1 Experiment: JDBC Programs using Statement.

1.1 To write a java program using JDBC to create table and perform insert, update and delete data using Oracle.

2 Experiment: JDBC Programs using Prepared Statement.

2.1 A program to insert a record and select records.

3 Experiment : Servlet Programming

3.1 A servlet program to display request details.
3.2 A servlet program to handle user form.
3.3 A servlet program to create & display cookie.
3.4 A servlet program to do session tracking.

4 Experiment : JSP Programming

4.1 JSP program to display hello world.
4.2 JSP program to demonstrate the Cookie.

5.Write a program in Java to implement a Client/Server application using RMI.

EX.NO:1

JDBC Program Using Statement

Aim: To write a java program using JDBC to create table and perform insert, update and delete data using Oracle.

Algorithm:

Step1:Start the program.
Step 2:Include packages java.io and java.sql.
Step 3:Define class with name "jdbc" and define the main function.
Step 4:Declare objects for Connection, Statement, ResultSet and alsodeclare the object for BufferedReader class.
Step 5:Declare local variables ch,rno,n as integer and na as String.
Step 6:Register the JdbcOdbcDriver and make a connection using getConnection() by giving Data Source Name "ORCL".
Step 7:
Define switch
case 1 for insert records,
case 2 for delete records,
case 3 for update records and
case 4for Display records.
Step 8:Do Step 7 until will give choice > 4.
Step 9:Close Statement object and Connection object.
Step 10:Stop the Program.

SOURCE CODE:

```
// Use JDBC connectivity and create table, insert, update and delete data
import java.io.*;
import java.sql.*;
class jdbc
```

```
{
public static void main(String ar[])throws Exception
{
 Connection con;
 Statement st;
 ResultSet rs;
 BufferedReader br=new BufferedReader (new
InputStreamReader(System.in));
 int ch,rno,n;
 String na;
 Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
 con=DriverManager.getConnection("jdbc:odbc:stud");
 st=con.createStatement();
 do
 {
 System.out.println("DATABASE MANIPULATION USING JDBC");
 System.out.println("1.Insert\n2.Delete\n3.Update\n4.Display");
 System.out.println("Enter the choice");
 ch=Integer.parseInt(br.readLine());
 switch(ch)
 {
 case 1:
 System.out.println("Enter Id to Insert:");
 rno=Integer.parseInt(br.readLine());
 System.out.println("Enter name to Insert:");
 na=br.readLine();
 try
 {
 n=st.executeUpdate("insert into student_tab values("+rno+",'"+na+"')");
 System.out.println(n+" row Inserted!!");
 }
 catch(SQLException e) { }
 break;
 case 2:
```

```
System.out.println("Enter Id to Delete:");
rno=Integer.parseInt(br.readLine());
try
{
n=st.executeUpdate("delete * from student_tab where id="+rno);
System.out.println(n+" row Deleted!!");
}
catch(SQLException e){}
break;
case 3:
System.out.println("Enter Id to Edit:");
rno=Integer.parseInt(br.readLine());
System.out.println("Enter name to Edit:");
na=br.readLine();
try
{
n=st.executeUpdate("update student_tab set name='"+na+"' where
id="+rno);
System.out.println(n+" row Updated!!");
}
catch(SQLException e){}
break;
case 4:
try
{
rs=st.executeQuery("select * from student_tab");
System.out.println("ID\tNAME\n**********");
while(rs.next())
{
System.out.println(rs.getInt(1)+"\t"+rs.getString(2));
}
}
catch(SQLException e) { }
break;
```

```
 default:
 System.out.println("Invalid Choice");
 }
 }while(ch<=4);
st.close();
con.close();
}
}
```

OUTPUT:

```
DATABASE MANIPULATION USING JDBC
1..Insert
2.Delete
3.Update
4.Display
Enter Choice:
1
Enter Id to Insert:
111
Enter name to Insert:
haafi
1 row Inserted!!
DATABASE MANIPULATION USING JDBC
1.Insert
2.Delete
3.Update
4.Display
Enter Choice:4
ID NAME
**************
111 haafi
```

222 sita
DATABASE MANIPULATION USING JDBC
1.Insert
2.Delete
3.Update
4.Display
Enter Choice:2
Enter Id to Delete:
222
1 row Deleted!!
DATABASE MANIPULATION USING JDBC
1.Insert
2.Delete
3.Update
4.Display
Enter Choice:
4
ID NAME
****
111 haafi
DATABASE MANIPULATION USING JDBC
1.Insert
2.Delete
3.Update.

4.Display
Enter Choice:3
Enter Id to Edit:
111
Enter name to Edit:
haasika
1 row Updated!!
DATABASE MANIPULATION USING JDBC
1.Insert
2.Delete

3.Update
4.Display
Enter Choice:
4
ID NAME
****
111 haasika
DATABASE MANIPULATION USING JDBC
1.Insert
2.Delete
3.Update
4.Display
Enter Choice: 5
Invalid Choice.

EX.NO:2

JDBC Program Using Prepared Statement

Aim: write a java program using JDBC to perform insert and Select the Records.

Algorithm:

Step1:Start the program.
Step 2:Include packages java.io and java.sql.
Step 3:Define class with name “jdbc” and define the main function.
Step 4:Declare objects for Connection, Prepared Statement, ResultSet and alsodeclare the object for BufferedReader class.
Step 5:Register the JdbcOdbcDriver and make a connection using getConnection() by giving Data Source Name “ORCL”.
Step 6: Insert the Record.
Step 7: Select the Records from Table.
Step 8: Display Selected Records on Console.
Step 9: Close the Prepared Statement Object and Connection Object.
Step 10: Stop the Program.

Source Code:

```
import java.sql.*;
public class SelectPrepared {
  public static void main(String[] args) {
      Connection con = null;
      PreparedStatement ps = null;
      ResultSet rs = null;
      try {
         Class.forName("oracle.jdbc.driver.OracleDriver");
         con = DriverManager.getConnection(
```

```
        "jdbc:oracle:thin:@localhost:1521:xe", "system", "manager");
        ps = con.prepareStatement("insert into employee values(?,?,?)");
        // set values to query parameters
        ps.setInt(1, 10);
        ps.setString(2, "Alice");
        ps.setString(3, "Hyd");
        int result = ps.executeUpdate();
        if (result != 0) {
           System.out.println("Record is successfully inserted");
        } else {
           System.out.println("Inserting record is failed");
        }
        String qry = "select * from employee where ename = ?";
        ps = con.prepareStatement(qry);
        // set value to query parameter
        ps.setString(1, "Alice");
        rs = ps.executeQuery();
        System.out.println(“ Selected Records:”);
        System.out.println(“   eid    “ +   “ename “   +  “Designation”);
        System.out.println(“----------------------------------------------“);
        while (rs.next()) {
           System.out.println(rs.getInt(1) + "\t" + rs.getString(2) + "\t"
        + rs.getString(3));
        }
rs.close();
        ps.close();
        con.close();
      } catch (Exception e) {
        e.printStackTrace();
      }}}
```

Output:

Record is successfully inserted.
Selected Records:

Eid	ename	Designation
7	Vasavi	Assoc.Prof
8	Suresh	Assoc.Prof
9	Sukumar	Asst.Prof

EX.NO:3

Servlet Programming

3.1)Aim:-Write a Servlet Program to Display the Request Details.

Source Code:

Program1: Define the web.xml file.

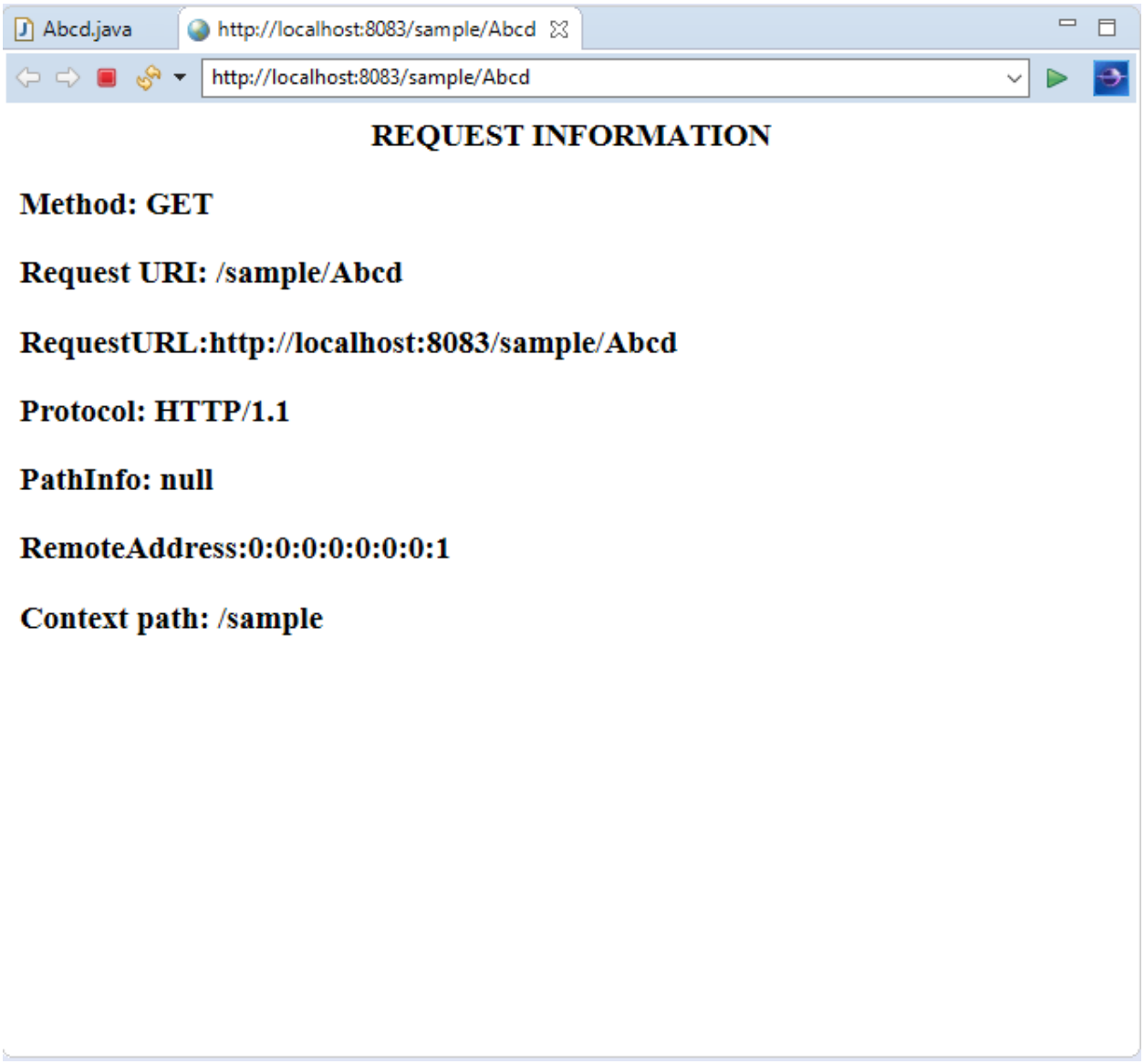
```
<web-app>
<servlet>
<servlet-name> RequestInfoSrv</servlet-name>
<servlet-class> RequestInfoSrv</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name> RequestInfoSrv</servlet-name>
<url-pattern>/request</url-pattern >
</servlet-mapping >
</web-app>
```

