WTMA

Exp 1

1.html program:

<!DOCTYPE html>

<html>

<head>

<title>Hotspot</title>

<!-- Link to the external CSS file -->

<link rel=*"stylesheet"* type=*"text/css"* href=*"style.css"*>

<style>

**h1** {

text-align: *center*;

color: *#333*;

margin-top: *20px*;

}

</style>

</head>

<body>

<!-- Inline style for the image itself (optional to override or enhance external styles) -->

<h1>Interactive Map of India</h1>

<img src=*"https://www.mapsofindia.com/images2/india-map-2019.jpg"* usemap=*"#indiamap"* width=*"900"* height=*"720"* alt=*"Map of India"*>

<map name=*"indiamap"*>

<area shape=*"rect"* coords=*"324,578,407,535"* alt=*"Tamil Nadu"* href=*"https://en.wikipedia.org/wiki/Tamil\_Nadu"*>

<area shape=*"rect"* coords=*"343,655,398,599"* alt=*"Tamil Nadu"* href=*"https://en.wikipedia.org/wiki/Tamil\_Nadu"*>

<area shape=*"rect"* coords=*"407,538,395,602"* alt=*"Tamil Nadu"* href=*"https://en.wikipedia.org/wiki/Tamil\_Nadu"*>

<area shape=*"rect"* coords=*"346,653,397,605"* alt=*"Tamil Nadu"* href=*"https://en.wikipedia.org/wiki/Tamil\_Nadu"*>

<area shape=*"rect"* coords=*"406,537,324,579"* alt=*"Tamil Nadu"* href=*"https://en.wikipedia.org/wiki/Tamil\_Nadu"*>

</map>

</body>

</html>

2.css program:

@charset *"ISO-8859-1"*;

**body** {

font-family: *Arial,* *sans-serif*;

margin: *0*;

padding: *0*;

background-color: *#f4f4f4*;

}

**img** {

display: *block*;

margin: *0* *auto*;

border-radius: *8px*;

box-shadow: *0* *4px* *6px* *rgba(0,* *0,* *0,* *0.1)*;

}

/\* Styling for the map areas \*/

**area** {

outline: *none*;

border: *none*;

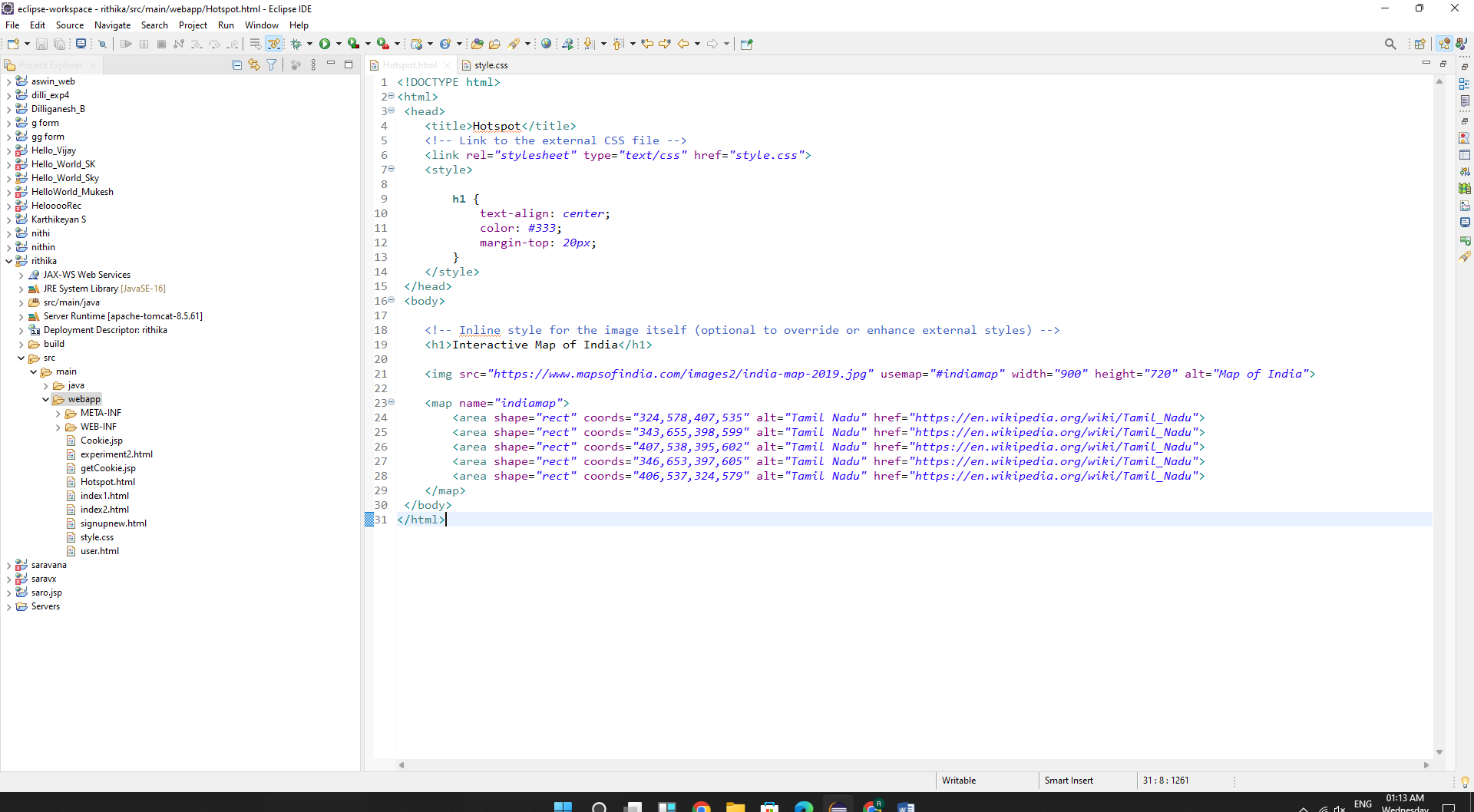
transition: *transform* *0.3s* *ease-in-out*;

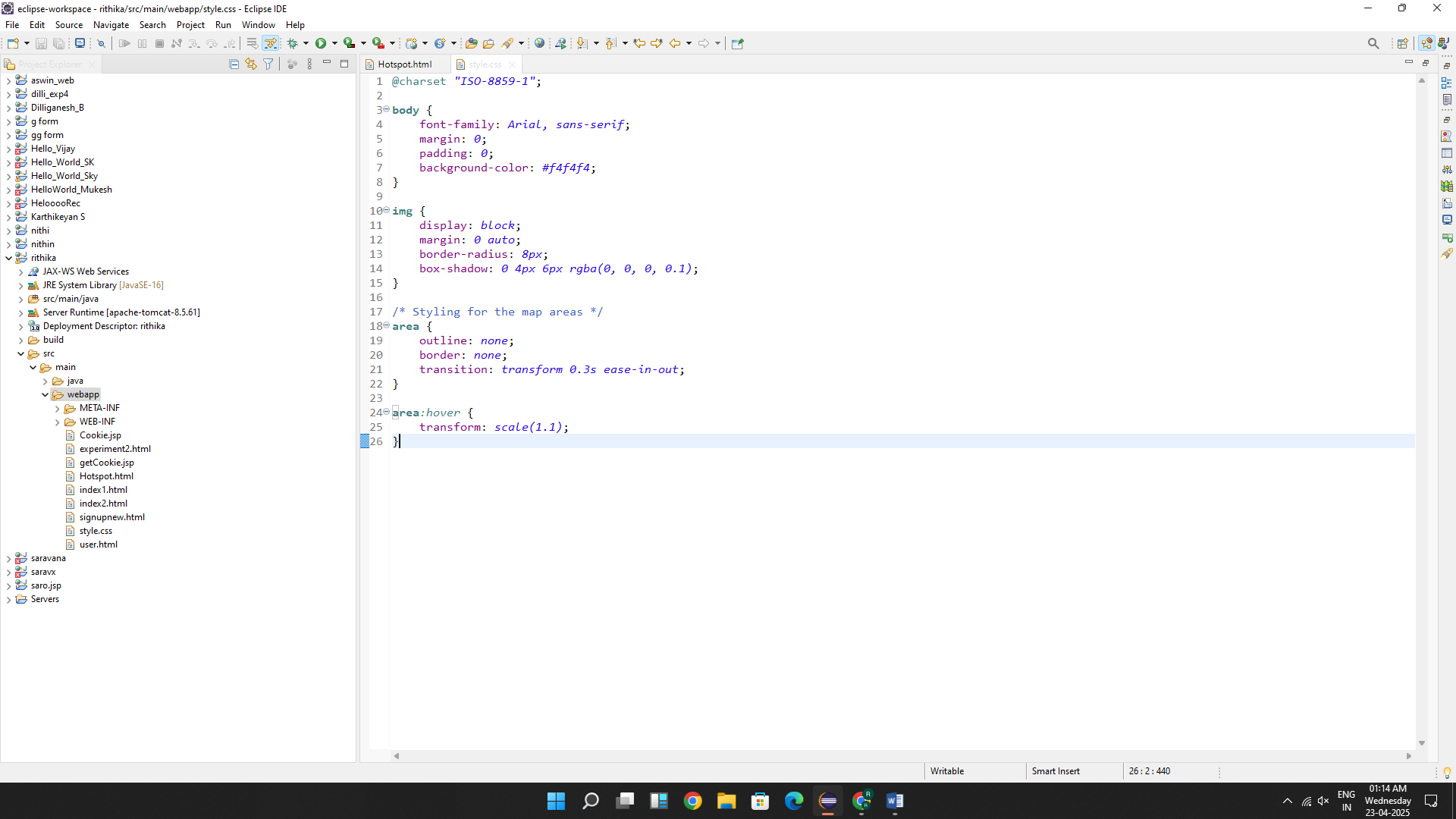
}

**area***:hover* {

transform: *scale(1.1)*;

}





Output:



Result:Thus the program for HTML and CSS executed successfully

EXP 2

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<form onsubmit="return validateForm()">

<label for=*"firstName"*>First Name:</label>

<input type=*"text"* id=*"firstName"* name=*"firstName"* required><br>

<label for=*"lastName"*>Last Name:</label>

<input type=*"text"* id=*"lastName"* name=*"lastName"* required><br>

<label for=*"password"*>Password:</label>

<input type=*"password"* id=*"password"* name=*"password"* required><br>

<label for=*"email"*>Email:</label>

<input type=*"email"* id=*"email"* name=*"email"* required><br>

<label for=*"mobileNumber"*>Mobile Number:</label>

<input type=*"text"* id=*"mobileNumber"* name=*"mobileNumber"* required><br>

<label for=*"address"*>Address:</label>

<textarea id=*"address"* name=*"address"* required></textarea><br>

<input type=*"submit"* value=*"Register"*>

</form>

<script type=*"text/javascript"*>

**function** validateForm() {

// Validate First Name

**var** firstName = document.getElementById("firstName").value;

**var** namePattern = /^[A-Za-z]+$/;

**if** (firstName.length < 6 || !namePattern.test(firstName)) {

alert("First Name must contain only alphabets and be at least 6 characters long.");

**return** **false**;

}

// Validate Password

**var** password = document.getElementById("password").value;

**if** (password.length < 6) {

alert("Password must be at least 6 characters long.");

**return** **false**;

}

// Validate Email

**var** email = document.getElementById("email").value;

**var** emailPattern = /^[a-zA-Z0-9.\_-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,6}$/;

**if** (!emailPattern.test(email)) {

alert("Please enter a valid email address.");

**return** **false**;

}

// Validate Mobile Number

**var** mobileNumber = document.getElementById("mobileNumber").value;

**var** mobilePattern = /^\d{10}$/;

**if** (!mobilePattern.test(mobileNumber)) {

alert("Mobile number must contain exactly 10 digits.");

**return** **false**;

}

// Validate Last Name and Address

**var** lastName = document.getElementById("lastName").value;

**var** address = document.getElementById("address").value;

**if** (lastName.trim() === "" || address.trim() === "") {

alert("Last Name and Address must not be empty.");

**return** **false**;

}

// If all validations pass

**return** **true**;

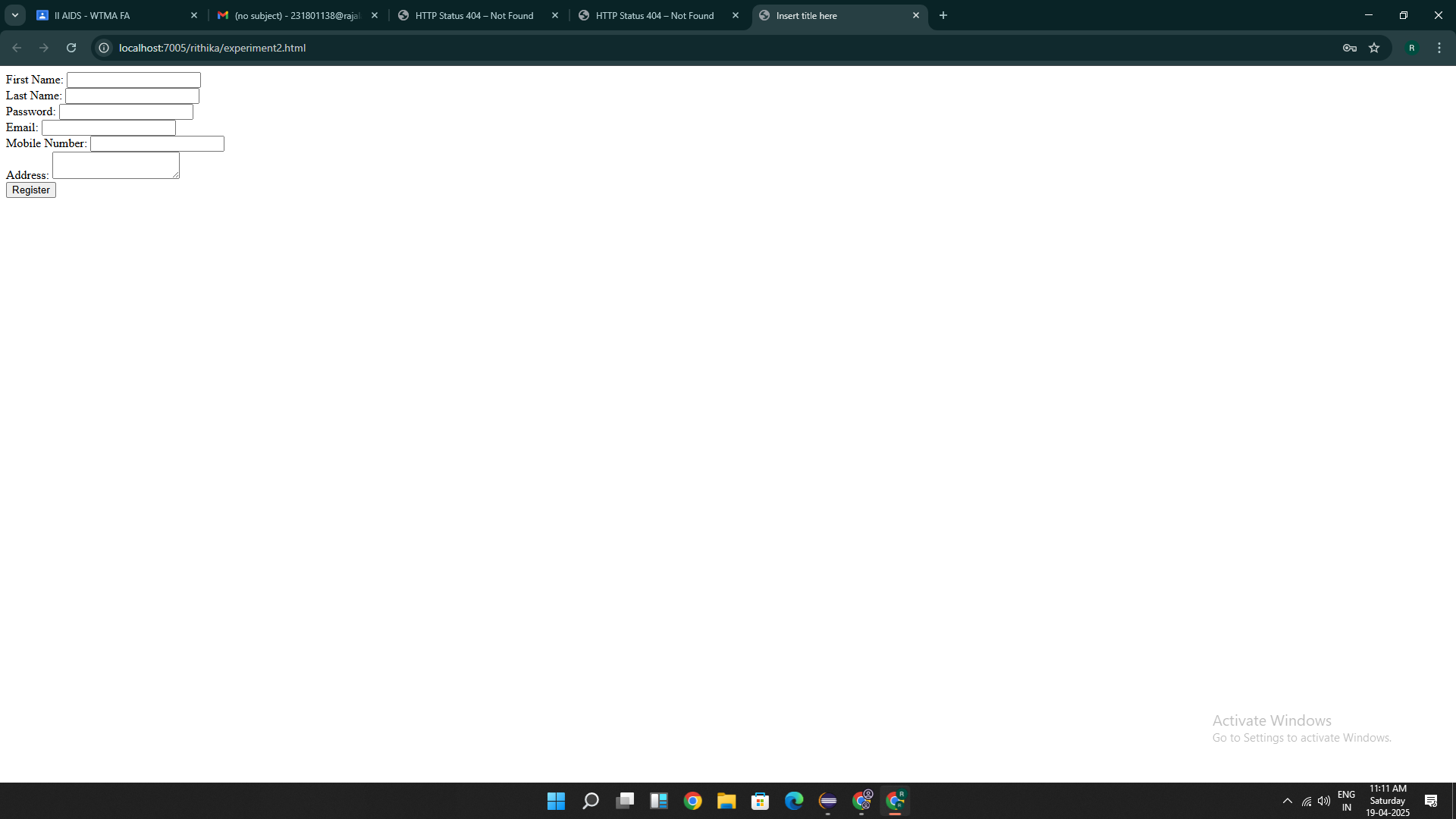
}

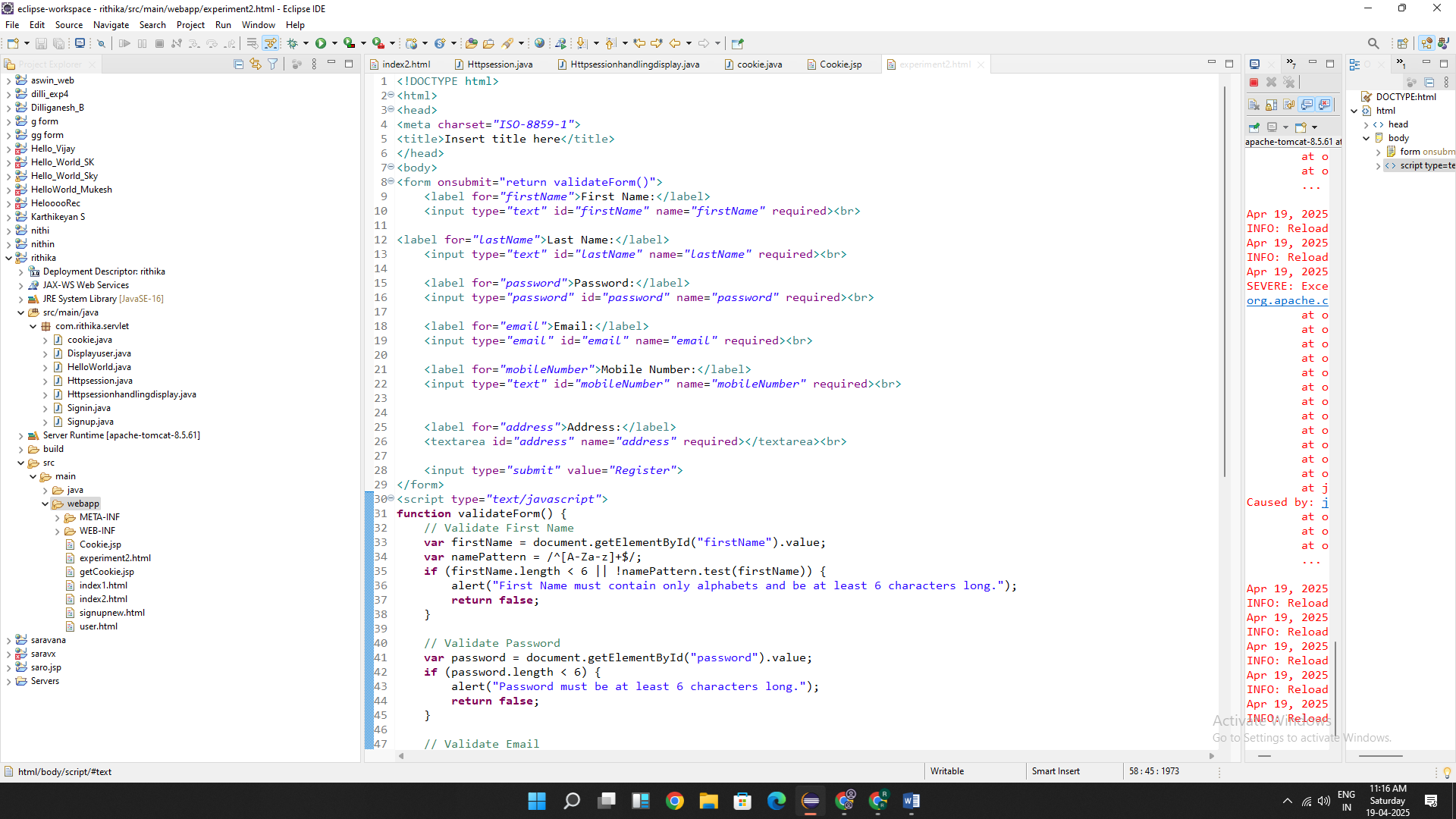
</script>

</body>

</html>

OUTPUT:





Result:Thus the programs are executed successfully

EXP 3

PROGRAM:

package com.rithika.servlet;

import java.io.IOException;

import java.io.PrintWriter;

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 HelloWorld

\*/

@WebServlet("/HelloWorld")

public class HelloWorld extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public HelloWorld() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

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

\*/

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

PrintWriter p=response.getWriter();

p.println("HelloWorld");

}

/\*\*

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

\*/

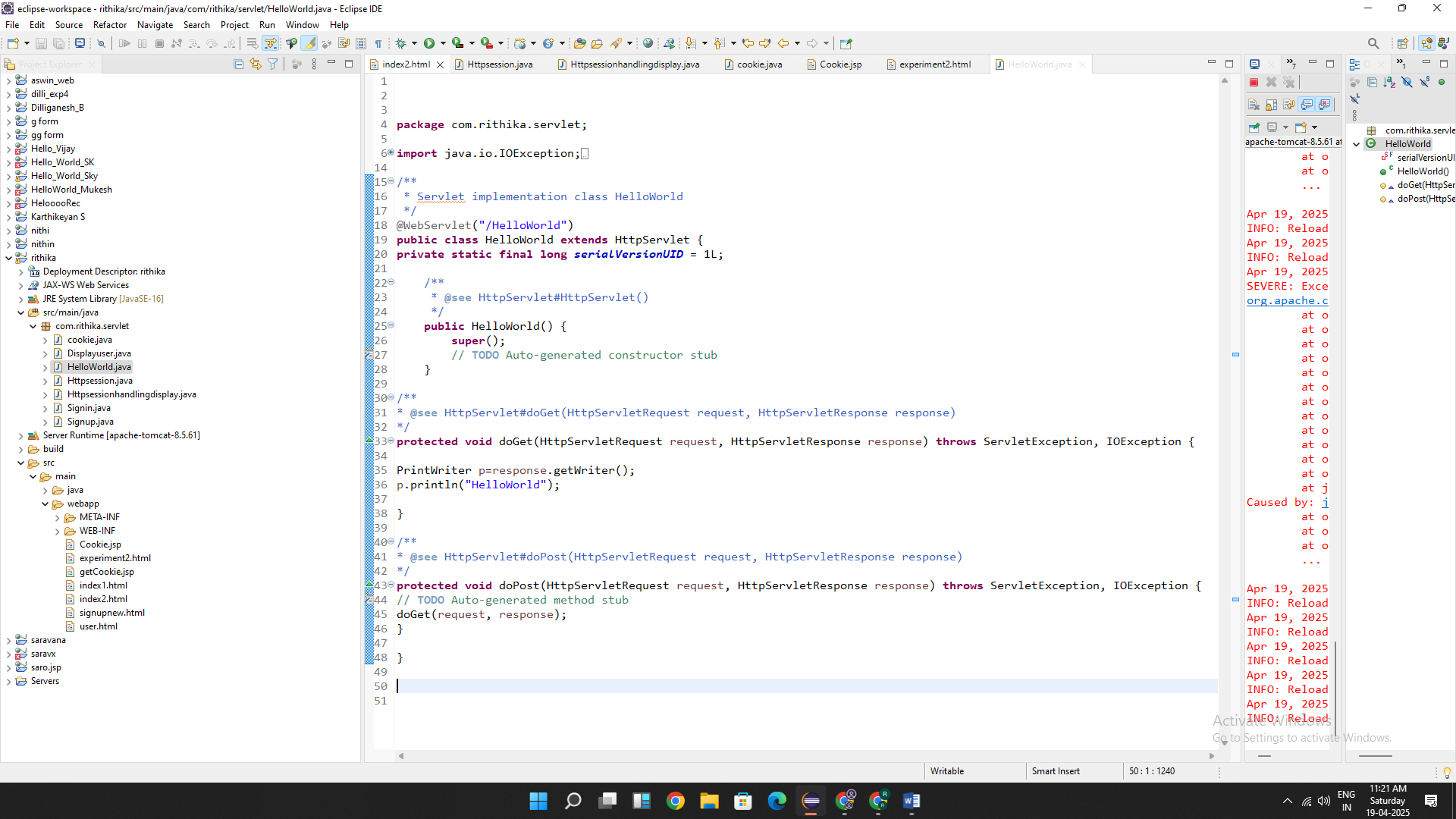
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

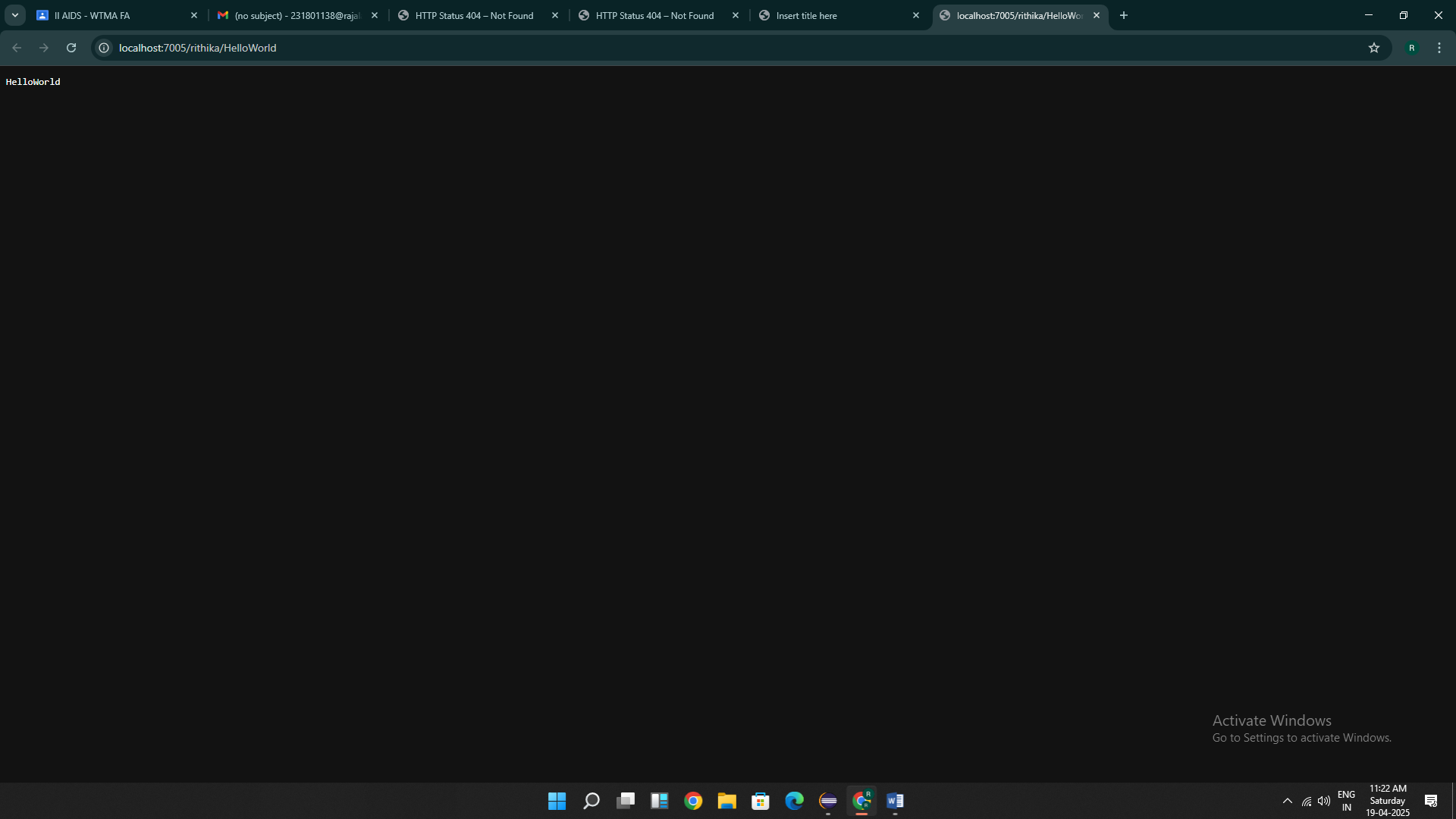
// TODO Auto-generated method stub

doGet(request, response);

}

}





RESULT:Thus the programs are executed successfully

EXP 4

PROGRAM:

1.Html

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<form action =*"/rithika/DisplayUser"*>

<p>

<label for=*"username"*>Username:</label>

<input type=*"text"* id=*"username"* name=*"username"*>

</p>

<p>

<label for=*"age"*>Age:</label>

<input type=*"number"* id=*"age"* name=*"age"*>

</p>

<input type=*"submit"* value=*"Accept"* name=*"submit"*>

</form>

</body>

</html>

2.java

package com.rithika.servlet;

import java.io.IOException;

import java.io.PrintWriter;

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 Displayuser

\*/

@WebServlet("/DisplayUser")

public class Displayuser extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public Displayuser() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

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

\*/

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

String a=request.getParameter("user");

String b=request.getParameter("age");

int age=Integer.parseInt(b);

PrintWriter p=response.getWriter();

if(a.isEmpty()) {

p.println("Username sholud not be empty !");

}

else if(age<1) {

p.println("Age should not be less than or equal to zero !");

}

else {

p.println("Username : "+a);

p.println("Age : "+age);

}

}

/\*\*

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

\*/

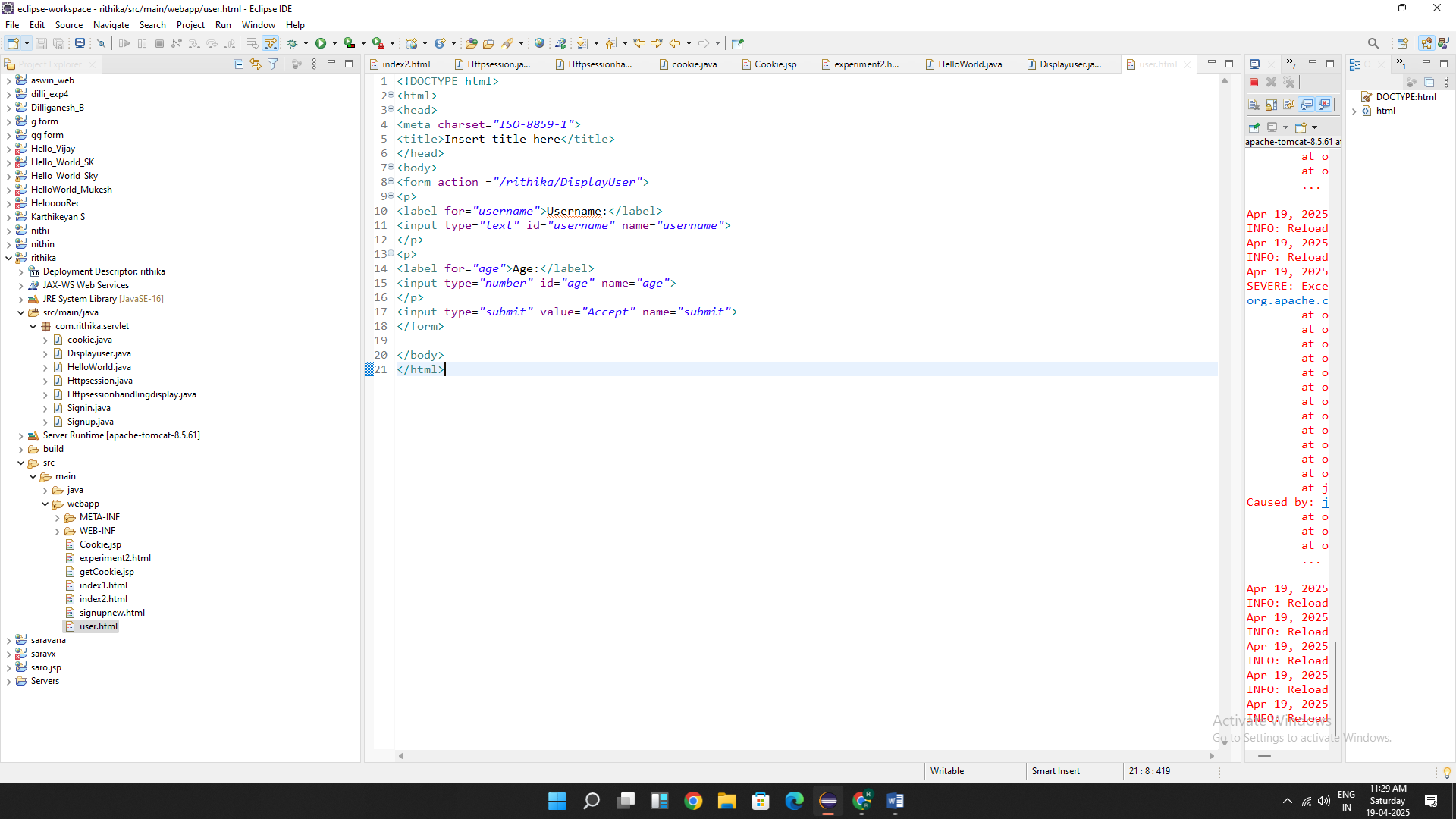
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

