action" value="login" required /></td>
  </tr>
</form>
</table>
</body>
</html>
```

Task7_2.jsp

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
<title>Insert title here</title>
</head>
```

```
<body align="center">
    <h1>Task 7</h1>
    <h3>Forget Password</h3>
    <h5>Oppss!! you forgot your password.. No worries we will help you</h5>
</body>

<table align="center">
    <form action="Task7Servlet" method="post">
        <tr>
            <td>Enter Email :</td>
            <td><input type="text" name="email" required /></td>
        </tr>
        <tr>
            <td></td>
            <td><Button type="submit" name="action" value="Check Email" >Submit</Button></td>
        </tr>
    </form>
</table>
</body>
</html>
```

Task_3.jsp

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
<title>Insert title here</title>
</head>
<body align="center">
    <h1>Task 7</h1>
```

```
<h3>Register</h3>
<table align="center" style="text-align: center;">
  <form action="Task7Servlet" method="post">
    <tr>
      <td><input type="hidden" name="email" value="<%=request.g
getParameter("email") %>" /></td>
    </tr>
    <td colspan="1">Answer these question :</td>
  </tr>
  <tr>
    <td>How is your Favorite Teacher? :</td>
    <td><input type="text" name="q1" required /></td>
  </tr>
  <tr>
    <td>Name one monument you want to visit :</td>
    <td><input type="text" name="q2" required /></td>
  </tr>
  <tr>
    <td>Name the place where you were born :</td>
    <td><input type="text" name="q3" required /></td>
  </tr>
  <tr>
    <td>Your favorite book :</td>
    <td><input type="text" name="q4" required /></td>
  </tr>
  <tr colspan="2">
    <td><input type="submit" name="action" value="Submit Answer"
/></td>
  </tr>
</form>
</table>
</body>
</html>
```

User.java

```
package model;

public class User {
    int id;
    String name;
    String email;
    String password;
    String q1;
    String q2;
    String q3;
    String q4;

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
```

```
        this.email = email;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    public String getQ1() {
        return q1;
    }

    public void setQ1(String q1) {
        this.q1 = q1;
    }

    public String getQ2() {
        return q2;
    }

    public void setQ2(String q2) {
        this.q2 = q2;
    }

    public String getQ3() {
        return q3;
    }

    public void setQ3(String q3) {
        this.q3 = q3;
    }
}
```

```
    public String getQ4() {
        return q4;
    }

    public void setQ4(String q4) {
        this.q4 = q4;
    }

    @Override
    public String toString() {
        return "User [id=" + id + ", name=" + name + ", email=" + email +
            ", password=" + password + ", q1=" + q1
                + ", q2=" + q2 + ", q3=" + q3 + ", q4=" + q4 + "]";
    }
}
```

UserManager.java

```
package dao;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Random;

import model.User;

public class UserManager {

    Connection conn;
    Statement statement;
    ResultSet resultSet;

    public UserManager() {
        // TODO Auto-generated constructor stub
        conn = DB_connection.getConnection();
    }
}
```

```
}

public User getUserByEmail(String email) {

    System.out.println("email:" + email);
    try {
        statement = conn.createStatement();
        resultSet = statement.executeQuery("Select * from user where
email = '" + email + "'");
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    User user = null;
    try {
        while (resultSet.next()) {
            user = new User();
            user.setId(resultSet.getInt("id"));
            user.setEmail(resultSet.getString("email"));
            user.setName(resultSet.getString("name"));
            user.setPassword(resultSet.getString("password"));
            user.setQ1(resultSet.getString("q1"));
            user.setQ2(resultSet.getString("q2"));
            user.setQ3(resultSet.getString("q3"));
            user.setQ4(resultSet.getString("q4"));
        }
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return user;
}

public boolean insert(User user) {
```

```
        try {
            statement = conn.createStatement();

            int result = statement
                .executeUpdate("INSERT into user ('name','email','password','q1','q2','q3','q4') values('"
                    + user.getName() + "','" + user.getEmail() +
                    "','" + user.getPassword() + "','"
                    + user.getQ1() + "','" + user.getQ2() + "','"
                    + user.getQ3() + "','" + user.getQ4() + "')");
            if (result > 0) {
                return true;
            }