Program 2: RequestInfoSrv.java

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class RequestInfoSrv extends HttpServlet {
   protected void doGet(HttpServletRequest req, HttpServletResponse resp)
   throws  ServletException, IOException {
      resp.setContentType("text/html");
      PrintWriter out = resp.getWriter();
      out.println("<h3>REQUEST INFORMATION</h3>");
      out.println("<h3>Method: " + req.getMethod() + "</h3>");
      out.println("<h3>Request URI: " + req.getRequestURI() + "</h3>");
      out.println("<h3>RequestURL:" + req.getRequestURL() + "</h3>");
```

```
            out.println("<h3>Protocol: " + req.getProtocol() + "</h3>");
            out.println("<h3>PathInfo: " + req.getPathInfo() + "</h3>");
            out.println("<h3>RemoteAddress:"+req.getRemoteAddr()+ </h3>");
            out.println("<h3>Context path: " + req.getContextPath() + "</h3>");
            //close stream out.close();
        }//doGet
}//class
```

OUTPUT:



3.2)Aim:-Write a Servlet Program to handle the User Form.

Source Code:

Program1: web.xml

```
 <web-app>
 <servlet>
 <servlet-name>RequestParametersSrv</servlet-name>
 <servlet-class>RequestParametersSrv</servlet-class>
 </servlet>
 <servlet-mapping>
 <servlet-name>RequestParametersSrv</servlet-name>
 <url-pattern>/RequestParametersSrv</url-pattern >
 </servlet-mapping >
</web-app>
```

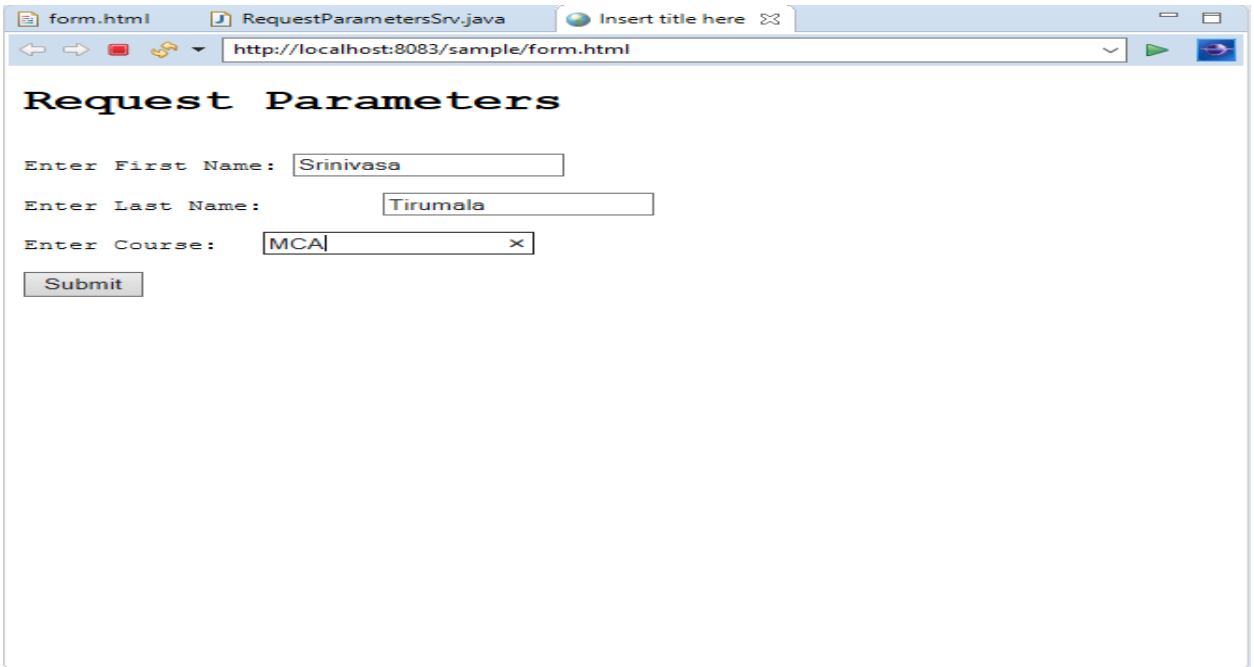
Program 2: RequestParams.html

```
 <!DOCTYPE html>
 <html>
 <head>
 <meta charset="ISO-8859-1">
 <title>RequestParams</title>
 </head>
 <body>
        <form action="RequestParametersSrv" method="get">
        <pre>
        <h1>Request Parameters</h1>
        Enter First Name: <input type="text" name="fname"><br>
        Enter Last Name:   <input type="text" name="lname"><br>
        Enter Course:       <input type="text" name="cname"><br>
        <input type="submit" value="Submit">
        </pre>
        </form>
 </body>
</html>
```

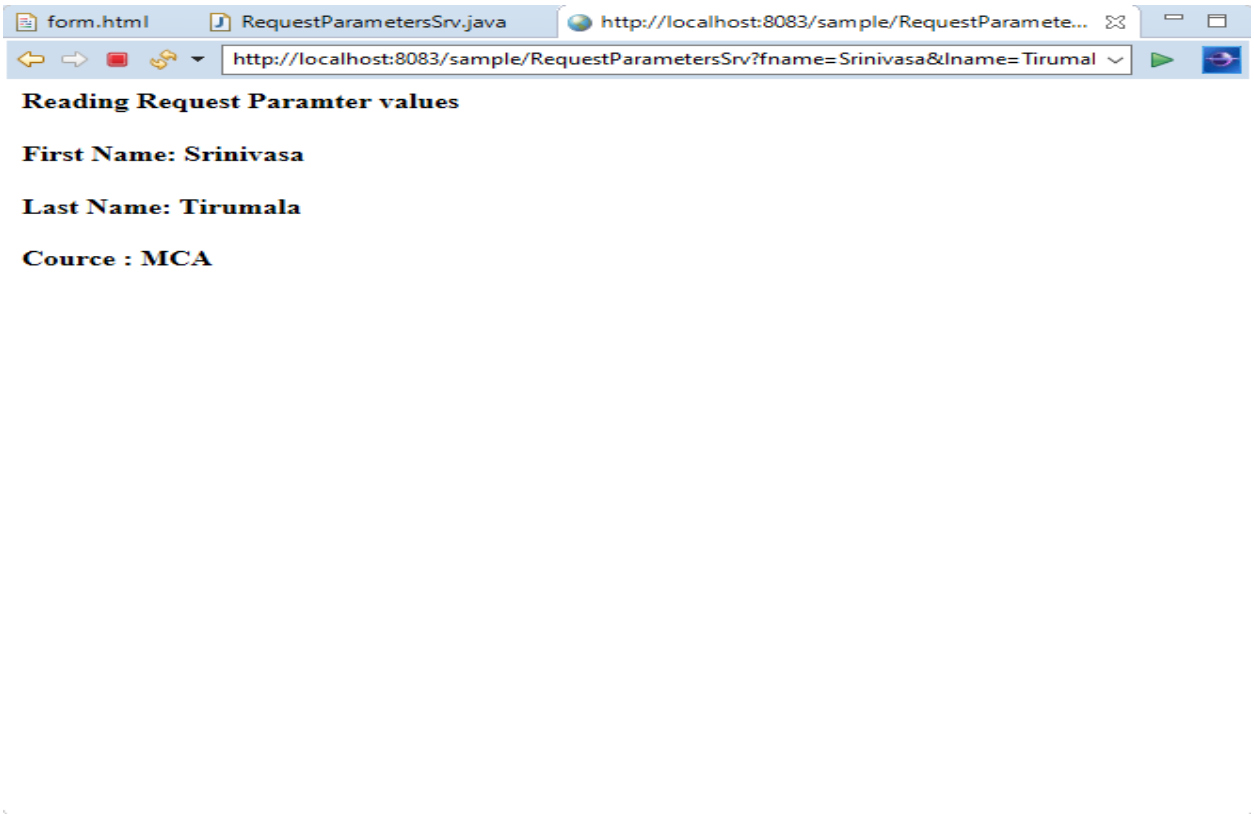
Program 3:RequestParametersSrv.java