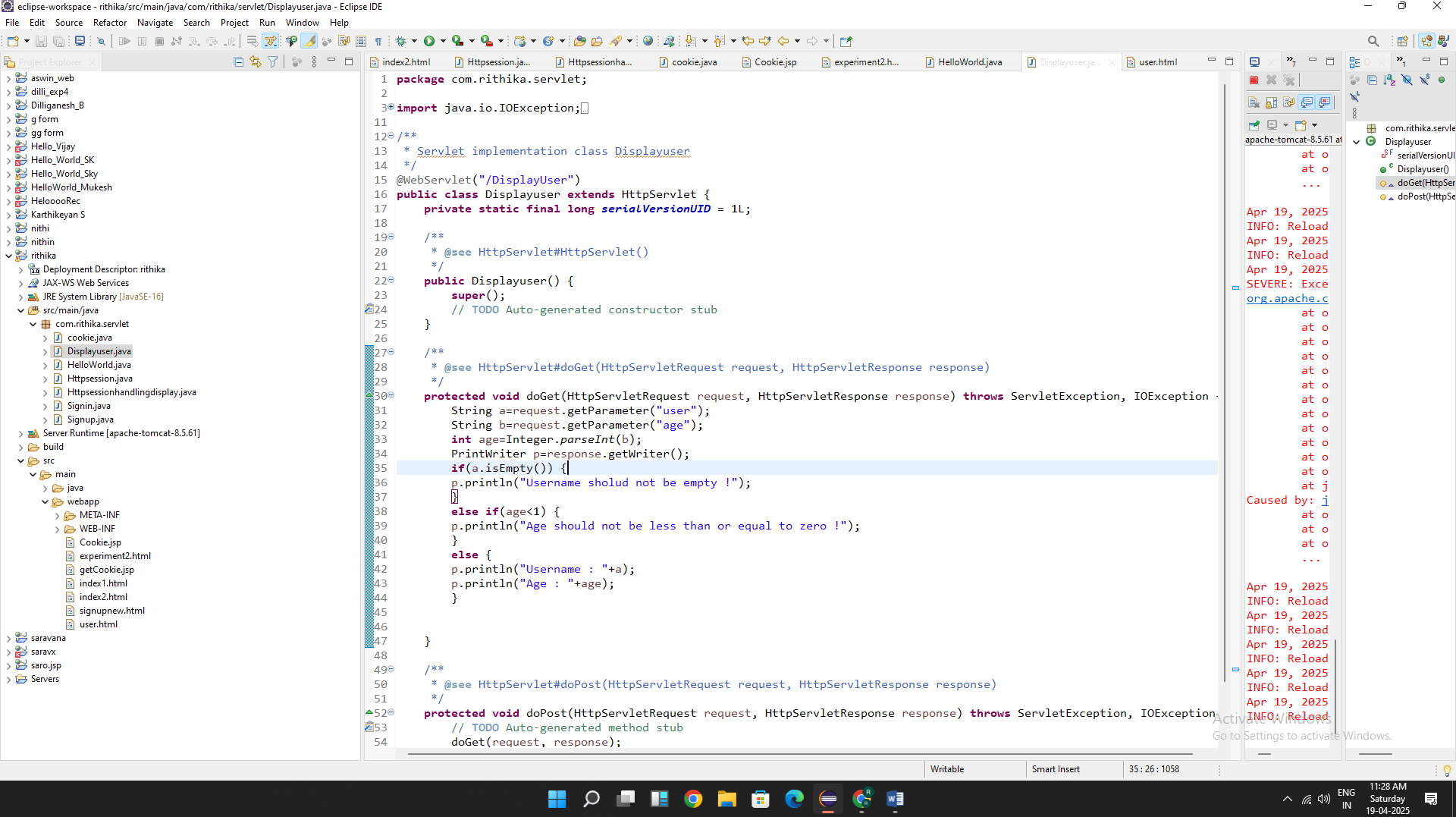
// TODO Auto-generated method stub

doGet(request, response);

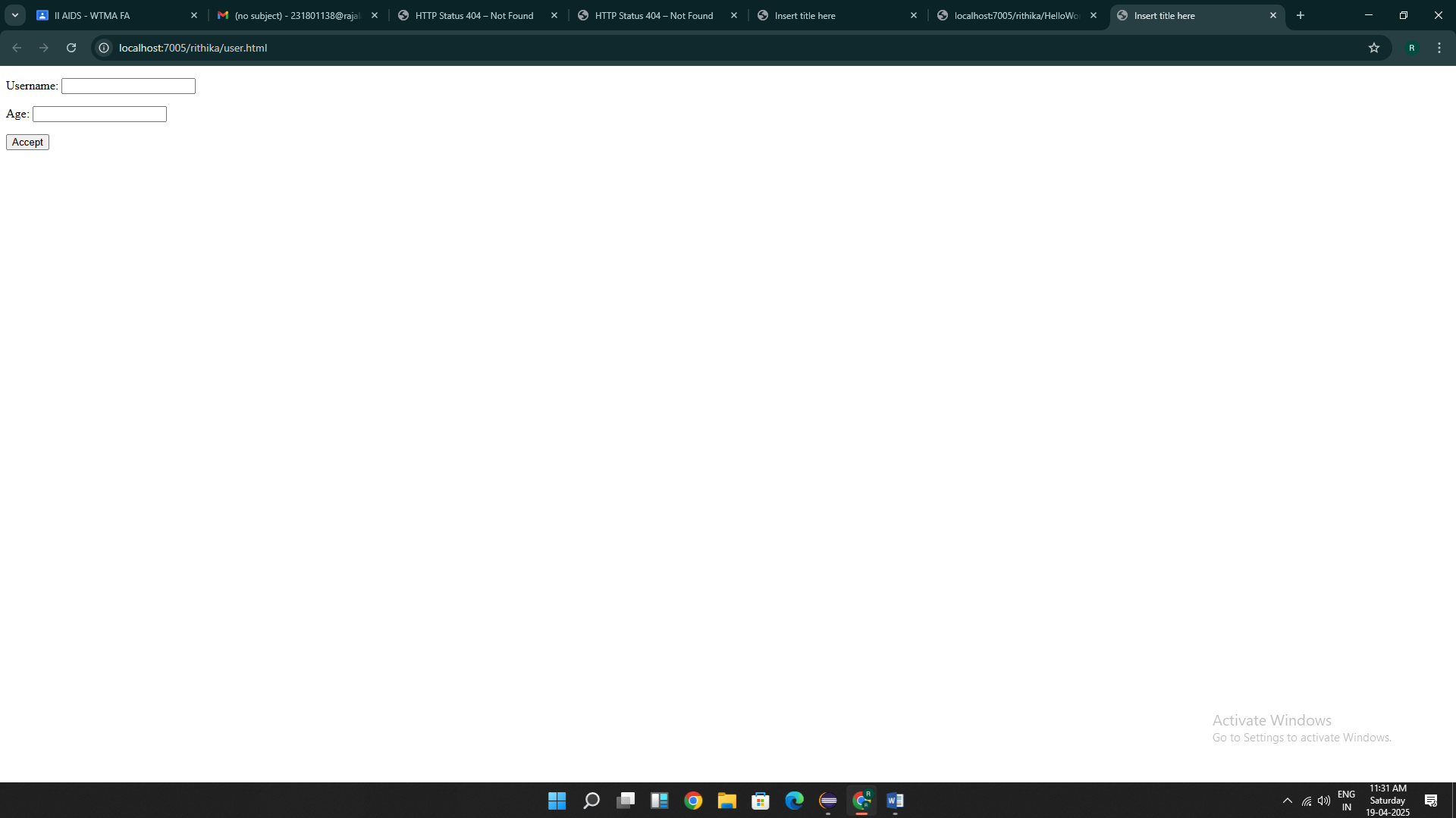
}

}





Output:



Result : Thus the programs are executed successfully

Exp 5

Program:

1.html

<!DOCTYPE html>

<html>

<head>

<title>Login Form</title>

</head>

<body>

<h2>Login Form</h2>

<form action=*"/rithika/Signup"* method=*"post"*>

<label for=*"Firstname"*>Firstname:</label>

<input type=*"text"* id=*"Firstname"* name=*"Firstname"* required><br><br>

<label for=*"lastname"*>Lastname:</label>

<input type=*"text"* id=*"lastname"* name=*"lastname"* required><br><br>

<label for=*"dob"*>Date of Birth:</label>

<input type=*"date"* id=*"dob"* name=*"dob"* required><br><br>

<label for=*"gender"*>Gender:</label>

<input type=*"radio"* id=*"male"* name=*"gender"* value=*"Male"* required>

<label for=*"male"*>Male</label>

<input type=*"radio"* id=*"female"* name=*"gender"* value=*"Female"* required>

<label for=*"female"*>Female</label><br><br>

<label for=*"address"*>Address:</label>

<textarea id=*"address"* name=*"address"* required></textarea><br><br>

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

</form>

</body>

</html>

2.

package com.rithika.servlet;

import java.io.IOException;

import java.io.PrintWriter;

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 Signup

\*/

@WebServlet("/Signup")

public class Signup extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public Signup() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

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

\*/

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

String a=request.getParameter("Fname");

String b=request.getParameter("Lname");

String c=request.getParameter("dob");

String d=request.getParameter("Gender");

String e=request.getParameter("address");

PrintWriter p=response.getWriter();

p.println(a);

p.println(b);

p.println(c);

p.println(d);

p.println(e);

}

/\*\*

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

\*/

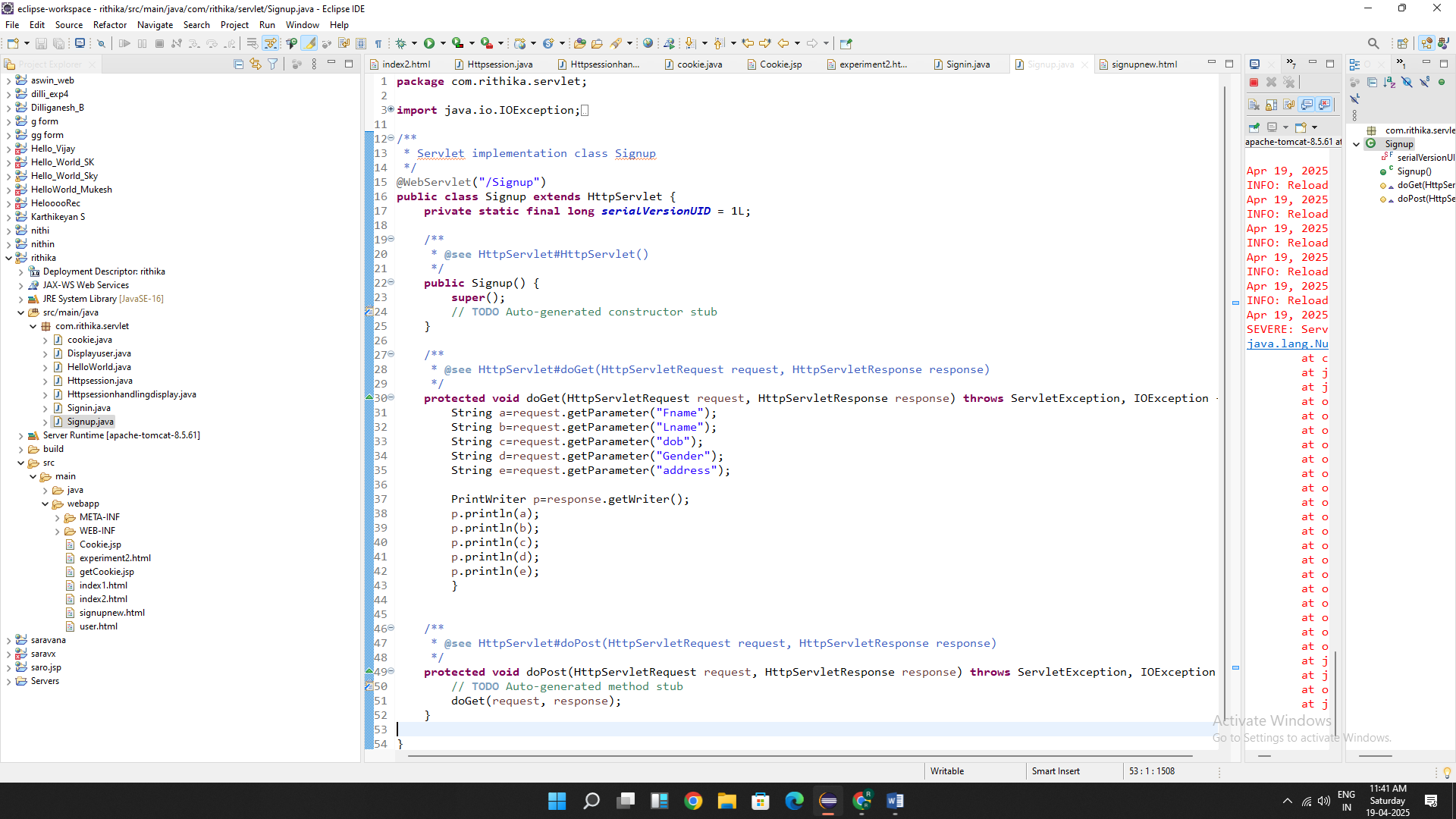
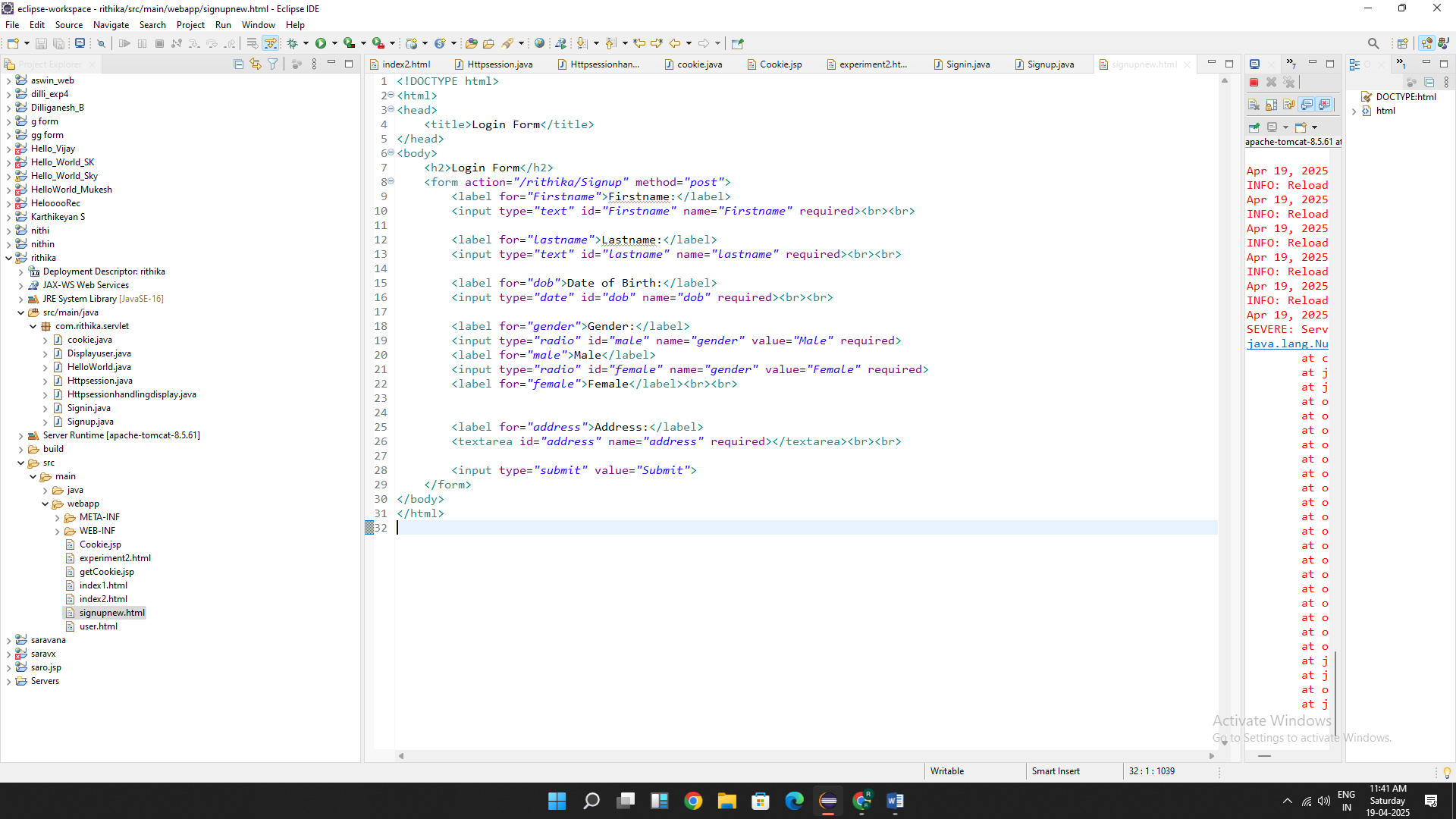
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

doGet(request, response);