        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        return false;
    }

    public String generatePassword() {

        StringBuilder builder = new StringBuilder();
        String alphanumeric = "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789abcdefghijklmnopqrstuvwxyz";

        Random random = new Random();
        for (int i = 0; i < 8; i++) {
            int index = random.nextInt(alphanumeric.length());

            builder.append(alphanumeric.charAt(index));
        }
        return builder.toString();
    }
}
```



```
}

public User getUserById(int id) {

    System.out.println("email:" + id);
    try {
        statement = conn.createStatement();
        resultSet = statement.executeQuery("Select * from user where
id = '" + id + "'");
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    User user = null;
    try {
        while (resultSet.next()) {
            user = new User();
            user.setId(resultSet.getInt("id"));
            user.setEmail(resultSet.getString("email"));
            user.setName(resultSet.getString("name"));
            user.setPassword(resultSet.getString("password"));
            user.setQ1(resultSet.getString("q1"));
            user.setQ2(resultSet.getString("q2"));
            user.setQ3(resultSet.getString("q3"));
            user.setQ4(resultSet.getString("q4"));
            System.out.println(user.toString());
        }
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return user;
}
```

```
public User updatePassword(String password, int id) {

    try {
        statement = conn.createStatement();

        statement.executeUpdate("Update user set password='" + password + "' where id=" + id + "");
        return getUserById(id);
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return null;
}
```

Task7Servlet

```
package controller;

import java.io.IOException;

import javax.servlet.RequestDispatcher;
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 dao.UserManager;
import model.User;

/**
 * Servlet implementation class Task7Servlet
 */
```

```
@WebServlet("/Task7Servlet")
public class Task7Servlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public Task7Servlet() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
onse
     *      response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
        // TODO Auto-generated method stub
        response.getWriter().append("Served at: ").append(request.getCont
extPath());
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
ponse
     *      response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
        // TODO Auto-generated method stub
        String action = request.getParameter("action");
```

```
System.out.println("Action:" + action);

if (action.equals("Check Email")) {

    UserManager manager = new UserManager();

    User user = manager.getUserByEmail(request.getParameter("email"));

    if (user != null) {

        request.setAttribute("email", user.getEmail());
        RequestDispatcher dispatcher = request.getRequestDispatcher("Task7_3.jsp");
        dispatcher.forward(request, response);
    } else {
        response.getWriter().append("<h1>Invalid email id try again</h1>");
    }

} else if (action.equals("Submit Answer")) {
    UserManager manager = new UserManager();

    User user = manager.getUserByEmail(request.getParameter("email"));

    if (user.getQ1().equals(request.getParameter("q1")) && user.getQ2().equals(request.getParameter("q2"))
        && user.getQ2().equals(request.getParameter("q2"))
        && user.getQ2().equals(request.getParameter("q2"))) {

        User updateUser = manager.updatePassword(manager.generatePassword(), user.getId());
        if (updateUser != null) {
```

```
        response.getWriter().append("<h1>We update your password!!</h1>")