```
import java.io.*;

import javax.servlet.*;

import javax.servlet.http.*;
public class RequestParametersSrv extends HttpServlet {
protected void doGet(HttpServletRequest req, HttpServletResponse resp)
throws ServletException, IOException {
        resp.setContentType("text/html");
        PrintWriter out = resp.getWriter();
        // reading request parameters
        String firstName = req.getParameter("fname");
        String lastName = req.getParameter("lname");
        String cource = req.getParameter("cname");
        out.println("<h4>Reading Request Paramter values</h4>");
out.println("<h4>First Name: " + firstName + "</h4>");
out.println("<h4>Last Name: " + lastName + "</h4>");
out.println("<h4>Cource : " + cource + "</h4>"); //close stream
        out.close();
    }//doGet
}//class
```

Output:



After pressing the Submit button in Form, we see the below output.



3.3)Aim:-Write a Servlet Program to create Cookie and Display Cookie.

Source Code:

Program1: web.xml

```
<web-app>
<servlet>
<servlet-name>SetCookiesSrv</servlet-name>
<servlet-class>SetCookiesSrv</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>SetCookiesSrv</servlet-name>
<url-pattern>/cookie</url-pattern >
</servlet-mapping >
```

```
 <servlet>
 <servlet-name>ShowCookiesSrv</servlet-name>
 <servlet-class>ShowCookiesSrv</servlet-class>
 </servlet>
 <servlet-mapping>
 <servlet-name>ShowCookiesSrv</servlet-name>
 <url-pattern>/showcookie</url-pattern >
 </servlet-mapping >
</web-app>
```

Program2: setCookieSrv.java

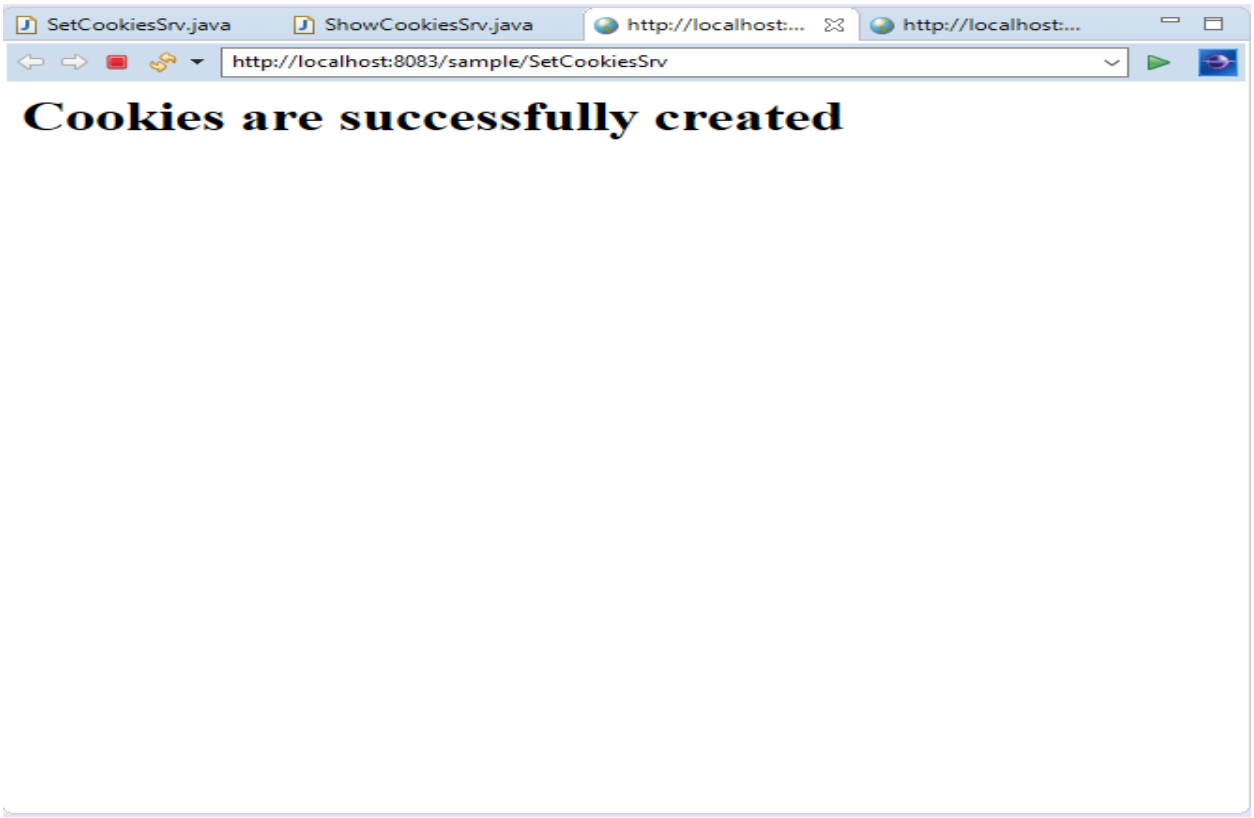
```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class SetCookiesSrv extends HttpServlet {
   protected void doGet(HttpServletRequest req, HttpServletResponse resp)
throws ServletException, IOException {
      resp.setContentType("text/html");
      PrintWriter out = resp.getWriter();
      // create in memory cookies
      Cookie ck1 = new Cookie("TS", "Hyderabad"); Cookie ck2 = new
Cookie("KA", "Banglore"); // add cookies
      resp.addCookie(ck1);
      resp.addCookie(ck2);
      // create persistent cookies
      Cookie ck3 = new Cookie("India", "Delhi");
      Cookie ck4 = new Cookie("USA", "Washington");
      ck3.setMaxAge(1800);
      ck4.setMaxAge(1800);
      // add cookies
```

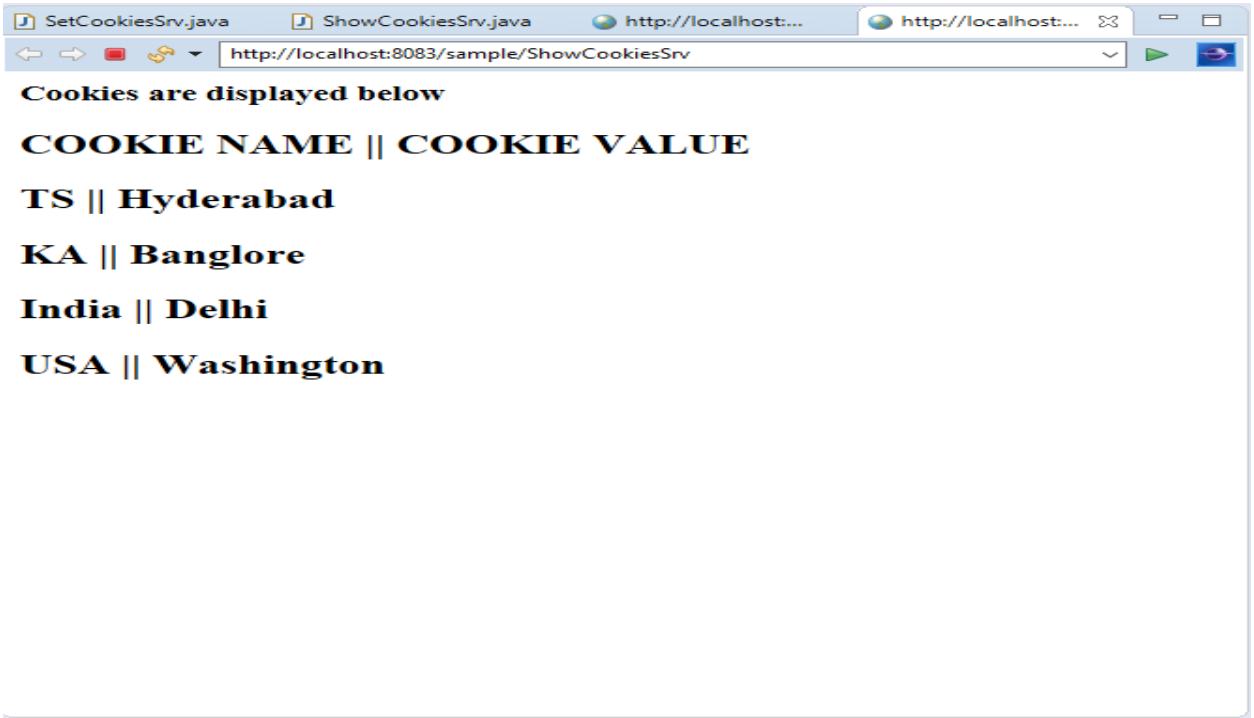
```
            resp.addCookie(ck3);
            resp.addCookie(ck4);
            // generate Response
            out.println("<h1>Cookies are successfully created</h1>");
            //close the stream out.close();
        }
}
```

Program 3:showCookiesSrv.java

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class ShowCookiesSrv extends HttpServlet {
   protected void doGet(HttpServletRequest req, HttpServletResponse resp)
throws ServletException, IOException {
        resp.setContentType("text/html");
        PrintWriter out = resp.getWriter();
        out.println("<h3>Cookies are displayed below</h3>");
        //read and display cookies
        Cookie ck[] = req.getCookies();
        //gives all cookies along with the request
        out.println("<h2> COOKIE NAME || COOKIE VALUE </h2>");
        if (ck != null) {
            for (Cookie cck : ck) {
                out.println("<h2>" + cck.getName() + " || " + cck.getValue()+
"</h2>");
            }
        }
    }//doGet
}//class
```

Output:


SetCookiesSrv.java
ShowCookiesSrv.java
http://localhost:...
http://localhost:...
http://localhost:8083/sample/SetCookiesSrv
Cookies are successfully created


SetCookiesSrv.java
ShowCookiesSrv.java
http://localhost:...
http://localhost:...
http://localhost:8083/sample/ShowCookiesSrv
Cookies are displayed below
COOKIE NAME || COOKIE VALUE
TS || Hyderabad
KA || Banglore
India || Delhi
USA || Washington

3.4)Aim:-Write a Servlet Program to do Session Tracking.