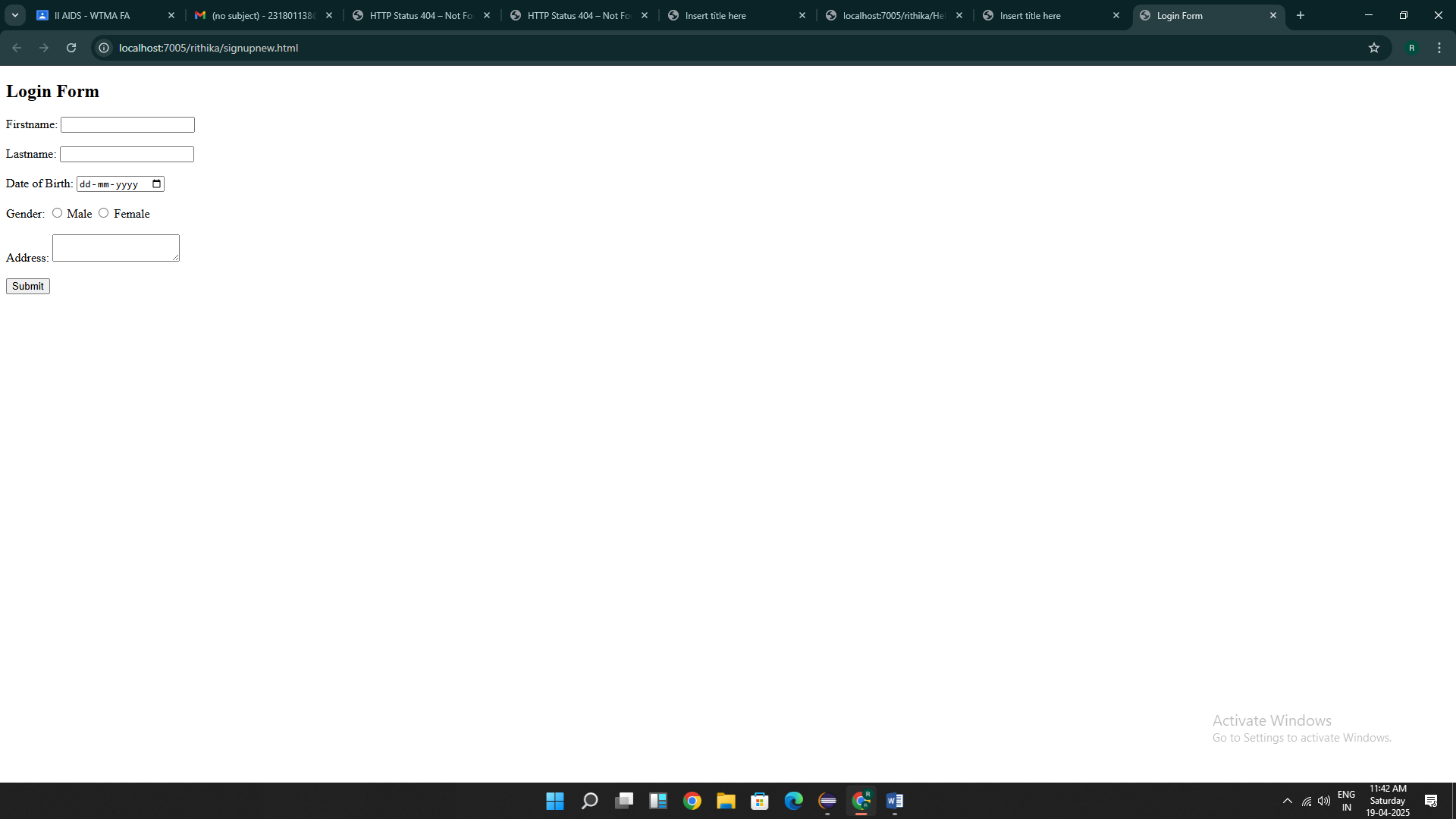
}

}



OUTPUT:





Result : Thus the programs are executed successfully

EXP 6

PROGRAM:

1.HTML:

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>HTTP Session</title>

</head>

<body>

<h1>Form Handle By HTTP Session</h1>

<form action=*"/rithika/Httpsession"* method=*"POST"* style="margin-left: *50px*">

<label>Username: </label><input type=*"text"* name=*"username"*/><br><br>

<label>Password: </label><input type=*"password"* name=*"password"*/><br><br>

<input type=*"submit"* value=*"Login"* style="margin-left: *130px*"/>

</form>

</body>

</html>

2. **package** com.rithika.servlet;

**import** java.io.IOException;

**import** java.io.PrintWriter;

**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** javax.servlet.http.HttpSession;

/\*\*

\* Servlet implementation class HttpSessionHandle

\*/

@WebServlet("/Httpsession")

**public** **class** Httpsession **extends** HttpServlet {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

@SuppressWarnings("deprecation")

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

String username = "WTMA";

String password = "WTMA";

PrintWriter p = response.getWriter();

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

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

**if**(user.equals(username) && pass.equals(password))

{

HttpSession session = request.getSession(**true**);

session.~~putValue~~("username", user);

response.sendRedirect(request.getContextPath() + "/Httpsessionhandlingdisplay");

}

**else**

{

p.println("fail");

}

}

}

3. **package** com.rithika.servlet;

**import** java.io.IOException;

**import** java.io.PrintWriter;

**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** javax.servlet.http.HttpSession;

/\*\*

\* Servlet implementation class HttpSessionHandleDisplay

\*/

@WebServlet("/Httpsessionhandlingdisplay")

**public** **class** Httpsessionhandlingdisplay **extends** HttpServlet {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

/\*\*

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

\*/

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

PrintWriter p = response.getWriter();

HttpSession session = request.getSession(**true**);

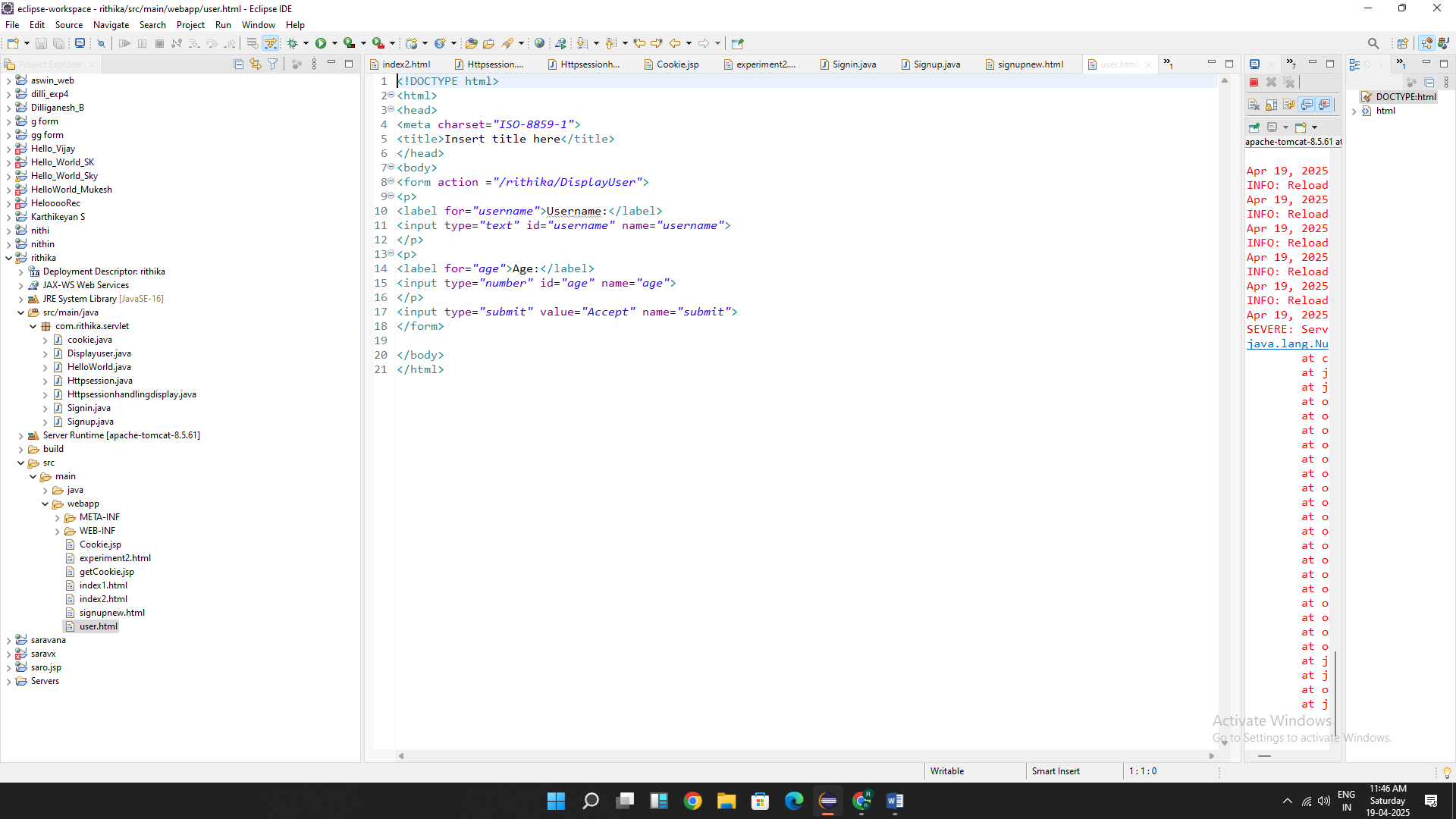
@SuppressWarnings("deprecation")

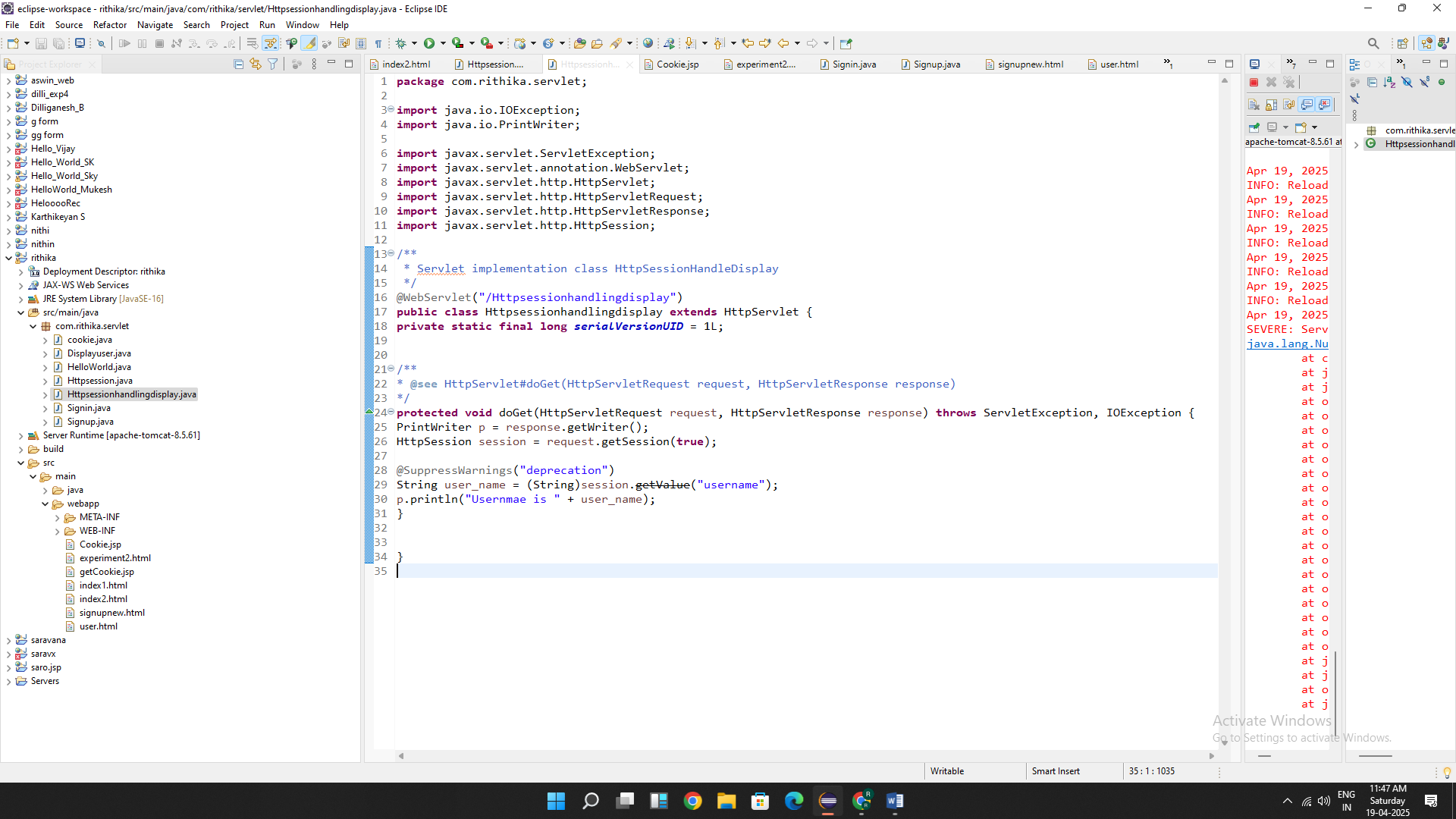
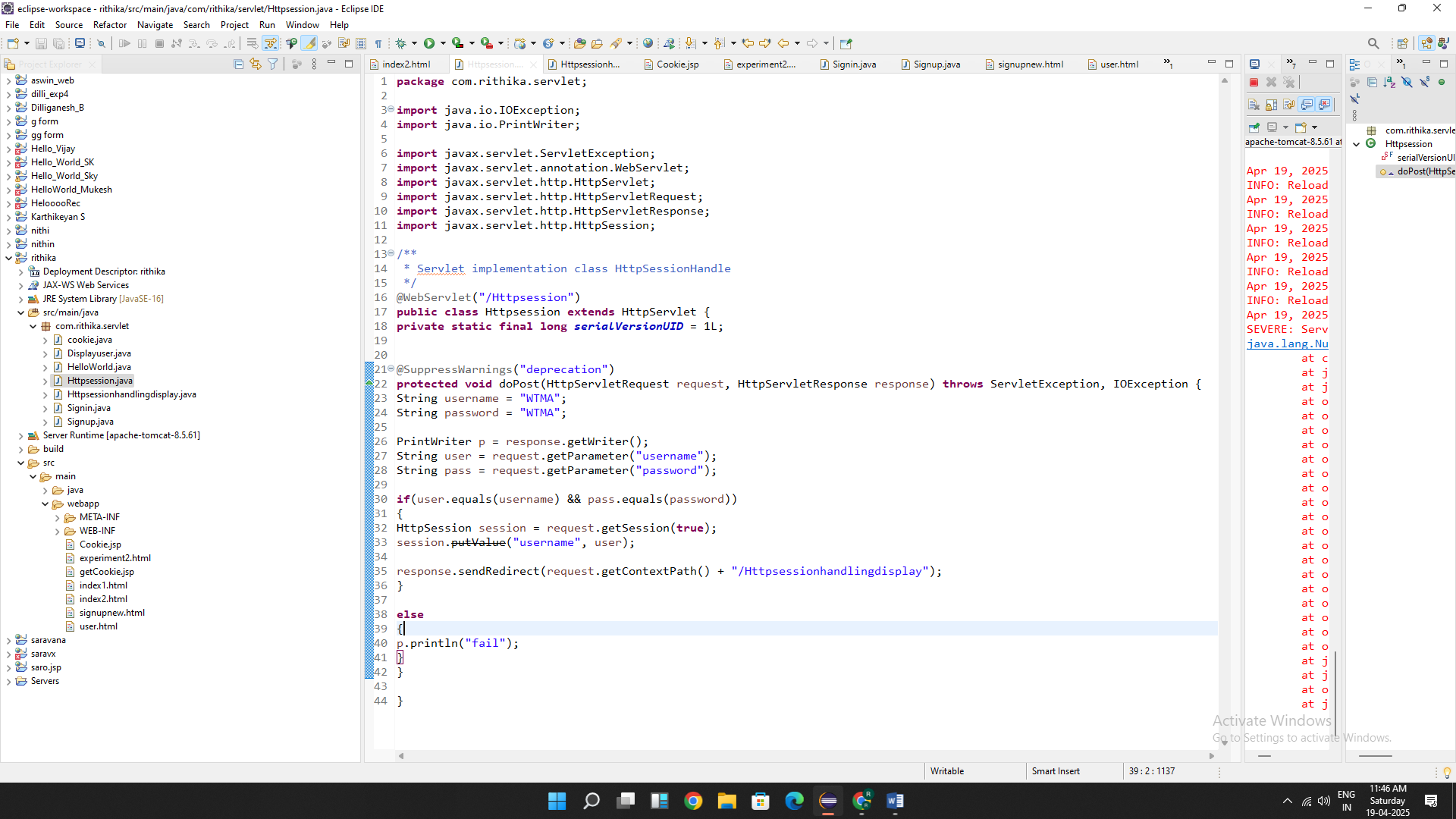
String user\_name = (String)session.~~getValue~~("username");

p.println("Usernmae is " + user\_name);

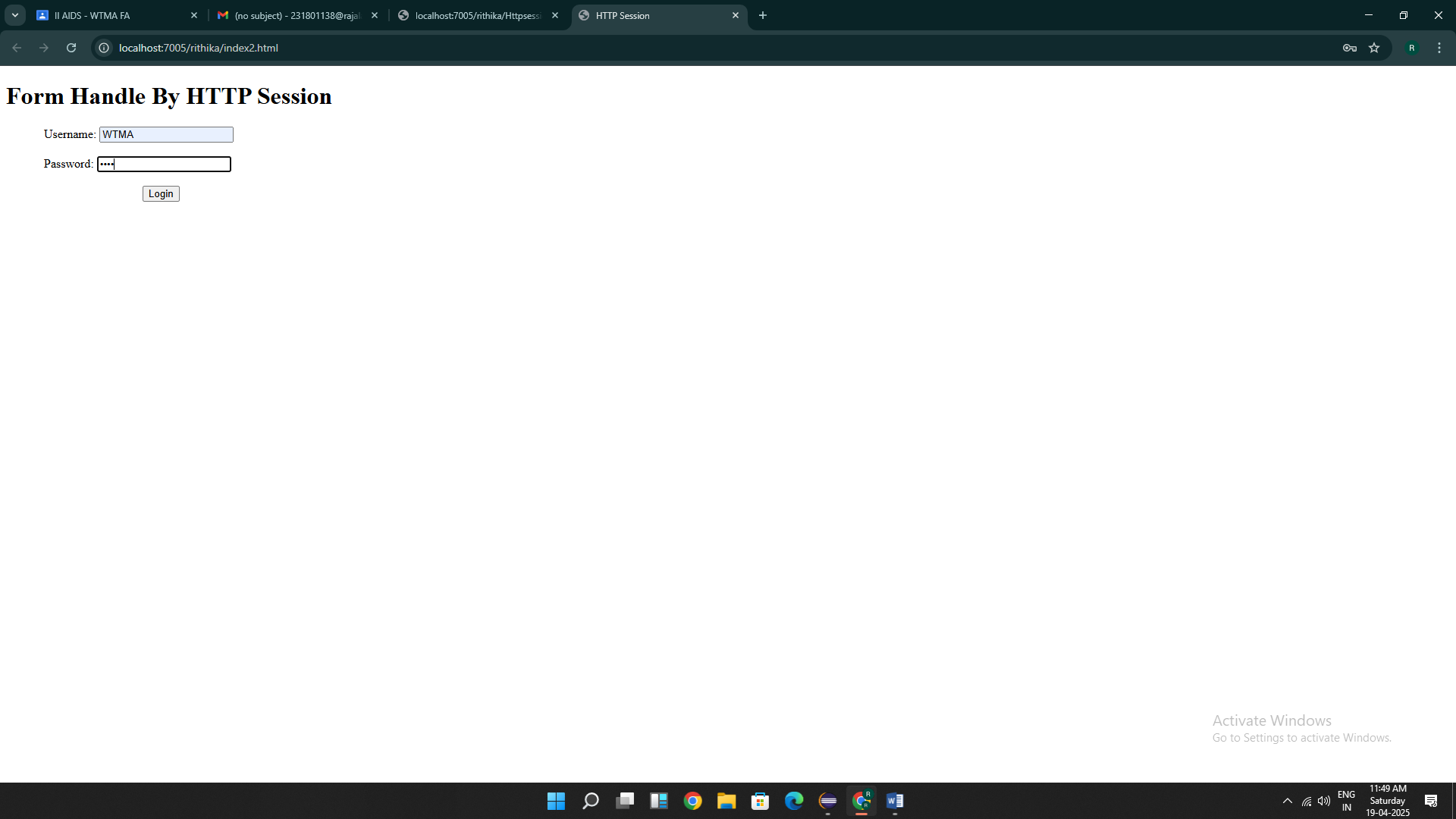
}

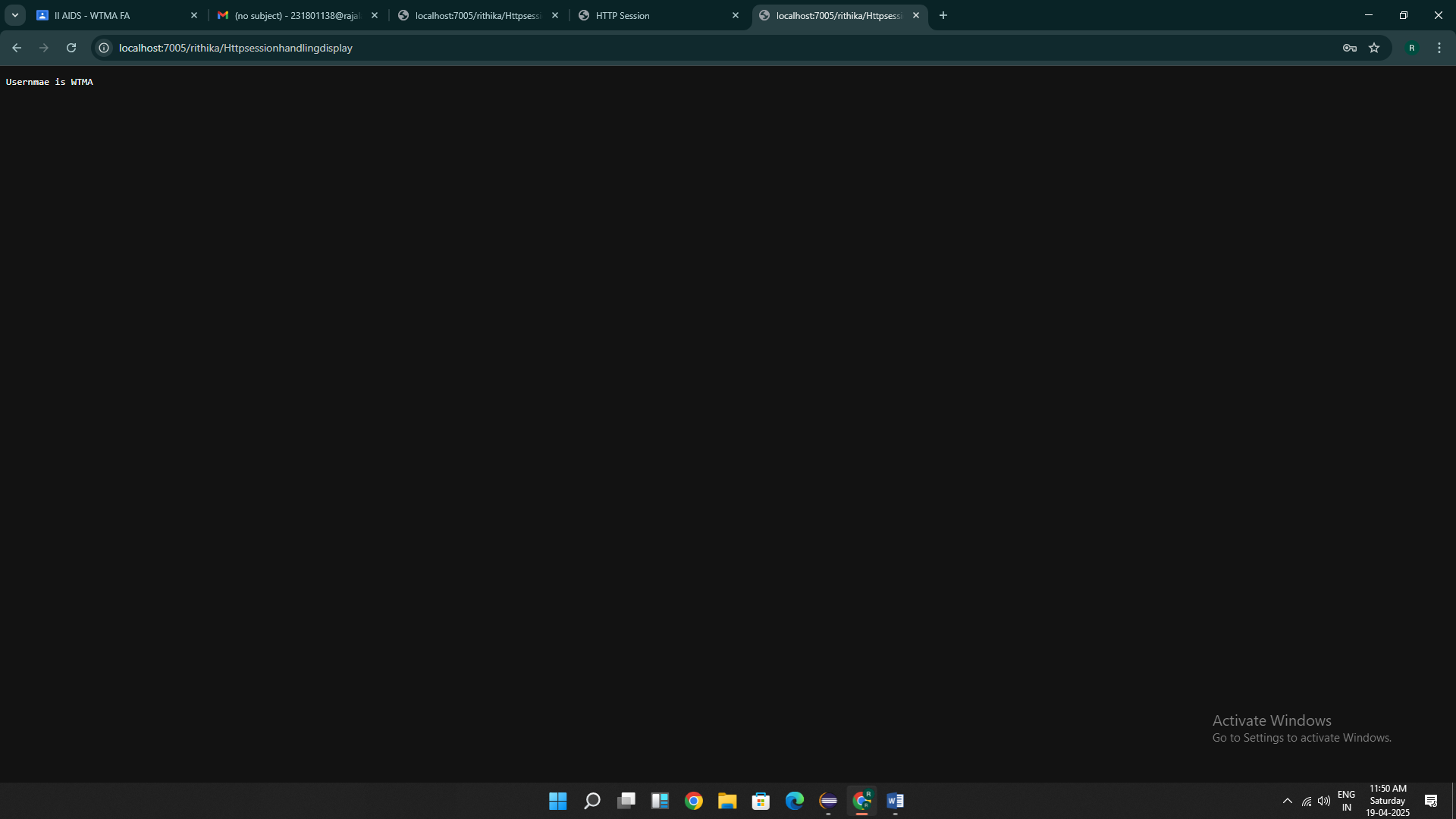
}





OUTPUT:





Result : Thus the programs are executed successfully

EXP 7:

1.

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

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Cookies</title>

</head>

<body>

<form method = *"post"* >

<p>

<label>Username: </label>

<input type=*"text"* name=*"username"*>

</p>

<p>

<label> password: </label>

<input type=*"password"* name = *"password"*>

</p>

<button type=*"submit"*> Submit </button>

</form>

<%

// Request form method == Post

**if** ("POST".equals(request.getMethod())) {

// Get submitted preferences

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

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

**if**(user.equals("admin") && pass.equals("admin123") ){

Cookie username = **new** Cookie("username", user);

username.setMaxAge(60\*60);

username.setPath("/");

response.addCookie(username);

response.sendRedirect("getCookie.jsp");

}

}

// Get current preference values

%>

</body>

</html>

2.

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

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<%

Cookie[] cookies = request.getCookies();

String cookieName = "username";

**if** (cookies != **null**) {

**for** (Cookie cookie : cookies) {

**if** (cookieName.equals(cookie.getName())) {

System.out.println("username "+cookie.getName());

out.println(cookie.getValue());

