**Steps to create servlet programs in eclipse**

1. Open eclipse IDE
2. Create dynamic web project

File🡪 new🡪 Dynamic web Project🡪give some project Name in textbox🡪 Finish

Project Name is seen project explorer

1. Create HTML File

Click project name on leftside🡪src🡪main🡪webapp🡪right click webapp🡪new🡪 html file

1. Create servlet

Click Java resources on leftside(project explorer)🡪src/main/java rightclick🡪New🡪servlet🡪 give servlet name as action name in the html form🡪finish

1. Add External jar file

Src/main/java🡪buildpath🡪configure build path🡪select libraries🡪select modulepath🡪 rightside click Add External Jar’s🡪select

C:\Program Files\Apache Software Foundation\Tomcat 10.1\lib\Servlet.api🡪apply🡪apply and close

1. Now press Run button in the HTML file

Start server by selecting tomcat 10.1 as localhost🡪 finish

Week 6: Design a controller with servlet that provides the interaction with web application

Sol:

Input.html

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

</head>

<body>

<form method=*"get"* action=*"HelloServlet"*>

<pre>

Enter your name: <input type=*"text"* name=*"t1"*>

<input type=*"submit"*> <input type=*"reset"*>

</pre>

</form>

</body>

</html>

HelloServlet.java

import jakarta.servlet.ServletException;

import jakarta.servlet.http.HttpServlet;

import jakarta.servlet.http.HttpServletRequest;

import jakarta.servlet.http.HttpServletResponse;

import java.io.IOException;

import java.io.PrintWriter;

public class HelloServlet extends HttpServlet {

private static final long *serialVersionUID* = 1L;

/\*\*

Default constructor.

\*/

public HelloServlet() {

}

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

PrintWriter pw=response.getWriter();

String s=request.getParameter("t1");

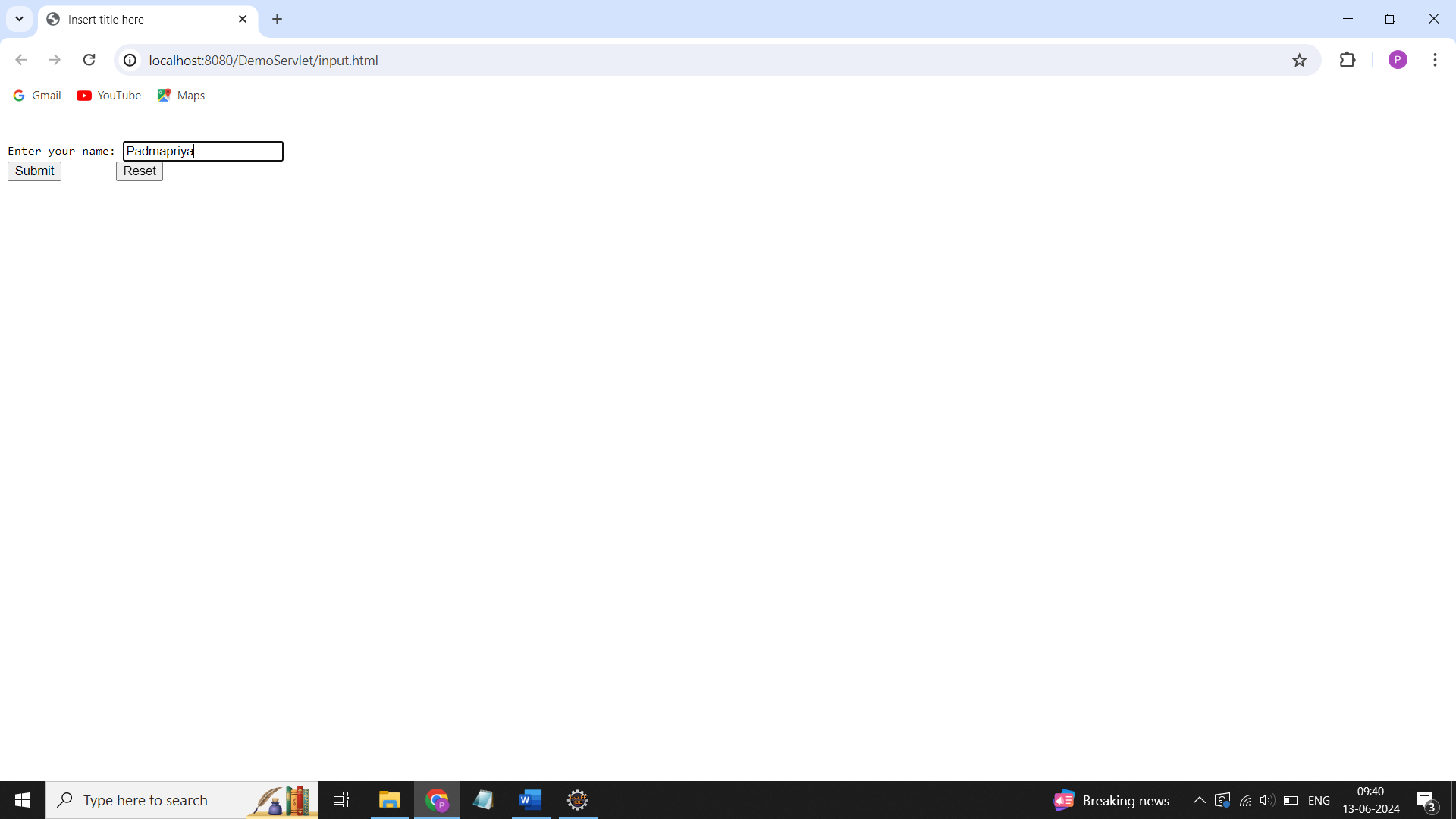
pw.println("<h1>Hello,"+s);