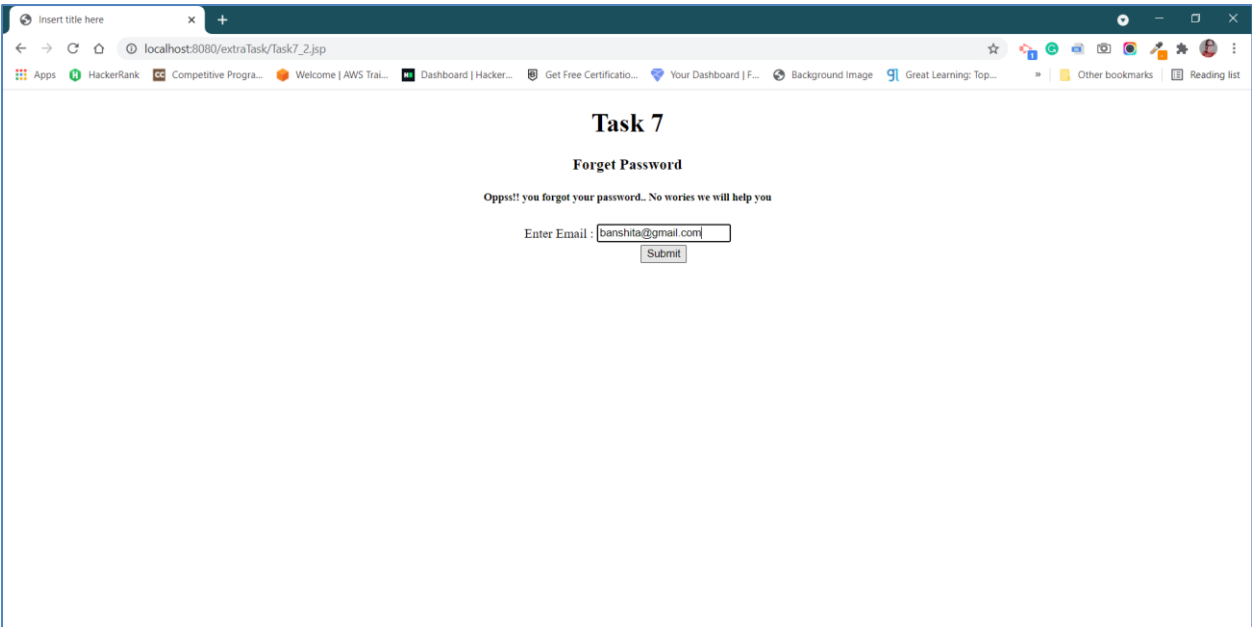
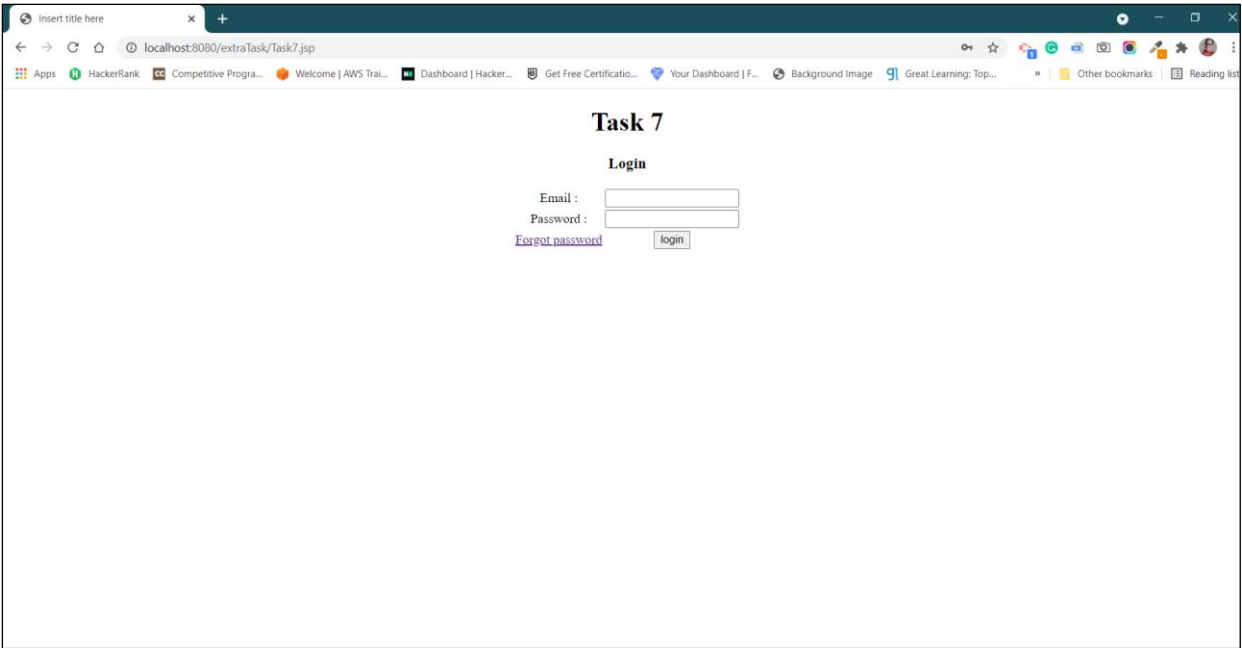
        + "<h3>Don't forget this time</h3>"
        + "<h5>Password : "+updateUser.getPassword()+"</h5>");
    } else {
        response.getWriter().append("Oppss... something went wrong try again");
    }
}