Source Code:

Program1:web.xml

```
<web-app>
    <servlet>
        <servlet-name> FirstSrv </servlet-name>
        <servlet-class>FirstSrv</servlet-
    class></servlet>
    <servlet-mapping>
        <servlet-name> FirstSrv </servlet-name>
        <url-pattern>/ FirstSrv </url-pattern>
    </servlet-mapping>
    <servlet>
        <servlet-name> SecondSrv </servlet-name>
        <servlet-class>SecondSrv</servlet-
    class></servlet>
    <servlet-mapping>
        <servlet-name> SecondSrv </servlet-name>
        <url-pattern>/ SecondSrv </url-pattern>
    </servlet-mapping>
</web-app>
```

Program2: Session.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>SessionTraking</title>
</head>
```

```
<body>
	<form action="FirstSrv" method="get">
	<h1 style="text-align: center;">HttpSession Tracking</h1><h1>IT Filling
	Registration</h1><pre>
	Enter PName:<input type="text" name="pname">
	Enter FName:<input type="text" name="fname">
	<input type="submit" value="Continue">
	</pre>
	</form>
</body>
</html>
```

Program3:firstSrv.java

```
	import java.io.*;
	import java.util.*;
	import javax.servlet.*;
	import javax.servlet.http.*;
	public class FirstSrv extends HttpServlet {
	   protected void doGet(HttpServletRequest req, HttpServletResponse resp)
	throws ServletException, IOException {
		resp.setContentType("text/html");
		PrintWriter out = resp.getWriter();
		// read form1 data
		String pname = req.getParameter("pname");
		String fname = req.getParameter("fname");
		//create Session for browser window
		HttpSession ses = req.getSession(true);
		//store form1/req1 data in Session Attribute
		ses.setAttribute("pname", pname);
		ses.setAttribute("fname", fname);
		//generate dynamic form2 data
		out.println("<form action='SecondSrv' method='get'>");
	out.println("<pre><h1>SESSION TRACKING</h1>");
		out.println("<h1>IT FILING REGISTRATION</h1>");
		out.println("Income for this year <input type ='text' name='income'>");
		out.println("Tax <input type ='text' name='tax'>");
		out.println("<input type ='submit' value='Register'><pre>");
		out.println("</form>");
```

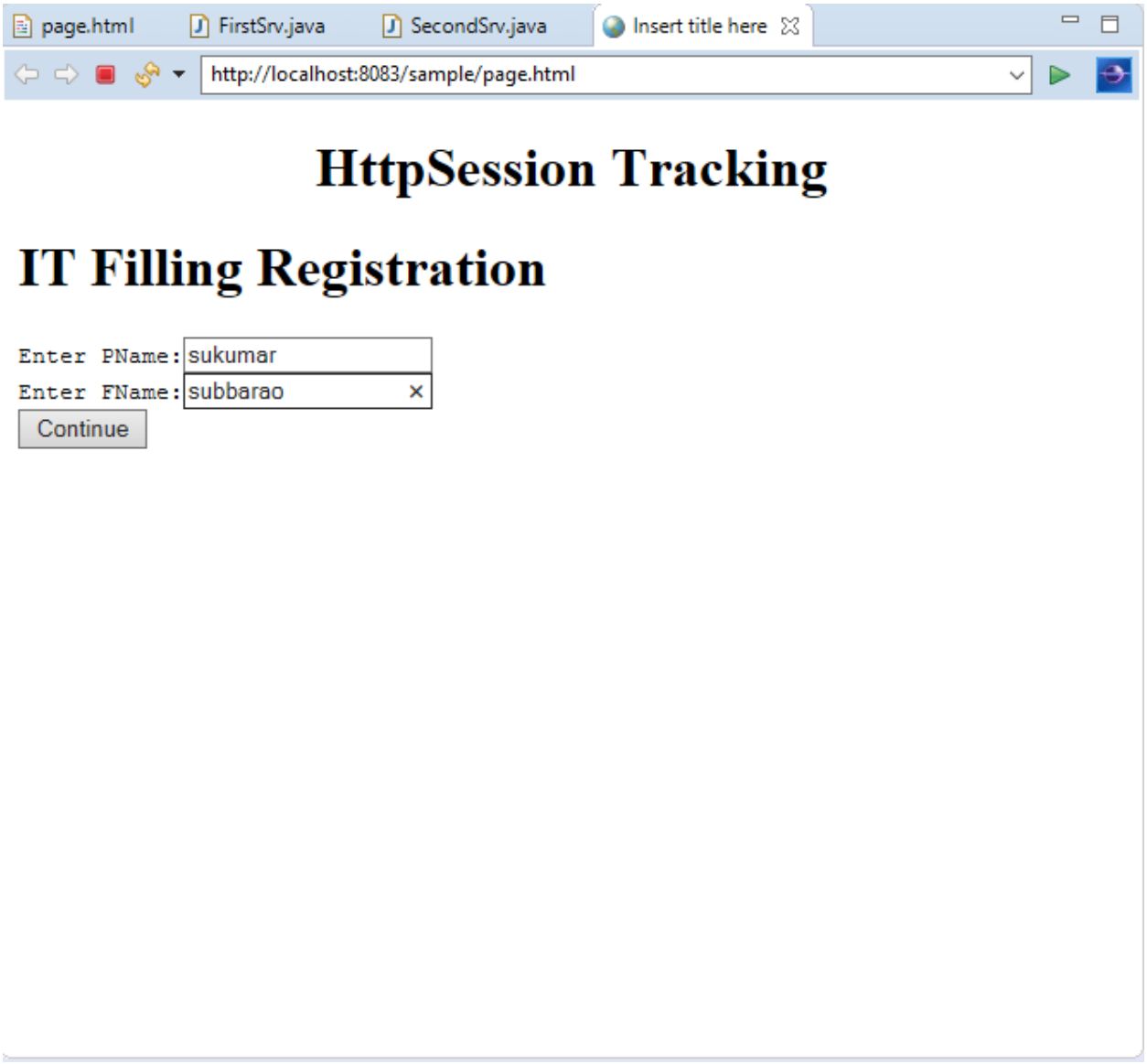
```
            out.println("<br>Session ID: " + ses.getId());
            Date created = new Date(ses.getCreationTime());
            Date accessed = new Date(ses.getLastAccessedTime());
            out.println("Session Created: " + created);
            out.println("Last Accessed: " + accessed);
        }
}
```

Program4:SecondSrv.java

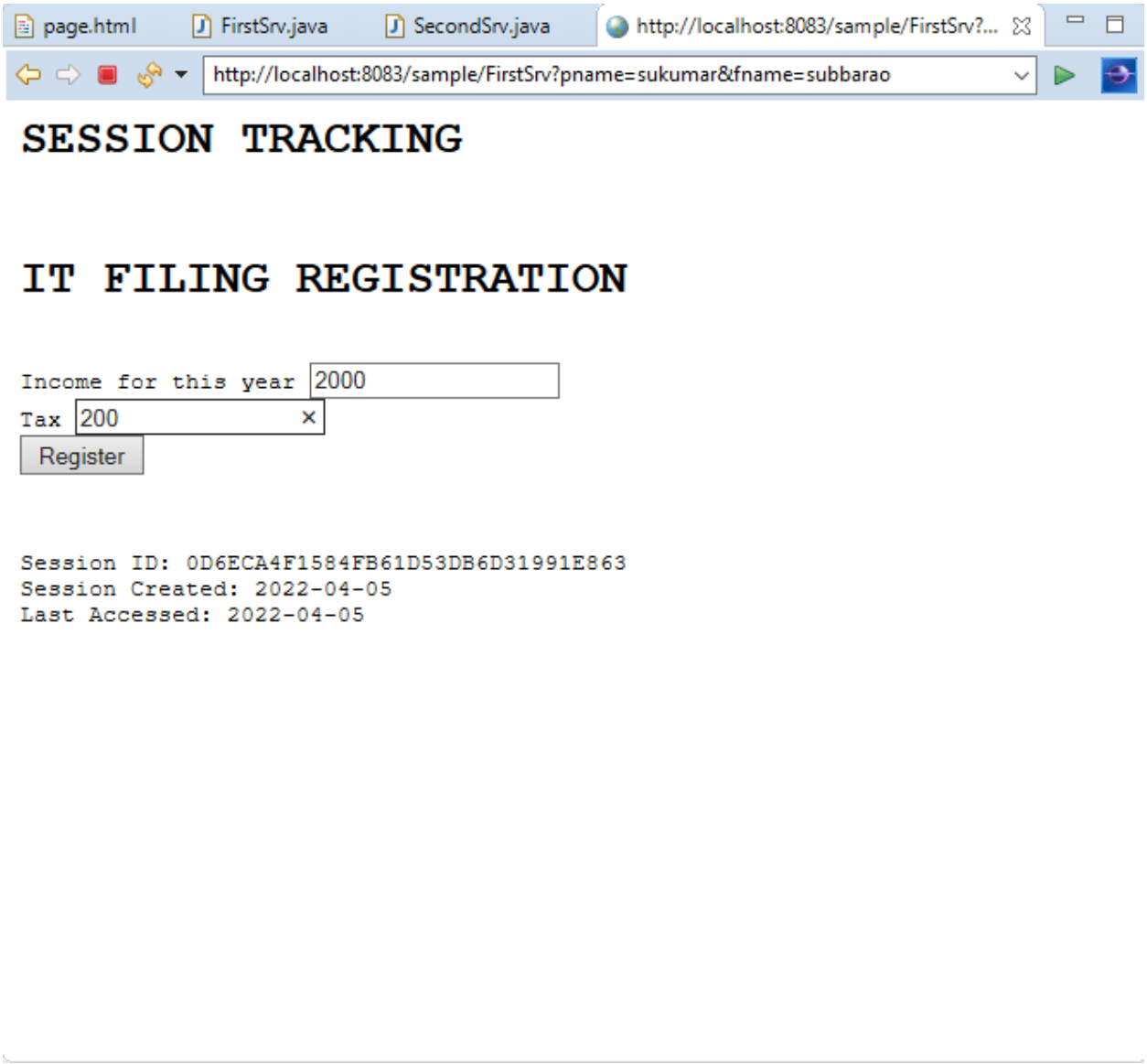
```
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class SecondSrv extends HttpServlet {
   protected void doGet(HttpServletRequest req, HttpServletResponse resp)
throws ServletException, IOException {
      resp.setContentType("text/html");
      PrintWriter out = resp.getWriter();
      // read form2/req2 data
      int income = Integer.parseInt(req.getParameter("income"));
      int tax = Integer.parseInt(req.getParameter("tax"));
      //Get access to Session object
      HttpSession ses = req.getSession(false);
      //read form2/req2 data from session attributes
      String pname = (String) ses.getAttribute("pname");
      String fname = (String) ses.getAttribute("fname");
      //display form1, form2 data
      out.println("<pre><h1>SESSION TRACKING</h1>");
      out.println("<h1>FORM1 DATA: Name: " + pname + " & " +
"F_Name: "+fname + "<h1>");
      out.println("<h1>FORM2 DATA: Income: " + income + " & " + "Tax:
" + tax+"</h1><pre>");
      out.println("<br>Session ID: " + ses.getId()); Date created = new
Date(ses.getCreationTime()); Date accessed = new
Date(ses.getLastAccessedTime());
      out.println("Session Created: " + created);
      out.println("Last Accessed: " + accessed);
      //invalidate the session ses.invalidate();
```