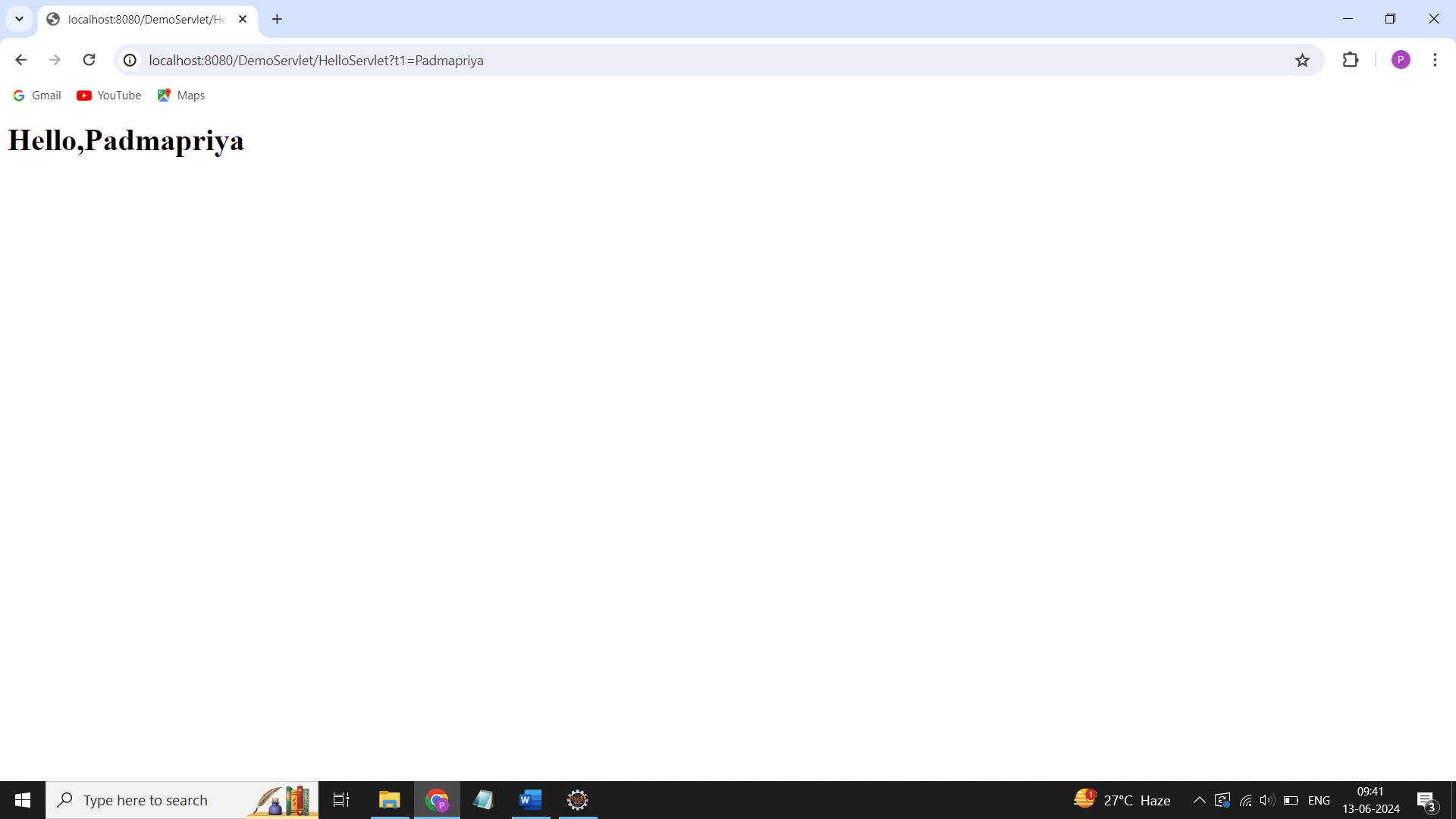
pw.close();