}

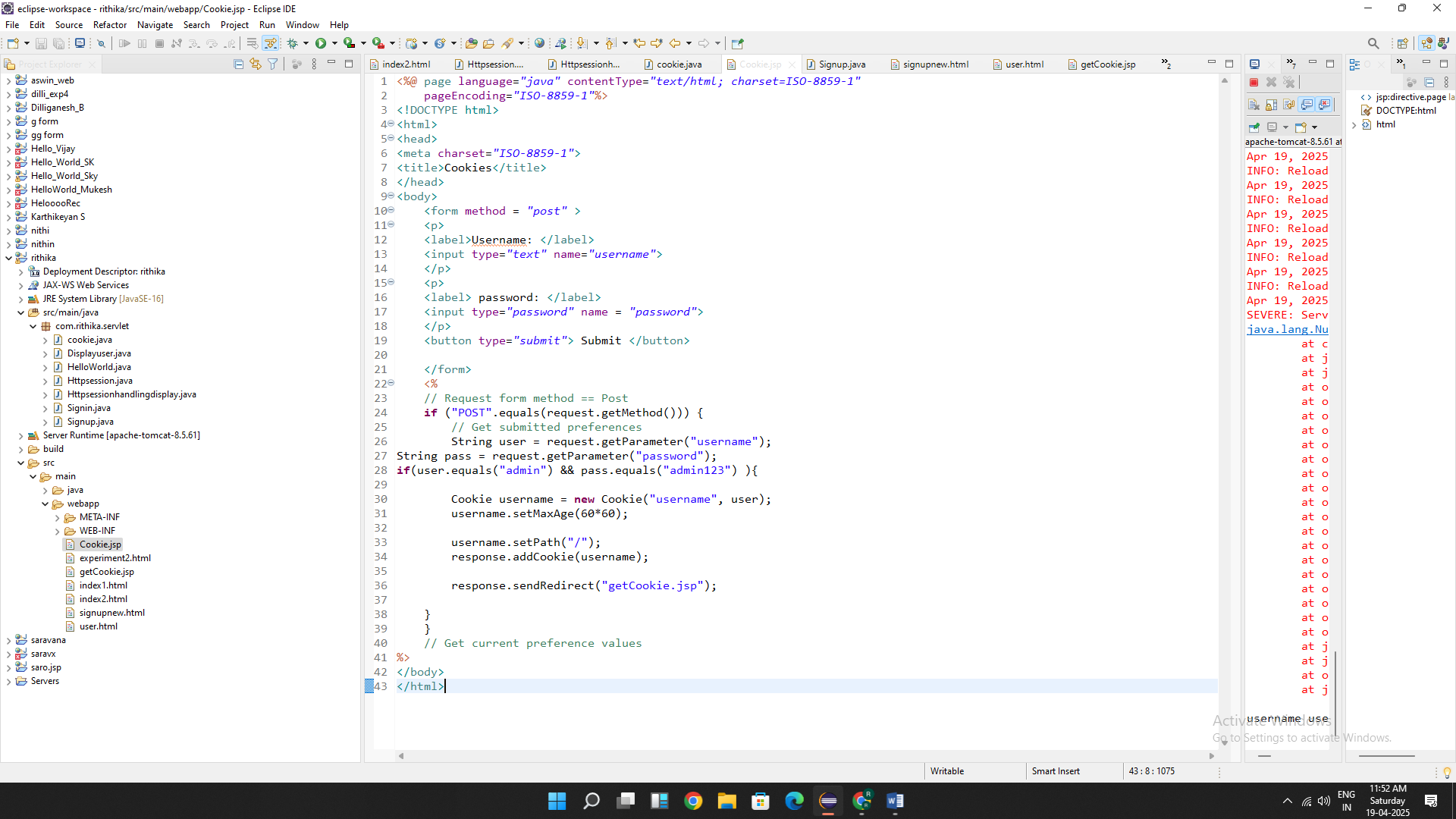
}

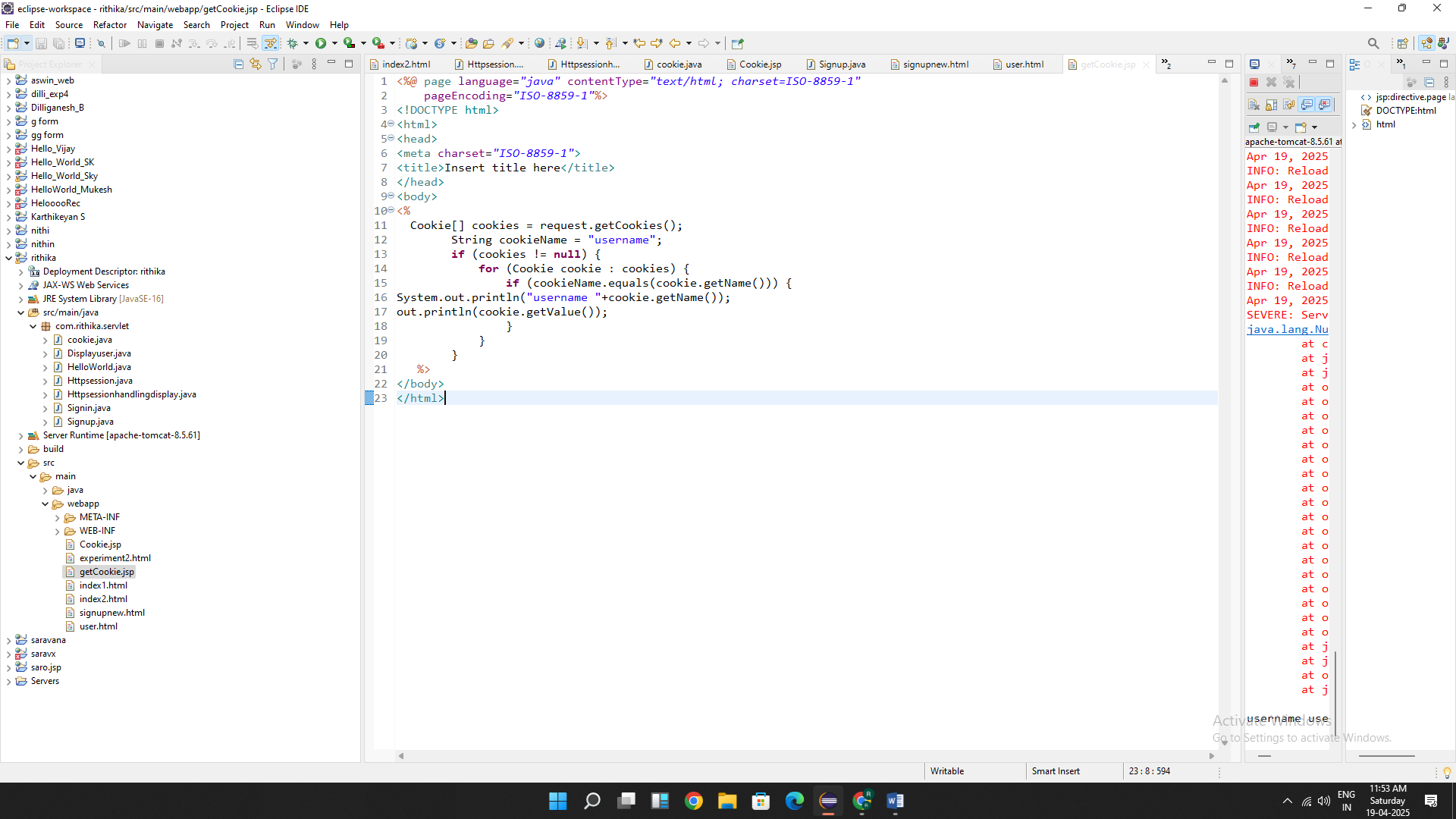
}

%>

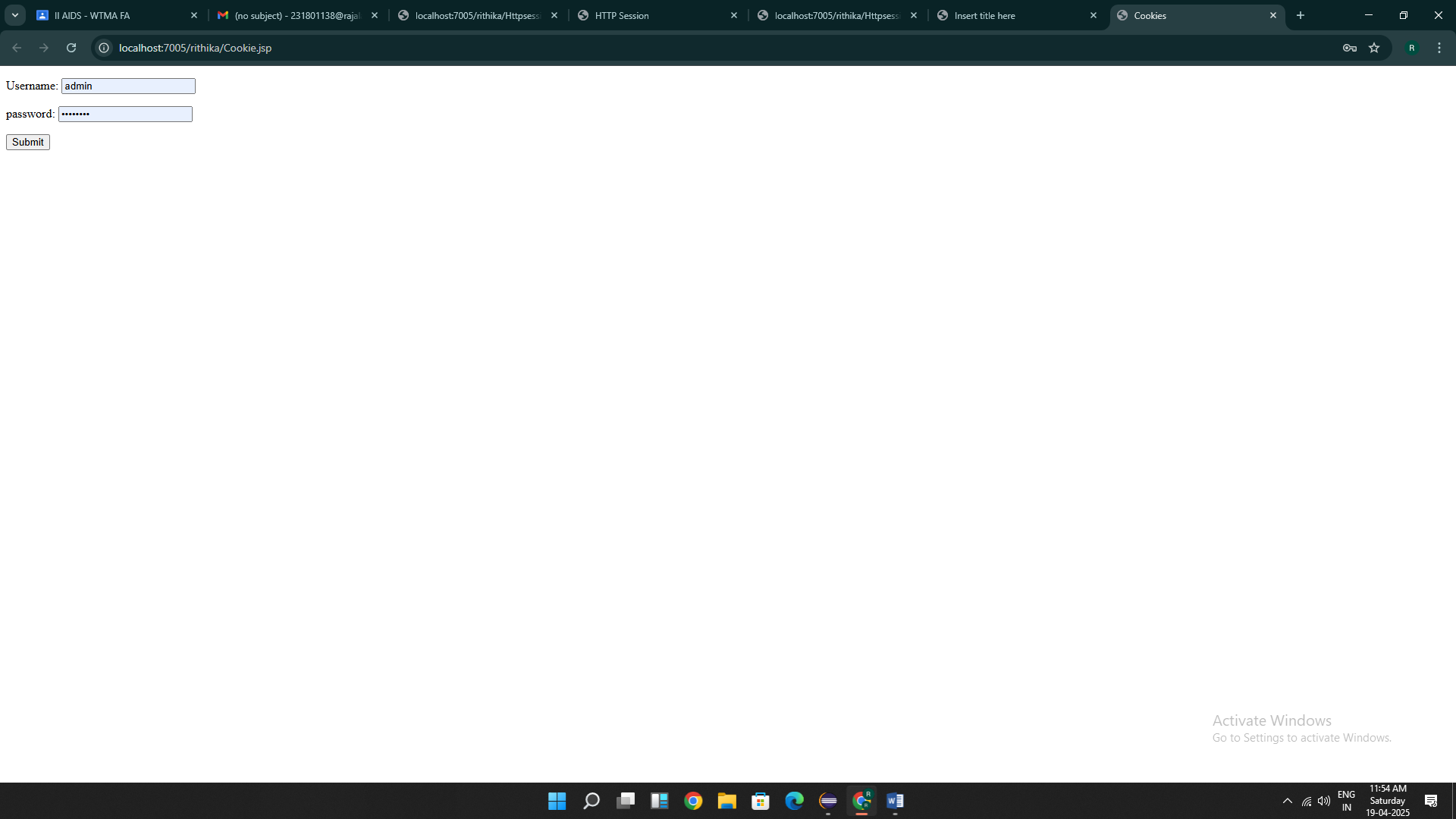
</body>

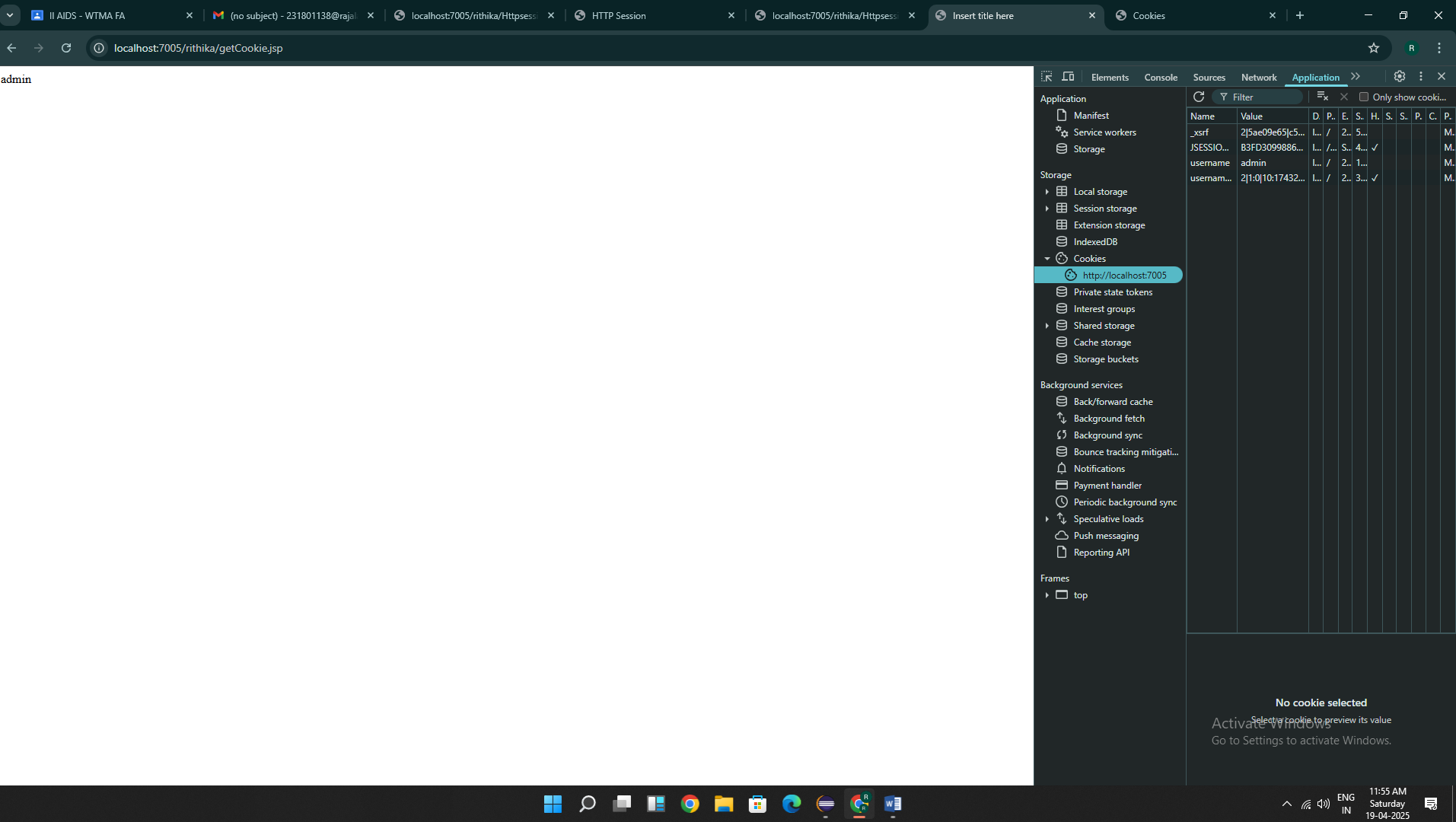
</html>





OUTPUT:





Result : Thus the programs are executed successfully

Exp:8

Code:

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

<%@ page contentType=*"text/html;charset=UTF-8"* language=*"java"* %>

<html>

<head>

<title>Library Management - Add Book</title>

<style>

**body** {

font-family: *Arial, sans-serif*;

background-color: *#f3f3f3*;

margin: *20px*;

padding: *20px*;

}

**h2** {

color: *#333*;

}

**form** {

background-color: *#fff*;

padding: *20px*;

border-radius: *8px*;

box-shadow: *0 2px 5px rgba(0,0,0,0.1)*;

width: *400px*;

}

**input**[type="text"]**,**

**input**[type="number"]**,**

**select** {

width: *100%*;

padding: *8px*;

margin: *6px 0 12px 0*;

border: *1px solid #ccc*;

border-radius: *4px*;

}

**input**[type="submit"] {

background-color: *#4CAF50*;

color: *white*;

border: *none*;

padding: *10px 16px*;

cursor: *pointer*;

border-radius: *4px*;

}

**input**[type="submit"]*:hover* {

background-color: *#45a049*;

}

**p** {

font-size: *16px*;

}

**a** {

color: *#007BFF*;

text-decoration: *none*;

}

**a***:hover* {

text-decoration: *underline*;

}

</style>

<script>

**function** validateForm() {

**const** bookId = document.getElementById("bookId").value.trim();

**const** title = document.getElementById("title").value.trim();

**const** author = document.getElementById("author").value.trim();

**const** year = document.getElementById("year").value.trim();

**const** category = document.getElementById("category").value;

**const** copies = document.getElementById("copies").value.trim();

**const** currentYear = **new** Date().getFullYear();

**if** (!/^[a-zA-Z0-9]+$/.test(bookId)) {

alert("Book ID must be alphanumeric.");

**return** **false**;

}

**if** (!/^[A-Za-z ]+$/.test(title)) {

alert("Title must contain only letters.");

**return** **false**;

}

**if** (!/^[A-Za-z ]+$/.test(author)) {

alert("Author name must contain only letters.");

**return** **false**;

}

**if** (!/^\d{4}$/.test(year) || parseInt(year) > currentYear) {

alert("Year must be a valid 4-digit year.");

**return** **false**;

}

**if** (category === "") {

alert("Please select a category.");

**return** **false**;

}

**if** (isNaN(copies) || parseInt(copies) <= 0) {

alert("Copies must be a positive number.");

**return** **false**;

}

**return** **true**;

}

</script>

</head>

<body>

<h2>Add Book to Library</h2>

<form method=*"post"* onsubmit="return validateForm()">

Book ID: <input type=*"text"* name=*"bookId"* id=*"bookId"*/><br/>

Title: <input type=*"text"* name=*"title"* id=*"title"*/><br/>

Author: <input type=*"text"* name=*"author"* id=*"author"*/><br/>

Year: <input type=*"text"* name=*"year"* id=*"year"*/><br/>

Category:

<select name=*"category"* id=*"category"*>

<option value=*""*>--Select--</option>

<option value=*"Science"*>Science</option>

<option value=*"Fiction"*>Fiction</option>

<option value=*"History"*>History</option>

</select><br/>

Copies: <input type=*"number"* name=*"copies"* id=*"copies"*/><br/><br/>

<input type=*"submit"* value=*"Add Book"*/>

<p>

<label>Book Details:-</label>

<a href=*"./display\_books.jsp"*>Click Here</a>

</p>

</form>

<%

// JSP Database logic

// TODO:Change password before run the code

// TODO: CREATE THE TABLE IN MYSQL

/\*\*

CREATE TABLE books (

book\_id VARCHAR(20) PRIMARY KEY,

title VARCHAR(100),

author VARCHAR(100),

year INT,

category VARCHAR(50),

copies INT

);

\*\*/

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

**if** (bookId != **null**) {

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

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

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

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

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

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

**try** {

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

Connection conn = DriverManager.getConnection(

"jdbc:mysql://localhost:3306/library\_management\_rithika", "root", "");

PreparedStatement ps = conn.prepareStatement(

"INSERT INTO books (book\_id, title, author, year, category, copies) VALUES (?, ?, ?, ?, ?, ?)");

ps.setString(1, bookId);

ps.setString(2, title);

ps.setString(3, author);

ps.setInt(4, Integer.parseInt(year));

ps.setString(5, category);

ps.setInt(6, Integer.parseInt(copies));

**int** result = ps.executeUpdate();

**if** (result > 0) {

%>

<script>alert("Book Added Successfully!!")</script>

<%

} **else** {

%>

<script>alert("Failed to add book!!")</script>

<%

out.println("<p style='color:red;'>Failed to add book.</p>");

}

ps.close();

conn.close();

} **catch** (Exception e) {

out.println("<p style='color:red;'>Error: " + e.getMessage() + "</p>");

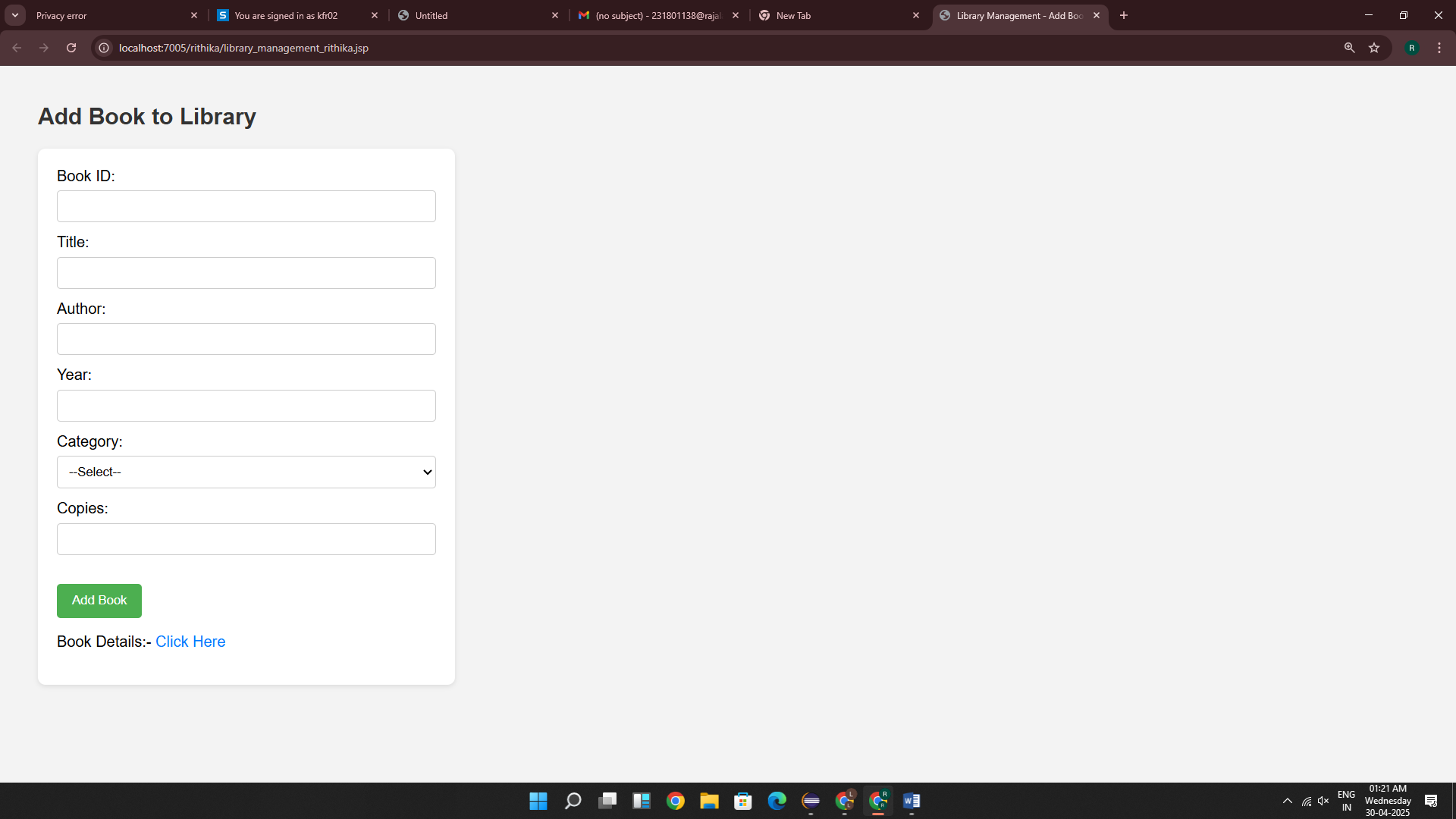
}

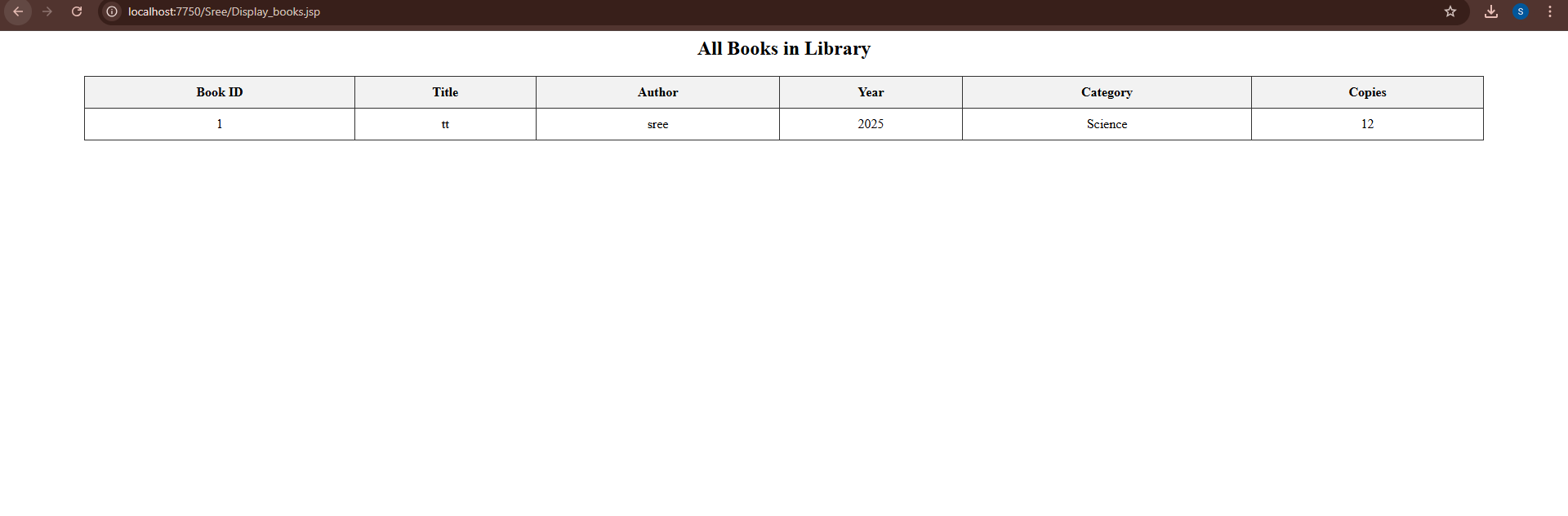
}

%>

</body>

</html>





Result : Thus the code is executed successfully.

EXP:9

**AIM:**

Develop an Android application using controls like Button, TextView, EditText for designing a calculator having basic functionality like Addition, Subtraction, multiplication, and Division.