```
            out.println("<a href='SessionTracking.html'><h4>Home</h4></a>");
        }
    }
```

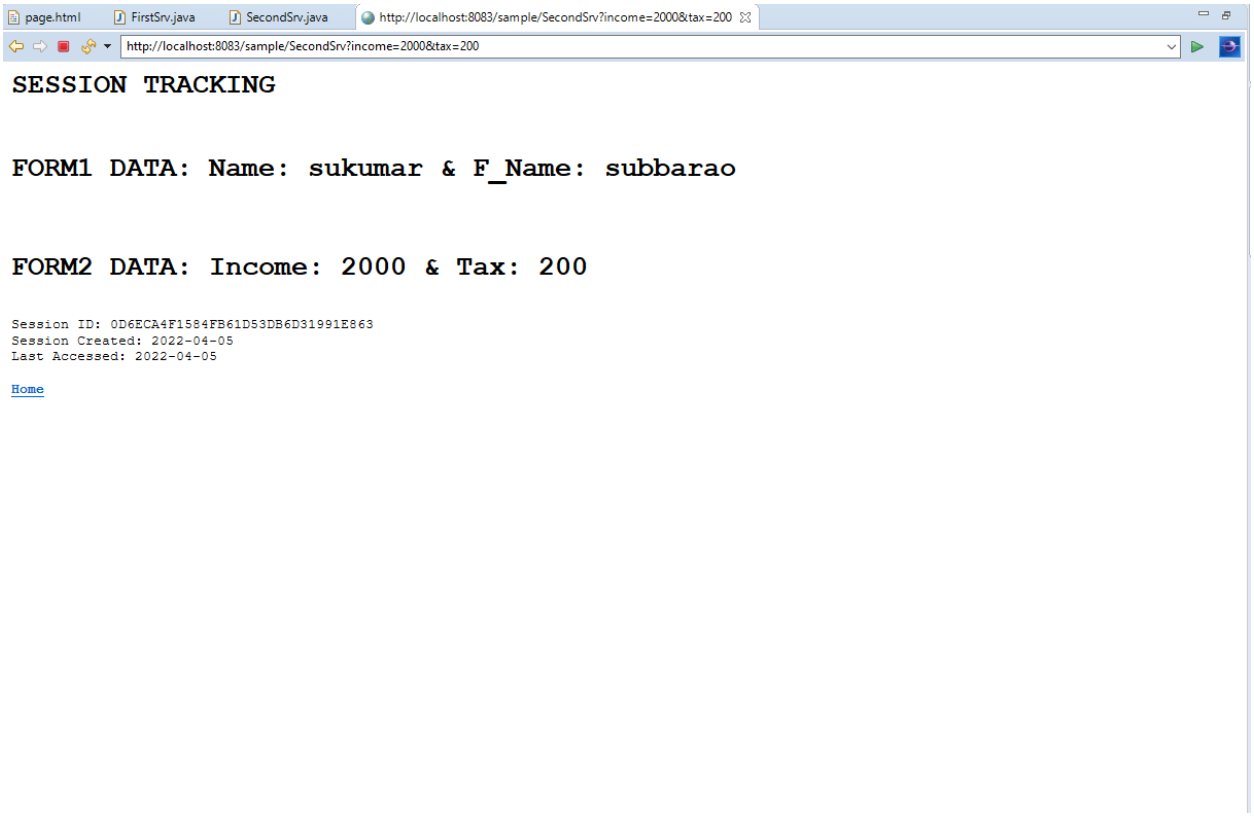
Output:



After pressing the continue button , we see below output.



After pressing the Register button, we see the below output.


page.html
FirstSrv.java
SecondSrv.java
http://localhost:8083/sample/SecondSrv?income=2000&tax=200
http://localhost:8083/sample/SecondSrv?income=2000&tax=200
SESSION TRACKING
FORM1 DATA: Name: sukumar & F_Name: subbarao
FORM2 DATA: Income: 2000 & Tax: 200
Session ID: 0D6ECA4F1584FB61D53DB6D31991E863
Session Created: 2022-04-05
Last Accessed: 2022-04-05
Home

EX.NO:4

JSP Programming

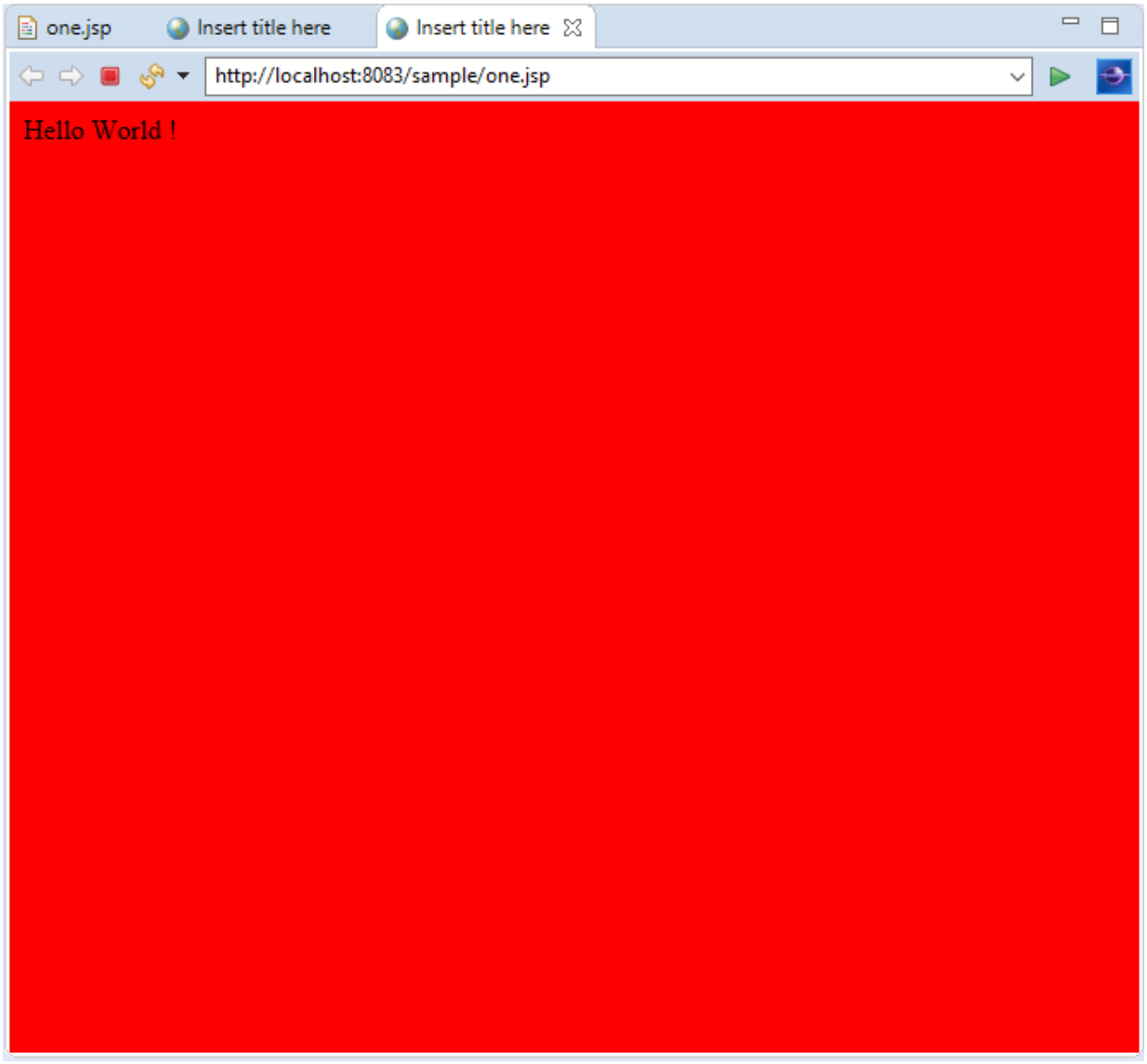
4.1)Aim:-Write a JSP Program to display the “Hello World”.

Source Code:

Program:

```
<html>
<body>
<! -- This is the JSP file-->
<%
out.println ("Hello World !");
%>
</body>
</html>
```

Output:


one.jsp
Insert title here
Insert title here
http://localhost:8083/sample/one.jsp
Hello World !

4.2)Aim:-Write a JSP Program to demonstrate the Cookie.

Source Code:

Program1: Cookieform.jsp

```
<%@ page language="java" %>
<html>
<head>
<title>Cookie Input Form</title>
</head>
<body>
<form method="post" action="setcookie.jsp">
<p><b>Enter Your Name: </b><input type="text" name="username"><br>
<input type="submit" value="Submit">
</form>
</body>
```

Program2 :setCookie.jsp

```
<%@ page language="java" import="java.util.*"%>
<%
String username=request.getParameter("username");
if(username==null) username="";
Date now = new Date();
String timestamp = now.toString();
Cookie cookie = new Cookie ("username",username);
cookie.setMaxAge(365 * 24 * 60 * 60);
response.addCookie(cookie);
%>
<html>
<head>
<title>Cookie Saved</title>
</head>
<body>
```

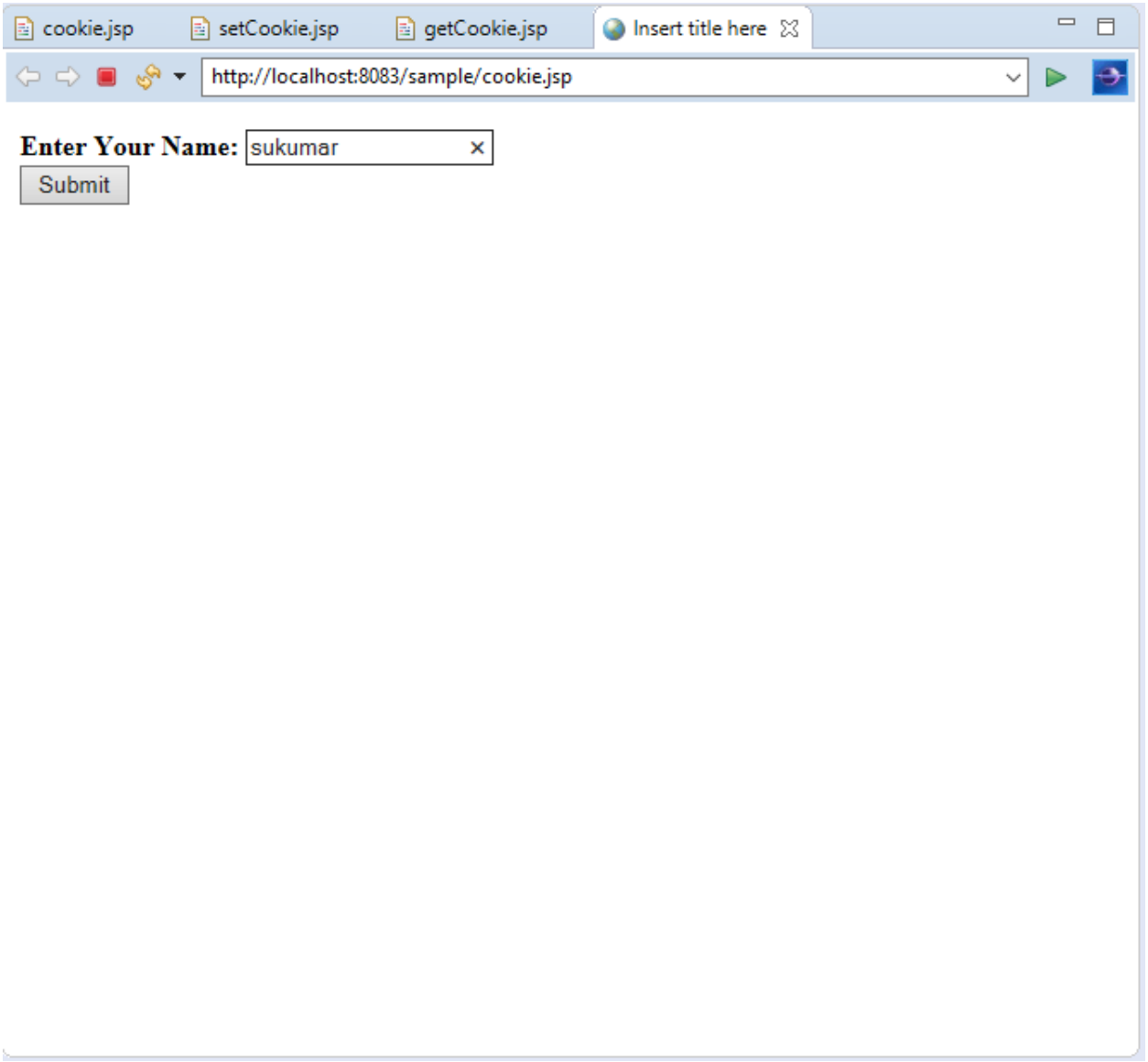
```
<p><a href="showcookievalue.jsp">Next Page to view the cookie value</a><p>
</body>
```

Program3:ShowCookie.jsp

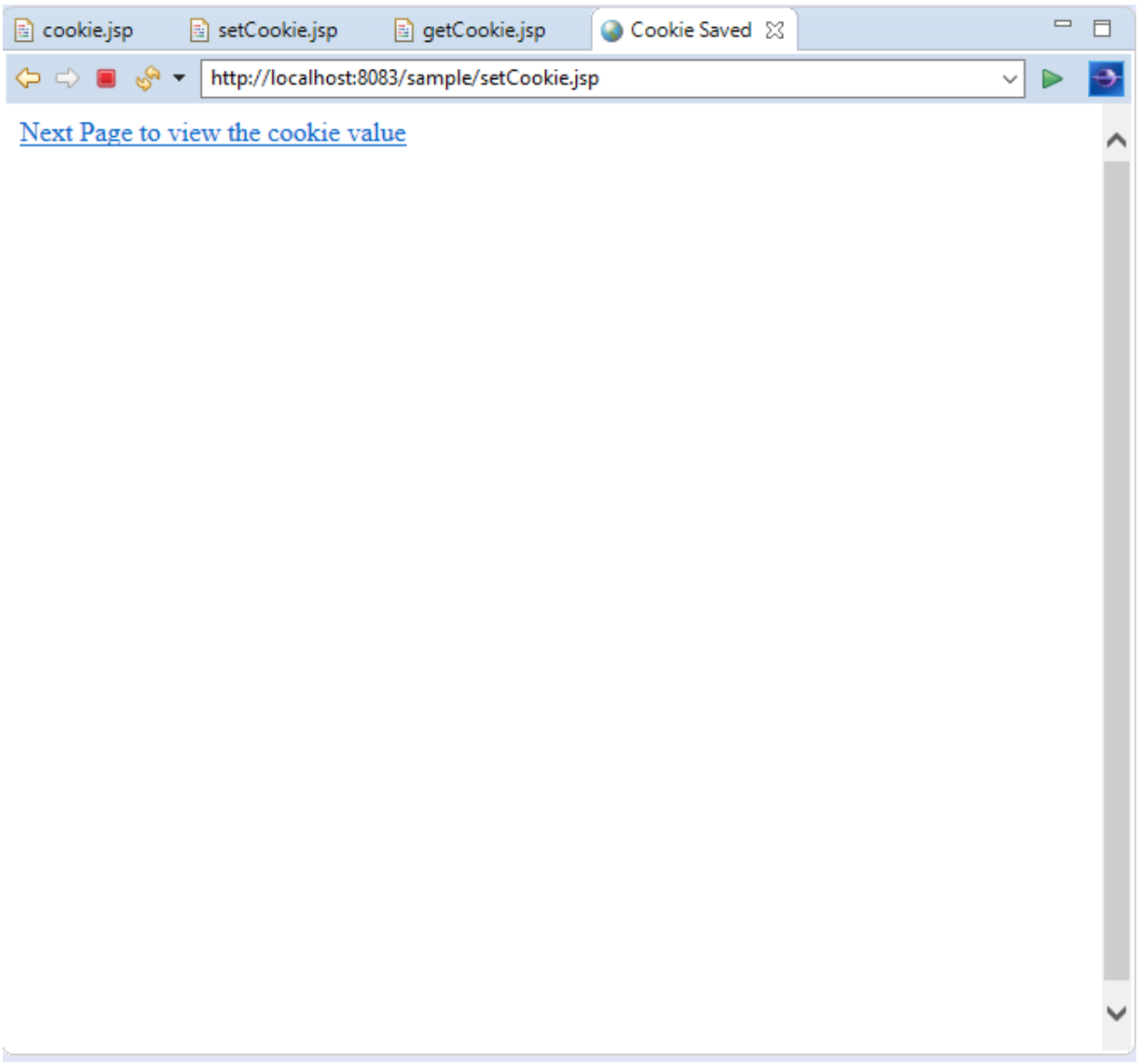
```
<%@ page language="java" %>
<%
String cookieName = "username";
Cookie cookies [] = request.getCookies ();
Cookie myCookie = null;
if (cookies != null)
{
for (int i = 0; i < cookies.length; i++)
{
if (cookies [i].getName().equals (cookieName))
{
myCookie = cookies[i];
break;
}
}
}
%>
<html>
<head>
<title>Show Saved Cookie</title>
</head>
<body>
<%
if (myCookie == null) {
%>
No Cookie found with the name <%=cookieName%>
<%
} else {
```