}

}

Output:





Week 7: Maintaining the transactional history of any user is very important. Explore the various session tracking mechanism using Cookies

Sol:

CookieDemo.html

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

</head>

<body>

<form action = *"CookiesDemo"* method = *"GET"*>

First Name: <input type = *"text"* name = *"first\_name"*>

<br><br>

Last Name: <input type = *"text"* name = *"last\_name"* />

<br><br>

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

</form>

</body>

</html>

CookieDemo,java

import jakarta.servlet.ServletException;

import jakarta.servlet.annotation.WebServlet;

import jakarta.servlet.http.Cookie;

import jakarta.servlet.http.HttpServlet;

import jakarta.servlet.http.HttpServletRequest;

import jakarta.servlet.http.HttpServletResponse;

import java.io.IOException;

import java.io.PrintWriter;

public class CookiesDemo extends HttpServlet {

private static final long *serialVersionUID* = 1L;

public CookiesDemo() {

super();

// TODO Auto-generated constructor stub

}

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

// Create cookies for first and last names.

Cookie firstName = new Cookie("first\_name",

request.getParameter("first\_name"));

Cookie lastName = new Cookie("last\_name",

request.getParameter("last\_name"));

// Set expiry date for both the cookies.

firstName.setMaxAge(60); // 1min-60secs

lastName.setMaxAge(60\*60\*2); // 2hrs

// Add both the cookies in the response header.

response.addCookie( firstName );

response.addCookie( lastName );

// Set response content type

response.setContentType("text/html");

PrintWriter out = response.getWriter();

out.println("<b> Cookies are created and the cookies are:<b><br>");

out.println("<b>First Name</b>: "

+ request.getParameter("first\_name") + "\n"