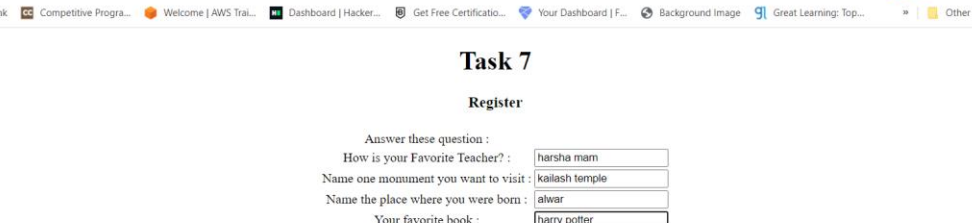
}

}
```

The screenshot shows the SQLyog Community 64 interface. The left sidebar displays a tree view of the database structure for 'localhost'. The main window shows a table view of a database, likely 'user', with columns: id, email, name, password, q1, q2, q3, and q4. The table contains one row of data for a user named 'Banshita Gangwar'.

id	email	name	password	q1	q2	q3	q4
1	banshita@gmail.com	Banshita Gangwar	banshita@12	harsha mam	kailash temple	alwar	harry potter
*	(Auto)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)





The screenshot shows a web browser window with the address bar displaying 'localhost:8080/extraTask/Task7Servlet'. The browser's bookmark bar is visible, showing various links like 'Apps', 'HackerRank', 'Competitive Progra...', 'Welcome | AWS Tral...', 'Dashboard | Hacker...', 'Get Free Certificatio...', 'Your Dashboard | F...', 'Background Image', 'Great Learning: Top...', 'Other bookmarks', and 'Reading list'. The main content of the page is a registration form titled 'Task 7' with the subtitle 'Register'. The form asks for the user's favorite teacher, a monument to visit, the place where they were born, and their favorite book. Each question has a corresponding text input field. The first three fields contain the text 'harsha mam', 'kailash temple', and 'ahwar' respectively. The fourth field contains 'harry potter'. A 'Submit Answer' button is located at the bottom of the form.

Insert title here

localhost:8080/extraTask/Task7Servlet

Apps HackerRank Competitive Progra... Welcome | AWS Tral... Dashboard | Hacker... Get Free Certificatio... Your Dashboard | F... Background Image Great Learning: Top... Other bookmarks Reading list

Task 7

Register

Answer these question :

How is your Favorite Teacher? :

Name one monument you want to visit :

Name the place where you were born :

Your favorite book :

A screenshot of a web browser window. The address bar shows the URL 'localhost:8080/extraTask/Task7Servlet'. The browser's tab bar at the top shows several open tabs, including 'localhost:8080/extraTask/Task7Servlet'. The browser's bookmark bar is visible below the address bar, showing various bookmarks like 'Apps', 'HackerRank', 'Competitive Progra...', 'Welcome | AWS Trai...', 'Dashboard | Hacker...', 'Get Free Certificatio...', 'Your Dashboard | F...', 'Background image', 'Great Learning: Top...', 'Other bookmarks', and 'Reading list'. The browser interface is in dark mode.

We update your password!!

Don't forget this time

Password :iJVjUQBb

