Name: Raghav Maheshwari
RollNO: 53
Panel: A

Lab Assignment - 12 (JSP)

★ Aim:
Develop Web Application for Employee Registeration using JSP Servlet & JDBC.

★ Objectives:
→ To understand server side scripting
→ To learn database connectivity using mySQL.
→ To learn working of servlet & JSP.

★ Theory:

→ Servlet:
Servlet is small Java program that runs within Web server servlets receive and respond from web clients, usually HTTP to implement this interface, you can write generic servlet that extends javax servlet.
Servlet: http. HttpServlet.

→ JSP:
JSP is technology which is used to create web Application just like servlet technology. It can be thought as extension to servlet because it provides more functionality than servlet such as expression language JSTL etc.

→ Database Connectivity:

Database connectivity is facility in computer science that allows client software to talk to database server software, whether on same machine or not. A connection is required to send commands and receive answers.

→ Platform: · JSP - 2.2+

- IDE → STS/Eclipse Neon 3
- JDK → 1.8 or later
- Apache Tomcat - 8.5
- JSTL - 1.2.1
- Servlet API - 2.5
- MySQL - mySQL - connector - java- 8.0.13 jar.

→ Conclusion:

Thus, studied and implemented servlets, JSP and database connectivity.

FAQ

1. What is difference b/w servlet and JSP?

Ans

Servlet	JSP
→ Servlet is javacode.	→ JSP is HTML code.
→ Servlet plays controller role in has MVC approach	→ JSP is view in MVC approach for storing output.
→ Servlet is faster than JSP.	→ JSP is slower cause it has to translate JSP to java.
→ Servlet can accept all protocol requests	→ JSP only accept HTTP requests.

2. List the advantages of using JSP.

Ans Dynamic part is written in java, not visual basic or MS specific language. So it is more powerful and easy to use second it is portable to other operating systems.

3. What is difference b/w GET and POST methods?

Ans GET carries request parameter appended in URL string while POST carries parameter message body.

4. Compare Generic Servlet and HTTP servlet.

Ans

Generic servlet	Http Servlet
→ Protocol independent.	→ Protocol dependent.
→ Service is method abstract.	→ Service is method non abstract.
→ Immediate subclass of servlet interface.	→ Immediate subclass of generic servlet.

5. What are major components of JDBC

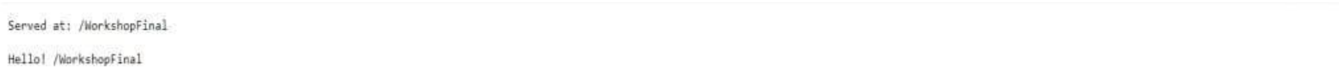
Ans → Drive Manager
→ Driver
→ Connection
→ Statement

Seen

Code 1(Servlet):

```
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 Hello

 */

@WebServlet("/Hello")

public class Hello extends HttpServlet {

        private static final long serialVersionUID = 1L;


   /**

    * @see HttpServlet#HttpServlet()

    */

   public Hello() {

      super();

      // TODO Auto-generated constructor stub

   }


        /**

         * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

         */
	protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
		// TODO Auto-generated method stub
		response.getWriter().append("Served at: ").append(request.getContextPath());
		response.getWriter().append("\n\nHello! ").append(request.getContextPath());
	}

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

}
```

Output:



```
Served at: /WorkshopFinal

Hello! /WorkshopFinal
```

Code 2(JSP):

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<p>
Hello World
</p>
</body>
</html>
```

Output:

```
Hello World
```

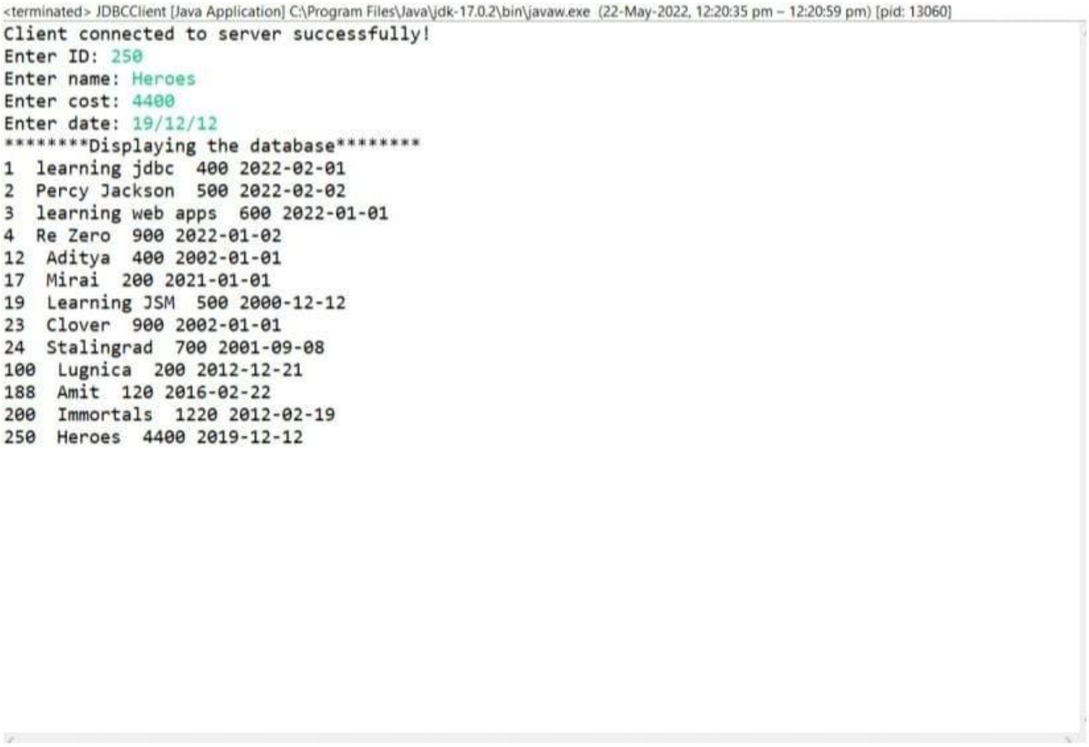
Code 3(JDBC):

```
package mitJDBC;


import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.*;
public class JDBCClient {


	public static void main(String[] args) throws ClassNotFoundException, SQLException {
		// TODO Auto-generated method stub
```

```
    Class.forName("com.mysql.cj.jdbc.Driver");
                        String jdbcurl="jdbc:mysql://localhost:3306/javaworkshop"; //made a URL
                        Connection con=DriverManager.getConnection(jdbcurl, "root", "Aditya!@#$12"); //make a socket connection for URL
                        if(con!=null)
                            System.out.print("Client connected to server successfully!");
                        //Create a statement
                        String query="insert into book values(?,?,?,?)";
                        Scanner sc=new Scanner(System.in);
                        PreparedStatement pstmt=con.prepareStatement(query); //PreparedStatement is an interface
                        System.out.print("\nEnter ID: ");
                        pstmt.setInt(1,sc.nextInt());
                        System.out.print("Enter name: ");
                        pstmt.setString(2,sc.next());
                        System.out.print("Enter cost: ");
                        pstmt.setInt(3, sc.nextInt());
                        System.out.print("Enter date: ");
                        pstmt.setString(4,sc.next());
                        pstmt.executeUpdate(); //fire the query

                        Statement stmt= con.createStatement();
                        String query2="select * from book";
                        ResultSet rs=stmt.executeQuery(query2);
```

```
                                        System.out.print("********Displaying the
database********\n");

                                        while(rs.next())
                                        {
                                              int id=rs.getInt(1);
                                              String name=rs.getString(2);
                                              int cost=rs.getInt(3);
                                              java.sql.Date date=rs.getDate(4);

                                              System.out.println(id+"  "+name+"  "+cost+"
"+date);

                                        }
                                        sc.close();
                                      }

}
```

Output:



<terminated> JDBCClient [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\javaw.exe (22-May-2022, 12:20:35 pm – 12:20:59 pm) [pid: 13060]

```
Client connected to server successfully!
Enter ID: 250
Enter name: Heroes
Enter cost: 4400
Enter date: 19/12/12
********Displaying the database********
1  learning jdbc  400 2022-02-01
2  Percy Jackson  500 2022-02-02
3  learning web apps  600 2022-01-01
4  Re Zero  900 2022-01-02
12  Aditya  400 2002-01-01
17  Mirai  200 2021-01-01
19  Learning JSM  500 2000-12-12
23  Clover  900 2002-01-01
24  Stalingrad  700 2001-09-08
100  Lugnica  200 2012-12-21
188  Amit  120 2016-02-22
200  Immortals  1220 2012-02-19
250  Heroes  4400 2019-12-12
```