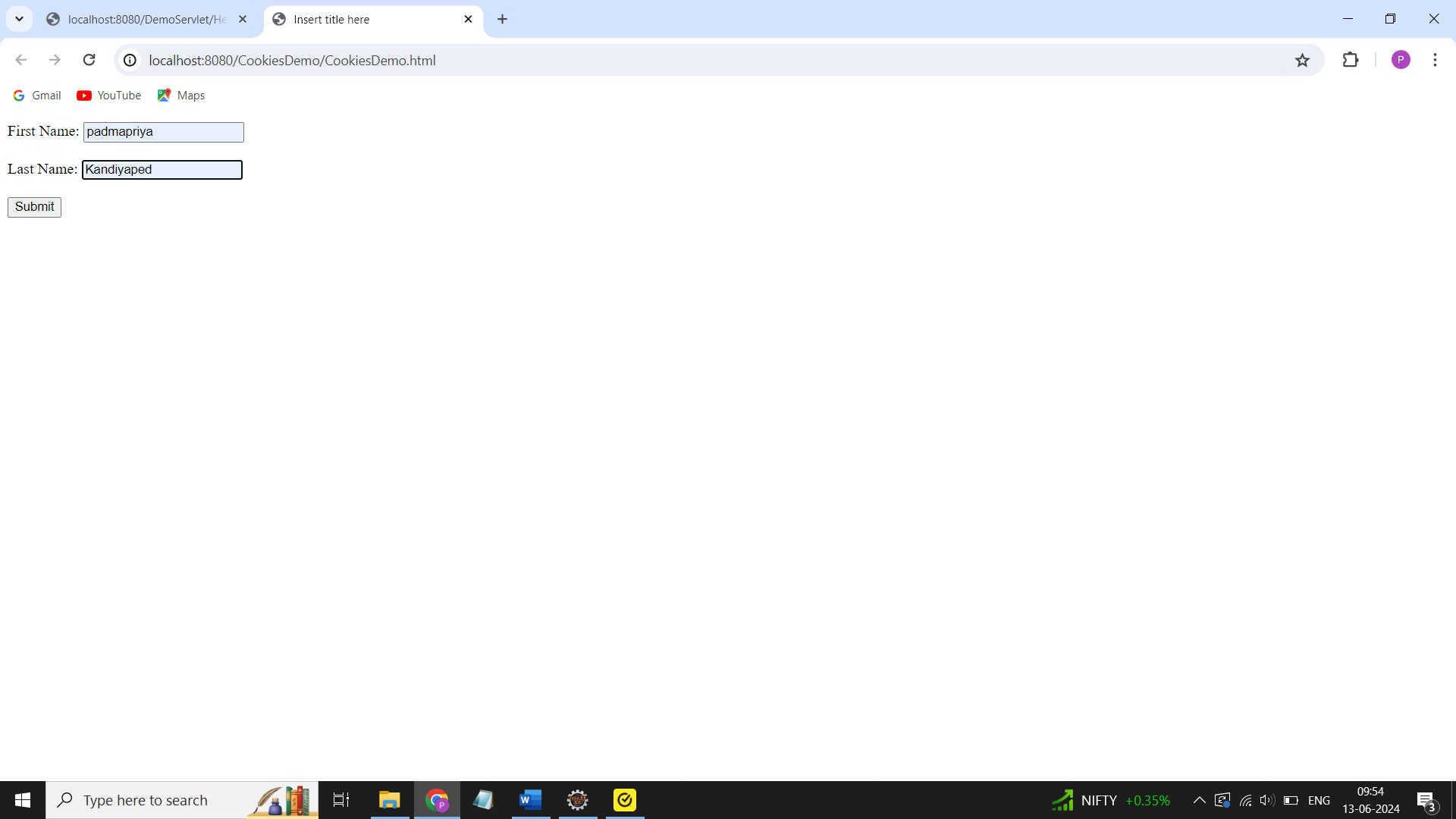
+ " <b>Last Name</b>: "