**CODE:**

**MainActivity.kt**

package com.example.myapplication\_rithika

import android.os.Bundle import android.view.View import android.widget.Button import android.widget.EditText import android.widget.Toast

import androidx.appcompat.app.AppCompatActivity class MainActivity : AppCompatActivity() {

private lateinit var display: EditText private var currentInput: String = "" private var operand1: Double = 0.0 private var operand2: Double = 0.0 private var operator: String = ""

override fun onCreate(savedInstanceState: Bundle?) { super.onCreate(savedInstanceState) setContentView(R.layout.*activity\_main*)

display = findViewById(R.id.*display*)

// Number buttons setButtonClickListener(R.id.*button0*) setButtonClickListener(R.id.*button1*) setButtonClickListener(R.id.*button2*) setButtonClickListener(R.id.*button3*) setButtonClickListener(R.id.*button4*) setButtonClickListener(R.id.*button5*) setButtonClickListener(R.id.*button6*) setButtonClickListener(R.id.*button7*) setButtonClickListener(R.id.*button8*) setButtonClickListener(R.id.*button9*)

// Operator buttons setOperatorClickListener(R.id.*buttonAdd*, "+") setOperatorClickListener(R.id.*buttonSubtract*, "-") setOperatorClickListener(R.id.*buttonMultiply*, "\*") setOperatorClickListener(R.id.*buttonDivide*, "/")

// Clear button findViewById<Button>(R.id.*buttonClear*).setOnClickListener {

currentInput = "" operand1 = 0.0

operand2 = 0.0 operator = "" display.setText("")

}

// Equal button findViewById<Button>(R.id.*buttonEqual*).setOnClickListener {

if (operator.*isEmpty*()) return@setOnClickListener

try {

operand2 = currentInput.*toDouble*()

val result = when (operator) { "+" -> operand1 + operand2 "-" -> operand1 - operand2 "\*" -> operand1 \* operand2 "/" -> {

if (operand2 == 0.0) {

Toast.makeText(this@MainActivity, "Cannot divide by zero", Toast.*LENGTH\_SHORT*).show()

return@setOnClickListener

}

operand1 / operand2

}

else -> 0.0

}

display.setText(result.toString())

operand1 = result // Update operand1 for subsequent calculations operator = ""

currentInput = result.toString()

} catch (e: Exception) {

Toast.makeText(this@MainActivity, "Error: Invalid Input", Toast.*LENGTH\_SHORT*).show()

}

}

}

// Set up number button listeners

private fun setButtonClickListener(buttonId: Int) { val button: Button = findViewById(buttonId) button.setOnClickListener {

currentInput += button.*text*.toString() display.setText(currentInput)

}

}

// Set up operator button listeners

private fun setOperatorClickListener(buttonId: Int, op: String) { val button: Button = findViewById(buttonId)

button.setOnClickListener {

if (currentInput.*isNotEmpty*()) { operand1 = currentInput.*toDouble*() currentInput = ""

operator = op

}

}

}

// Optional: Add functionality to handle decimal point (if needed) private fun isDecimalPointValid(): Boolean {

return !currentInput.*contains*(".")

}

}

**activity\_main.xml**

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="<http://schemas.android.com/apk/res/android>" xmlns:tools="<http://schemas.android.com/tools>" android:id="@+id/calculatorLayout" android:layout\_width="match\_parent" android:layout\_height="match\_parent"

android:orientation="vertical" android:padding="16dp" tools:context=".MainActivity">

<EditText android:id="@+id/display"

android:layout\_width="match\_parent" android:layout\_height="80dp" android:background="#000000" android:textColor="#FFFFFF" android:textSize="28sp" android:gravity="end|center\_vertical" android:inputType="none" android:focusable="false" android:clickable="false" android:padding="12dp" android:layout\_marginBottom="12dp" />

<!-- Row 1: 7 8 9 / -->

<LinearLayout android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:orientation="horizontal">

<Button android:id="@+id/button7" style="@style/CalcButton" android:text="7" />

<Button android:id="@+id/button8" style="@style/CalcButton" android:text="8" />

<Button android:id="@+id/button9" style="@style/CalcButton" android:text="9" />

<Button android:id="@+id/buttonDivide" style="@style/CalcButton" android:text="/" />

</LinearLayout>

<!-- Row 2: 4 5 6 \* -->

<LinearLayout

android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:orientation="horizontal">

<Button android:id="@+id/button4" style="@style/CalcButton" android:text="4" />

<Button android:id="@+id/button5" style="@style/CalcButton" android:text="5" />

<Button android:id="@+id/button6" style="@style/CalcButton" android:text="6" />

<Button android:id="@+id/buttonMultiply" style="@style/CalcButton" android:text="\*" />

</LinearLayout>

<!-- Row 3: 1 2 3 - -->

<LinearLayout android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:orientation="horizontal">

<Button android:id="@+id/button1" style="@style/CalcButton" android:text="1" />

<Button android:id="@+id/button2" style="@style/CalcButton" android:text="2" />

<Button android:id="@+id/button3" style="@style/CalcButton" android:text="3" />

<Button android:id="@+id/buttonSubtract" style="@style/CalcButton" android:text="-" />

</LinearLayout>

<!-- Row 4: 0 C = + -->

<LinearLayout android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:orientation="horizontal">

<Button android:id="@+id/button0" style="@style/CalcButton" android:text="0" />

<Button android:id="@+id/buttonClear" style="@style/CalcButton" android:text="C" />

<Button android:id="@+id/buttonEqual" style="@style/CalcButton" android:text="=" />

<Button android:id="@+id/buttonAdd" style="@style/CalcButton" android:text="+" />

</LinearLayout>

</LinearLayout>

**styles.xml**

<resources>

<style name="CalcButton">

<item name="android:layout\_width">0dp</item>

<item name="android:layout\_height">wrap\_content</item>

<item name="android:layout\_weight">1</item>

<item name="android:padding">16dp</item>

<item name="android:textSize">24sp</item>

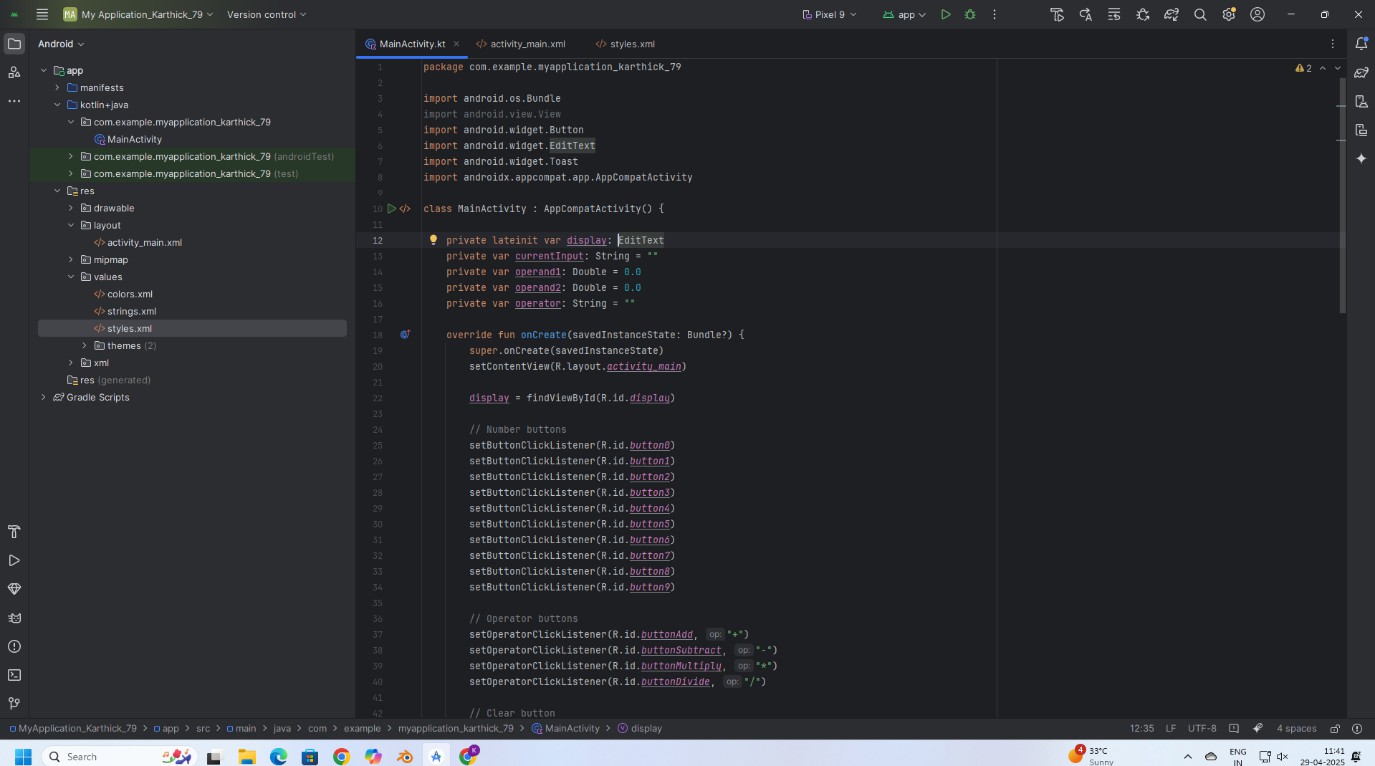
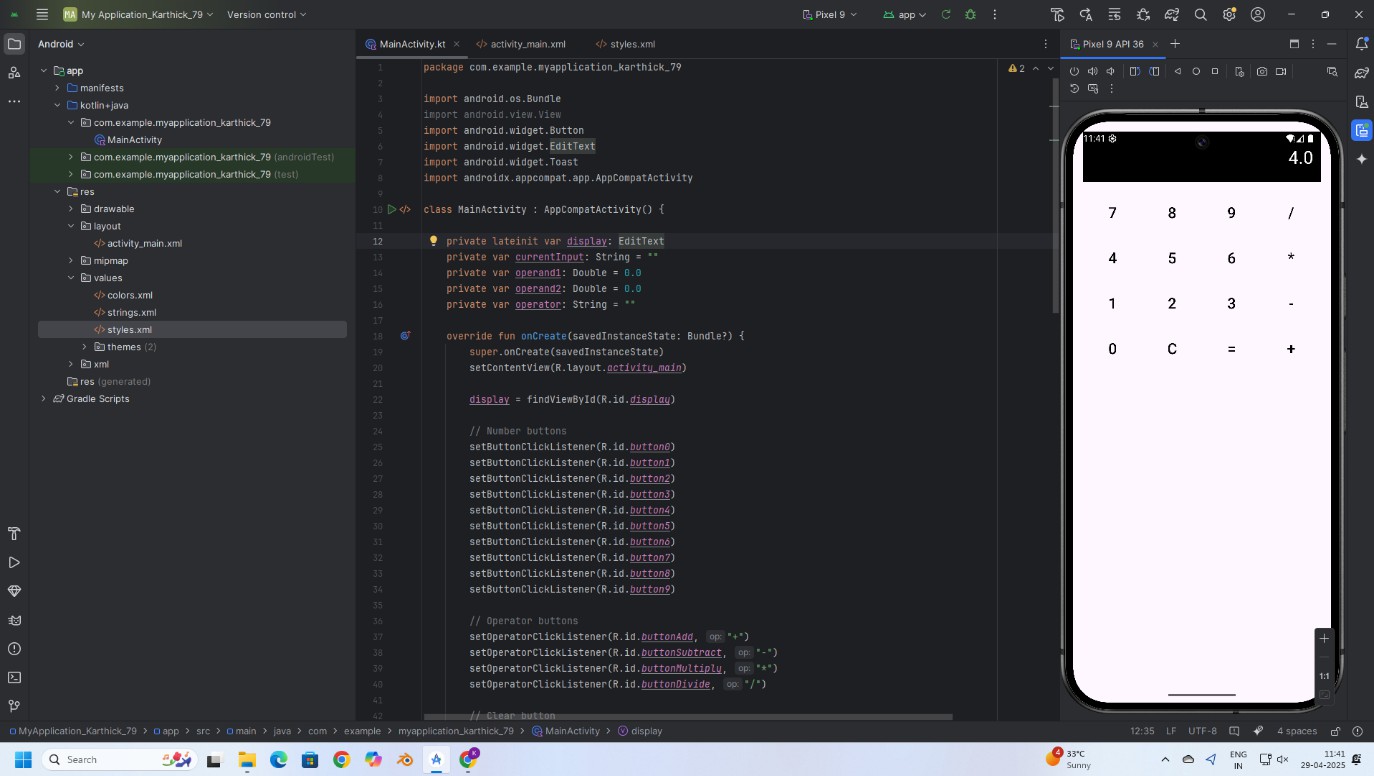
<item name="android:background">?attr/selectableItemBackground</item>

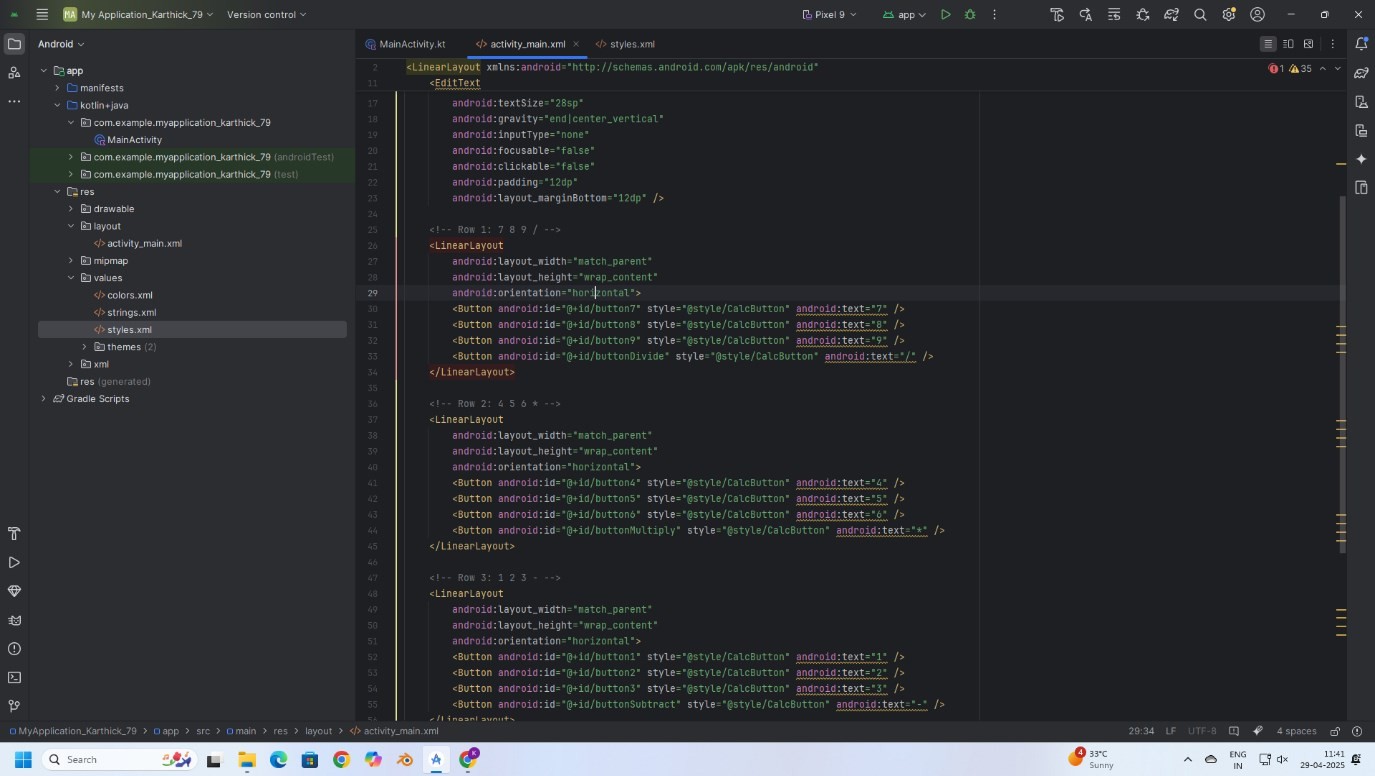
<item name="android:textColor">#000000</item>

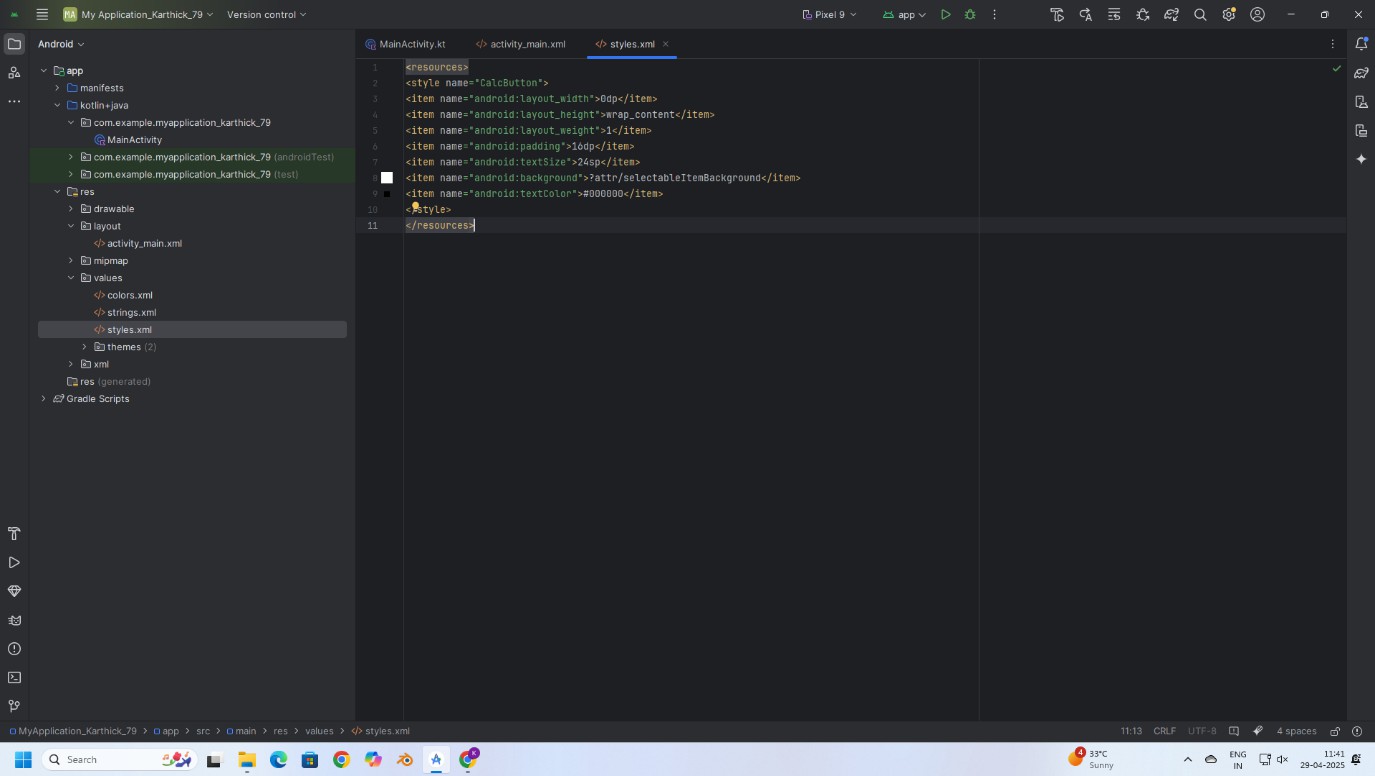
</style>

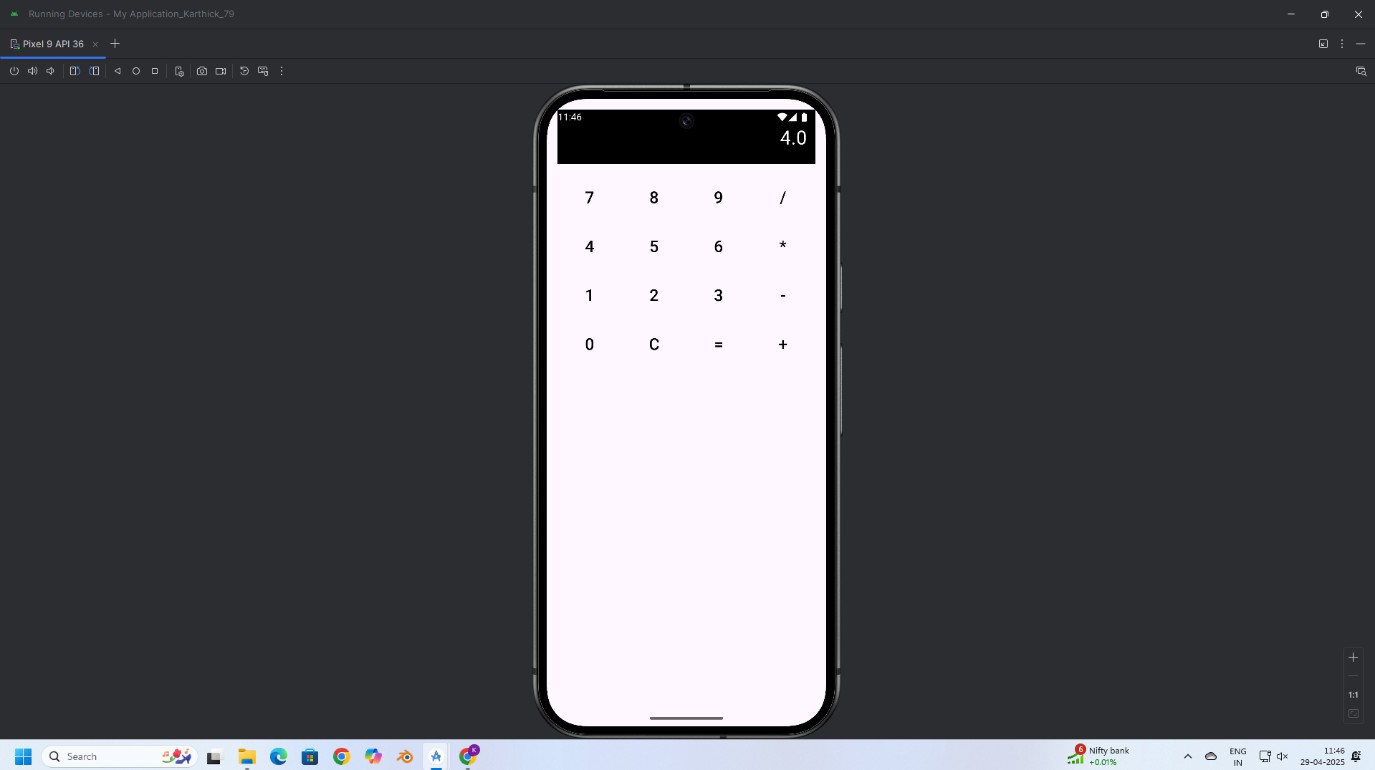
</resources>

**OUTPUT:**









**RESULT:**

Thus, a basic calculator application was successfully developed using Android controls like Button, TextView, and EditText to perform addition, subtraction, multiplication, and division operations.

EXP:10

**AIM:**

Develop an application to change the font and color of the text and display toast message when the user presses the button.

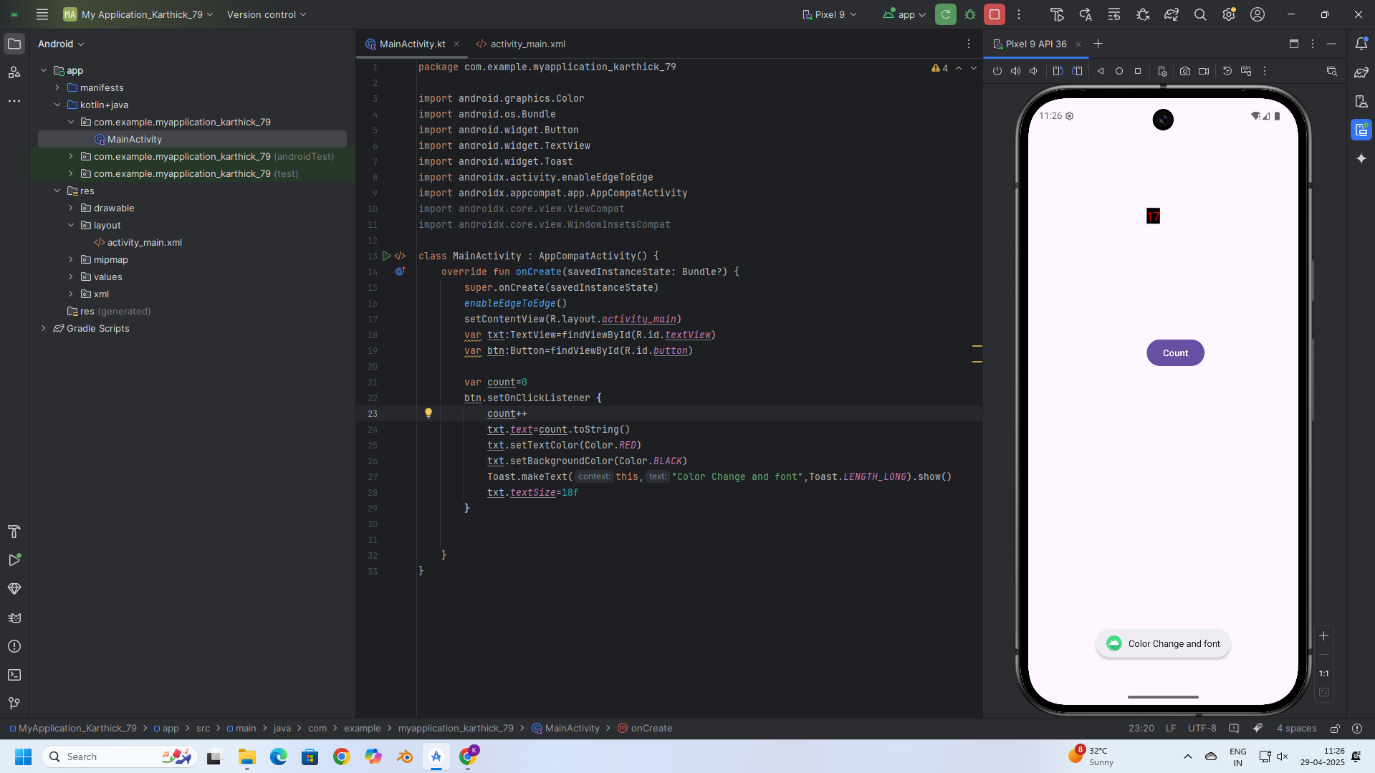
**CODE:**

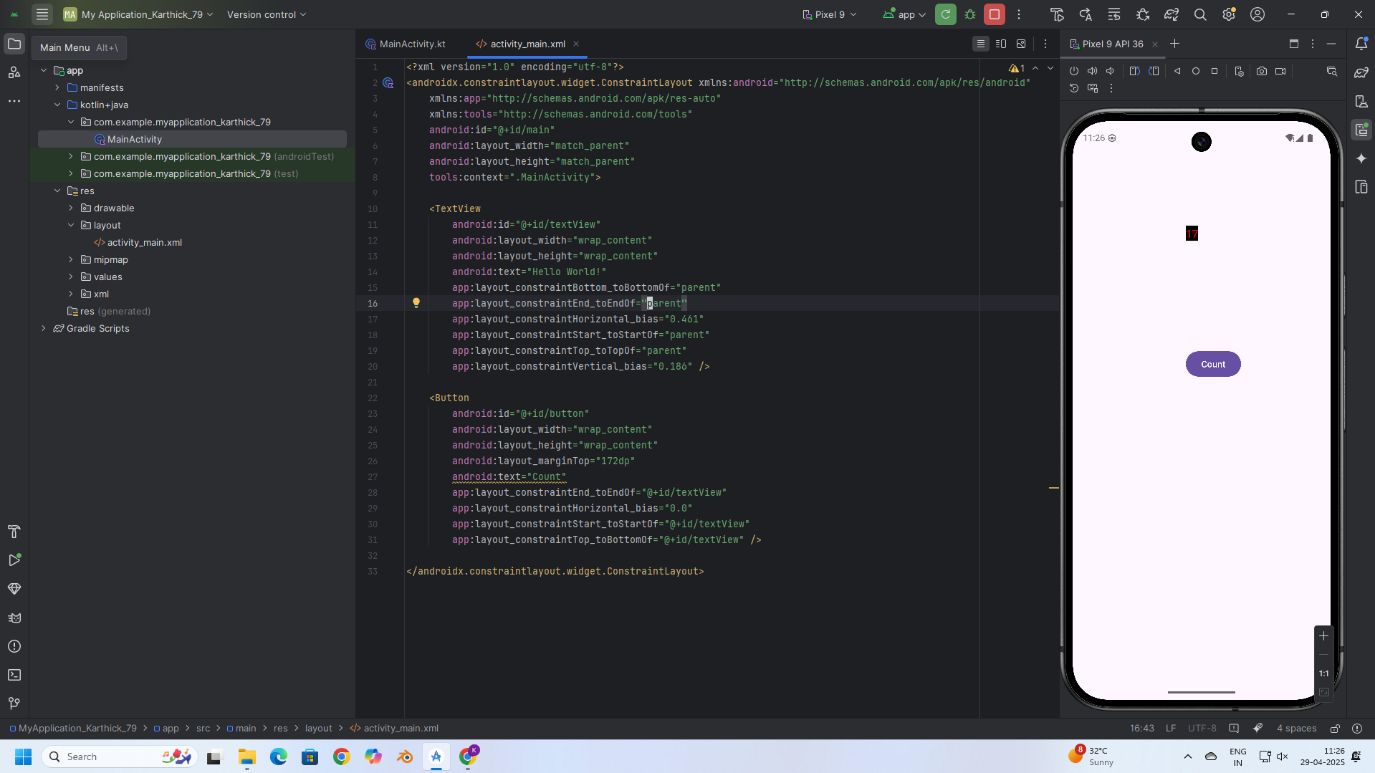
**MainActivity.kt**

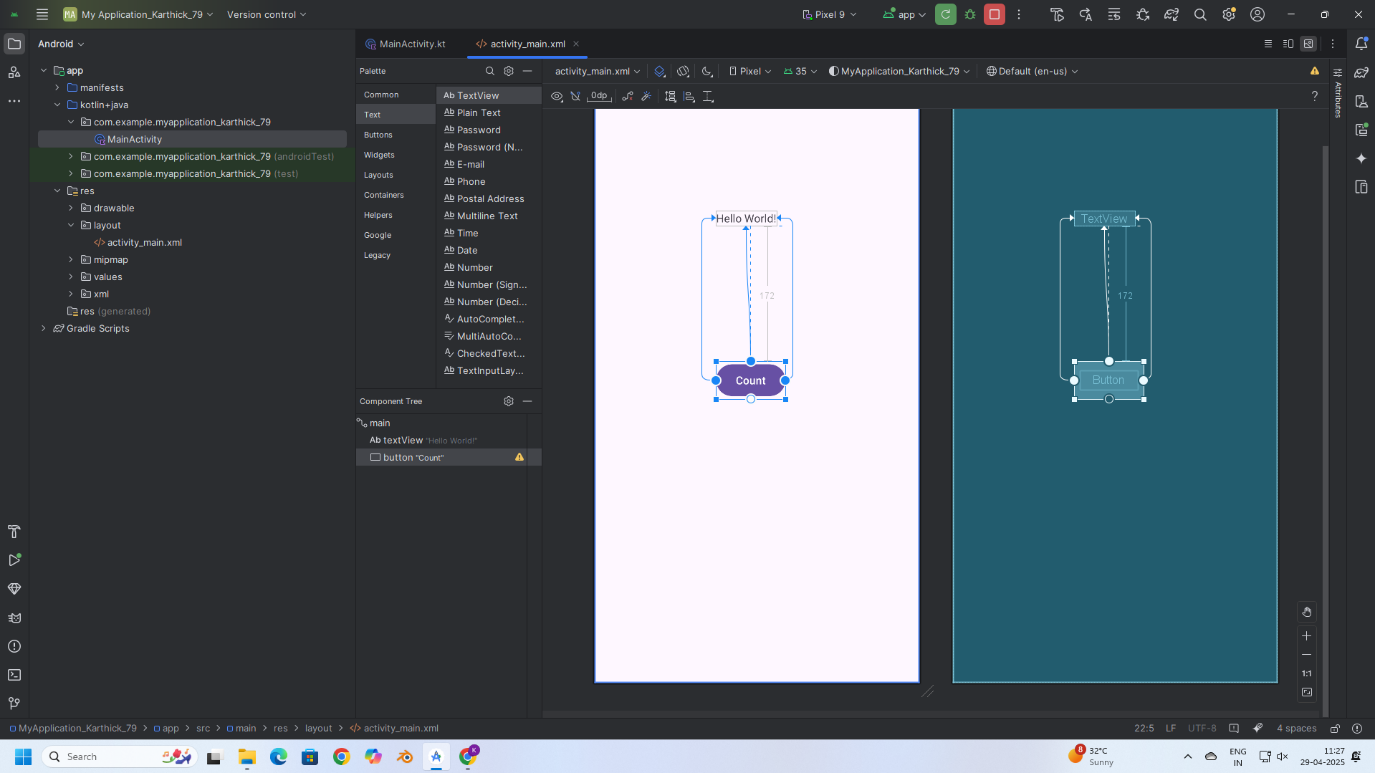
package com.example.myapplication\_rithika\_  
  
import android.graphics.Color  
import android.os.Bundle  
import android.widget.Button  
import android.widget.TextView  
import android.widget.Toast  
import androidx.activity.enableEdgeToEdge  
import androidx.appcompat.app.AppCompatActivity  
import androidx.core.view.ViewCompat  
import androidx.core.view.WindowInsetsCompat  
  
class MainActivity : AppCompatActivity() {  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 enableEdgeToEdge()  
 setContentView(R.layout.activity\_main)  
 var txt:TextView=findViewById(R.id.textView)  
 var btn:Button=findViewById(R.id.button)  
  
 var count=0  
 btn.setOnClickListener {  
 count++  
 txt.text=count.toString()  
 txt.setTextColor(Color.RED)  
 txt.setBackgroundColor(Color.BLACK)  
 Toast.makeText(this,"Color Change and font",Toast.LENGTH\_LONG).show()  
 txt.textSize=18f  
 }  
  
  
 }  
}

**activity\_main.xml**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="<http://schemas.android.com/apk/res/android>"  
 xmlns:app="<http://schemas.android.com/apk/res-auto>"  
 xmlns:tools="<http://schemas.android.com/tools>"  
 android:id="@+id/main"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <TextView  
 android:id="@+id/textView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Hello World!"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.461"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.186" />  
  
 <Button  
 android:id="@+id/button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="172dp"  
 android:text="Count"  
 app:layout\_constraintEnd\_toEndOf="@+id/textView"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="@+id/textView"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

**OUTPUT:**

****

****

**RESULT:**

Thus, text customization and user interaction were successfully implemented by changing font and color, and displaying a Toast message on button click in the application.