```
%>
<p>Welcome: <%=myCookie.getValue()%>.
<%
}
%>
</body>
```

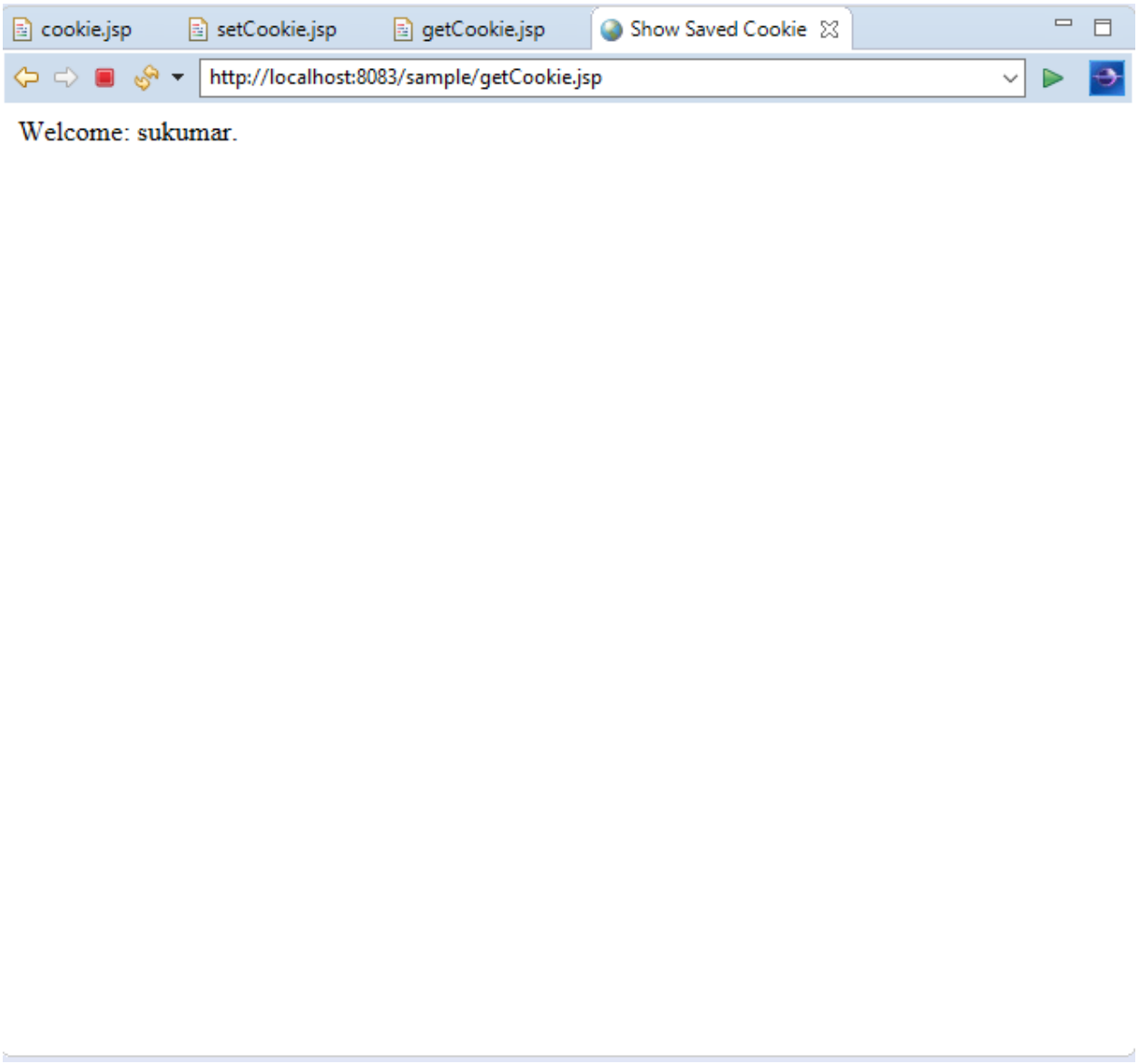
Output:



After pressing the submit button, we see the below output.



Click the Link "Next page to view the cookie value" to see the cookie.


cookie.jsp
setCookie.jsp
getCookie.jsp
Show Saved Cookie
http://localhost:8083/sample/getCookie.jsp
Welcome: sukumar.

EX.NO:5

Aim: write a java program to implement a Client/Server application using RMI.

Algorithm:

Program 1: Define the Remote Interface

Step 1:Start the program.
Step 2:Import the package java.rmi.*.
Step 3:Define interface "AddServerIntf" by extends from Remote.
Step 4:Declare the methods to perform arithmetic operation add, sub, mul, div, modulo and throwsRemoteException.
Step 5:Stop the program.

Program 2: Implement Remote Interface

Step 1:Start the program.
Step 2:Import the packages java.rmi.* and java.rmi.server.*.
Step 3:Define class AddServerImpl by extends from "UnicastRemoteObject" and implements"AddServerIntf".
Step 4:Define the procedure for interface methods by throwing RemoteException.
Step 5:Stop the program.

Program 3: Implementation of Server Machine
Step 1:Start the program.
Step 2:Import the packages java.rmi.* and java.net.*
Step 3:Define class server with main function.
Step 4:Create object for the class AddServerImpl.

Step 5:Using naming.rebind() method add the interface to the server .
Step 6:Stop the program.

Program 4: Implementation of Client Machine

Step 1:Start the program.
Step 2:Define class “client” with main() function
Step 3:Create object for server interface with proper URL definition using naming.lookup();
Step 4:Using this object call required methods and handling exception.
Step 5:Stop the program.

SOURCE CODE:

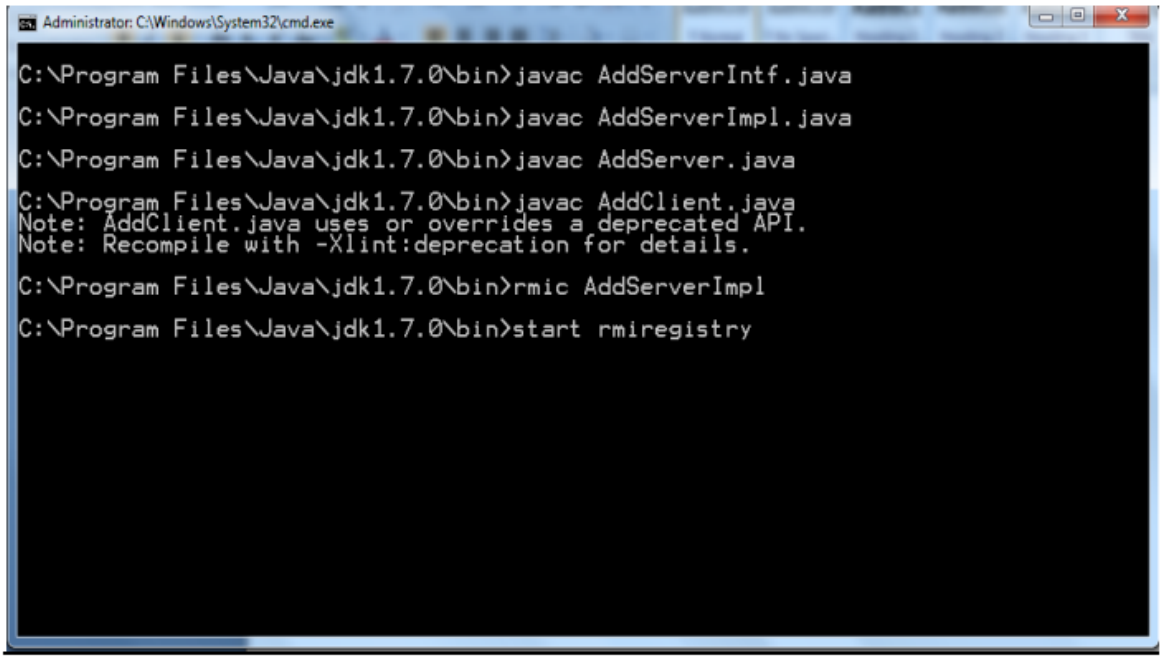
```
// 1. Define the Remote Interface

import java.rmi.*;
public interface AddServerIntf extends Remote
{
int add(int a,int b) throws RemoteException;
int sub(int a,int b) throws RemoteException;
int mul(int a,int b) throws RemoteException;
int div(int a,int b) throws RemoteException;
int mod(int a,int b) throws RemoteException;
}

// 2. Implement Remote Interface
import java.rmi.*;
import java.rmi.server.*;
public class AddServerImpl extends UnicastRemoteObject implements
AddServerIntf
{
public AddServerImpl() throws RemoteException
{}
public int add(int a,int b)throws RemoteException
```

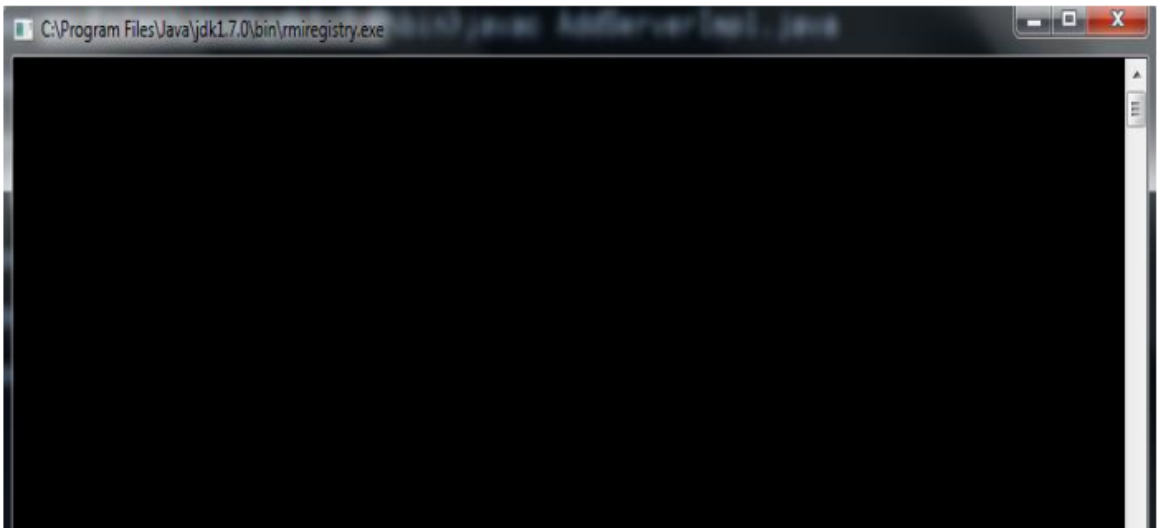
```
{
return (a+b);
}
public int sub(int a,int b)throws RemoteException
{
return (a-b);
}
public int mul(int a,int b)throws RemoteException
{
return (a*b);
}
public int div(int a,int b)throws RemoteException
{
return (a/b);
}
public int mod(int a,int b)throws RemoteException
{
return (a%b);
}
}

// 3. Implementation of Server Machine
import java.rmi.*;
import java.net.*;
public class AddServer
{
public static void main(String args[])
{
try
{
AddServerImpl obj = new AddServerImpl();
Naming.rebind("addserver", obj);
System.out.println("server started");
}
```

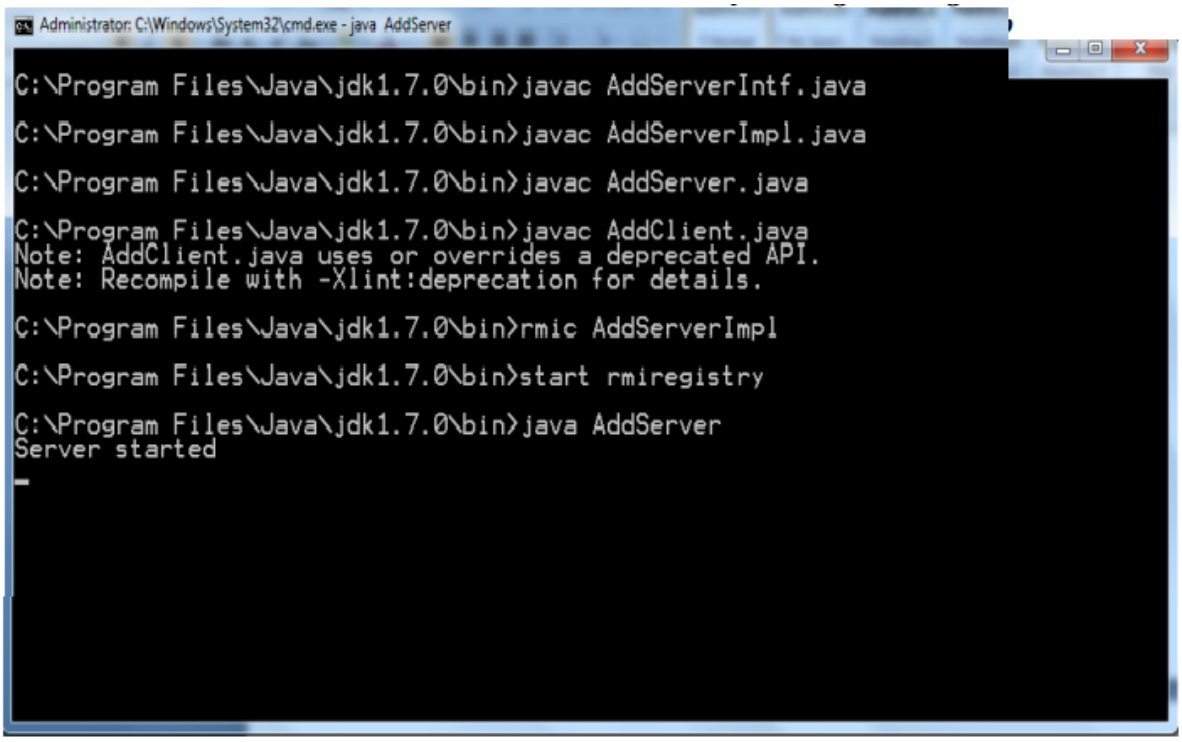
```
catch (Exception e)
{
System.out.println("Exception: " + e);
}
}
}

// 4. Implementation of Client Machine
import java.rmi.*;
import java.io.*;
public class AddClient
{
public static void main(String args[])
{
try
{
DataInputStream ds=new DataInputStream(System.in);
String s="rmi://MY-PC/addserver";
AddServerIntf obj = (AddServerIntf)Naming.lookup(s);
System.out.println("ENTER THE VALUES FOR a & b:");
int a=Integer.parseInt(ds.readLine());
int b=Integer.parseInt(ds.readLine());
System.out.println("ADDITION="+obj.add(a,b));
System.out.println("SUBTRACTION="+obj.sub(a,b));
System.out.println("MULTIPLICATION="+obj.mul(a,b));
System.out.println("DIVISION="+obj.div(a,b));
System.out.println("MODULODIVISION="+obj.mod(a,b));
}
catch (Exception e)
{
System.out.println("Exception: " + e);
 }
}
}
```

OUTPUT:

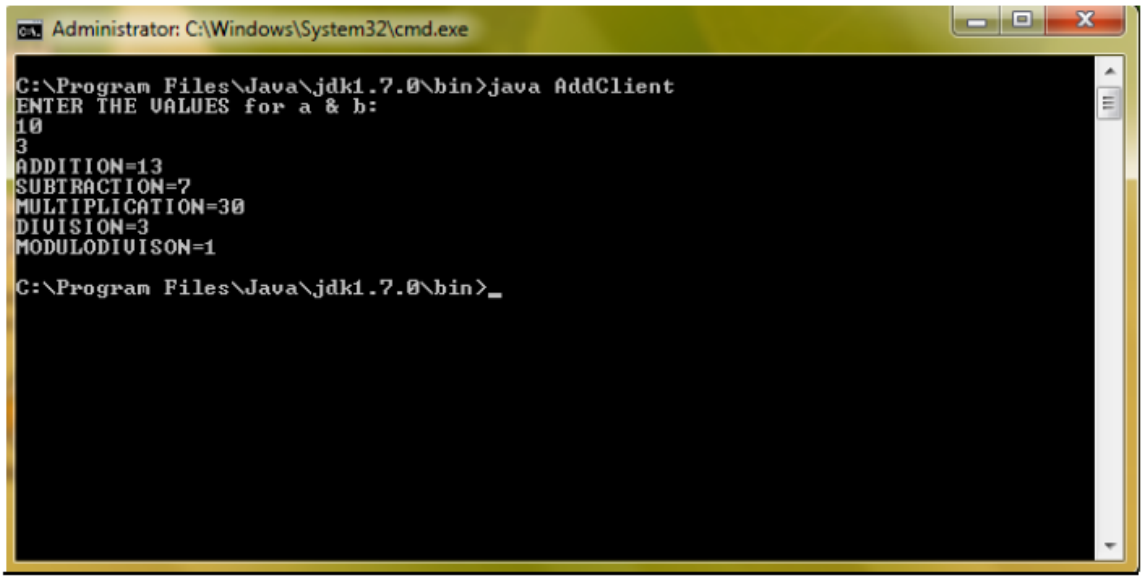
LOCAL HOST SERVER SIDE COMMAND WINDOW:







LOCAL HOST CLIENT SIDE COMMAND WINDOW:



RESULT:

The above program has been executed successfully and the output was verified.