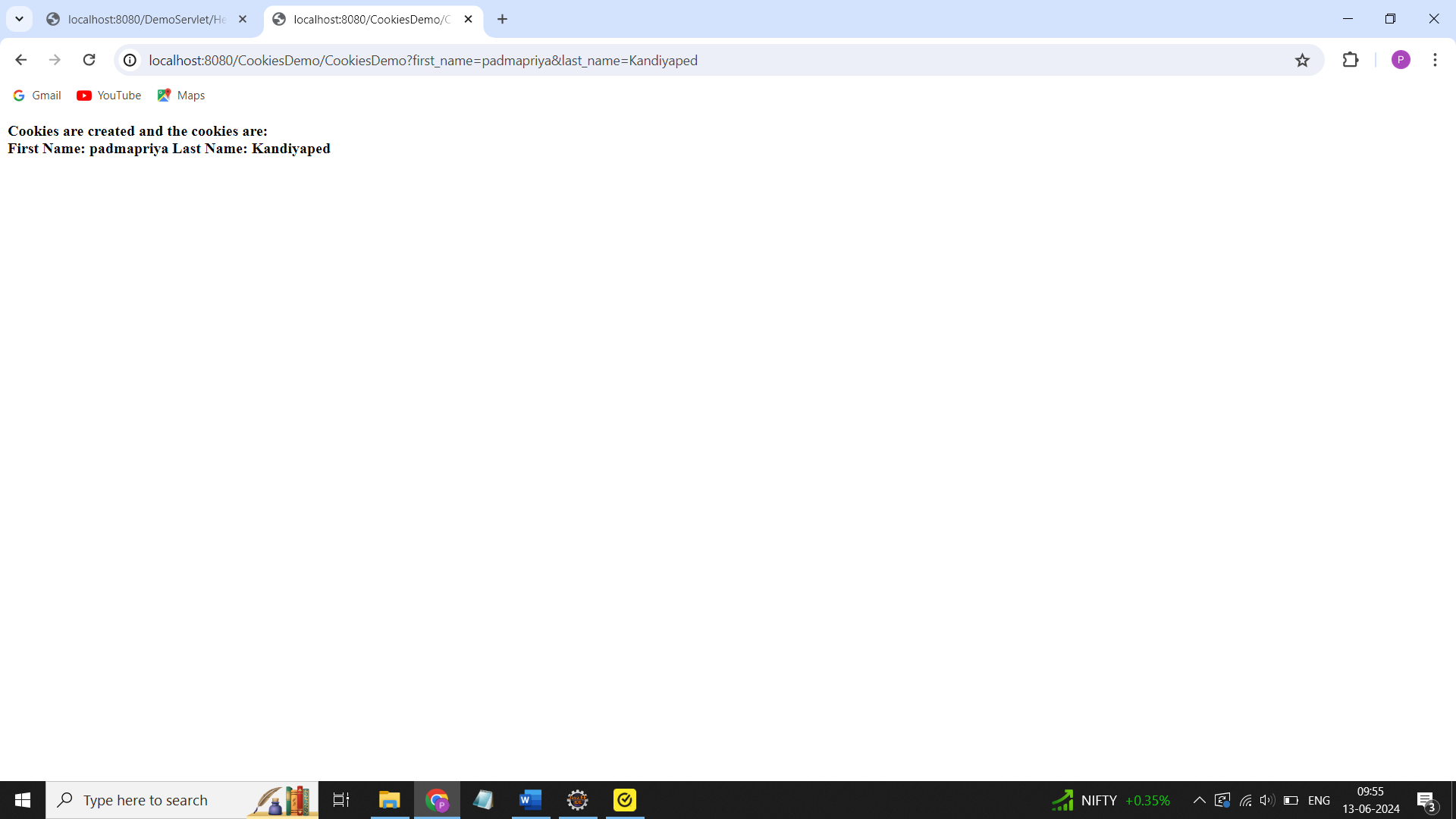
+ request.getParameter("last\_name") + "\n");

}

}

Output:





Week 8: Maintaining the transactional history of any user is very important. Explore the various session tracking mechanism using Sessions.

Sol:

import jakarta.servlet.ServletException;

import jakarta.servlet.http.HttpServlet;

import jakarta.servlet.http.HttpServletRequest;

import jakarta.servlet.http.HttpServletResponse;

import jakarta.servlet.http.HttpSession;

import java.io.IOException;

import java.io.PrintWriter;

import java.util.Date;

public class SessionTracking extends HttpServlet {

private static final long *serialVersionUID* = 1L;

public SessionTracking() {

super();

// TODO Auto-generated constructor stub

}

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

// Create a session object if it is already not created.

HttpSession s1 = request.getSession(true);

// Get session creation time.

Date createTime = new Date(s1.getCreationTime());

// Get last access time of this web page.

Date lastAccessTime = new Date(s1.getLastAccessedTime());

@SuppressWarnings("removal")

Integer visitCount = new ~~Integer~~(0);

String visitCountKey = new String("visitCount");

String userIDKey = new String("userID");

String userID = new String("ABCD");

// Check if this is new comer on your web page.

if (s1.isNew())

{

s1.setAttribute(userIDKey, userID);

}

else

{

visitCount = (Integer)s1.getAttribute(visitCountKey);

visitCount = visitCount + 1;

userID = (String)s1.getAttribute(userIDKey);

}

s1.setAttribute(visitCountKey, visitCount);

// Set response content type

response.setContentType("text/html");

PrintWriter out = response.getWriter();

out.println("<h1>session id is" + s1.getId() +

"<br>Creation Time"+ createTime +

"<br>last access time" + lastAccessTime +

"<br>User ID is" +userID +

"<br>Number of visits is" +

visitCount+"</h1>" );

}

}

Output:

