# ALA PRACTICAL 1

## AIM : Write a Client-server application using UDP Protocol in which client sends a message to server. Server will convert that message into upper case and returns the result back to client.

### server.java

package alapractical1;

import java.io.\*;

import java.net.\*;

import java.util.\*;

public class server

{ public static void main(String [] args){

try

{ DatagramSocket ds=new DatagramSocket(3333);

System.out.println("Server Ready.");

byte buf[]=new byte[1000];

DatagramPacket dp=new DatagramPacket(buf,buf.length);

ds.receive(dp);

String str=new String(dp.getData());

System.out.println("Message from client: "+str);

String s=str.toUpperCase();

byte msg[]=s.getBytes();

DatagramPacket dp1=new DatagramPacket(msg,msg.length,dp.getAddress(),dp.getPort());

ds.send(dp1);

ds.close();

}

catch(Exception e)

{ System.out.println(e); }

}

}

### client.java

package alapractical1;

import java.io.\*;

import java.net.\*;

import java.util.\*;

public class client

{ public static void main(String [] args)

{ DatagramSocket ds=null;

Scanner sc=new Scanner(System.in);

try

{ ds=new DatagramSocket();

System.out.println("Socket created.");

System.out.println("Enter a String: ");

String s="";

s=sc.nextLine();

byte[] m=s.getBytes();

InetAddress ip=InetAddress.getLocalHost();

int serverport=3333;

DatagramPacket request=new DatagramPacket(m,m.length,ip,3333);

ds.send(request);

byte[] buffer=new byte[1000];

DatagramPacket reply=new DatagramPacket(buffer,buffer.length);

ds.receive(reply);

System.out.println("Reply from server"+":"+new String(reply.getData()));

}

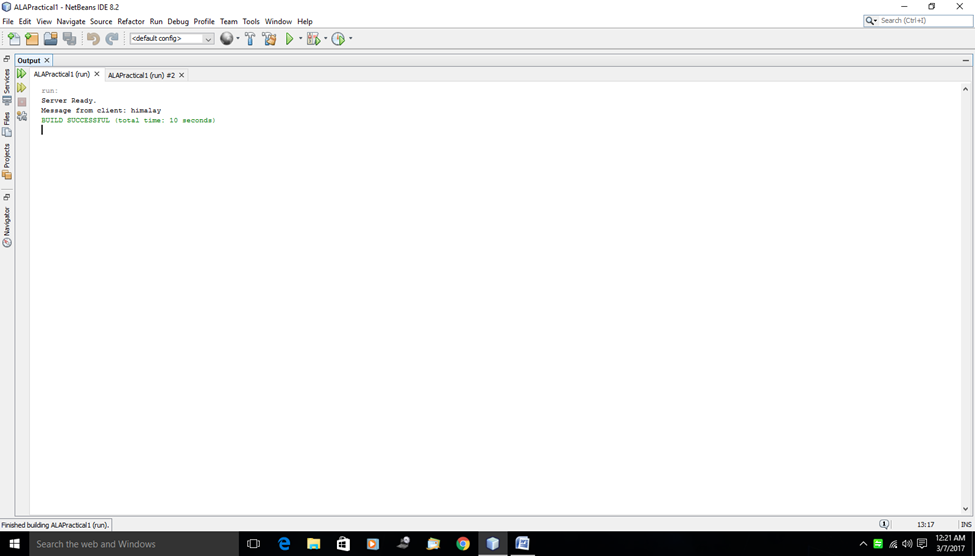
catch(Exception e)

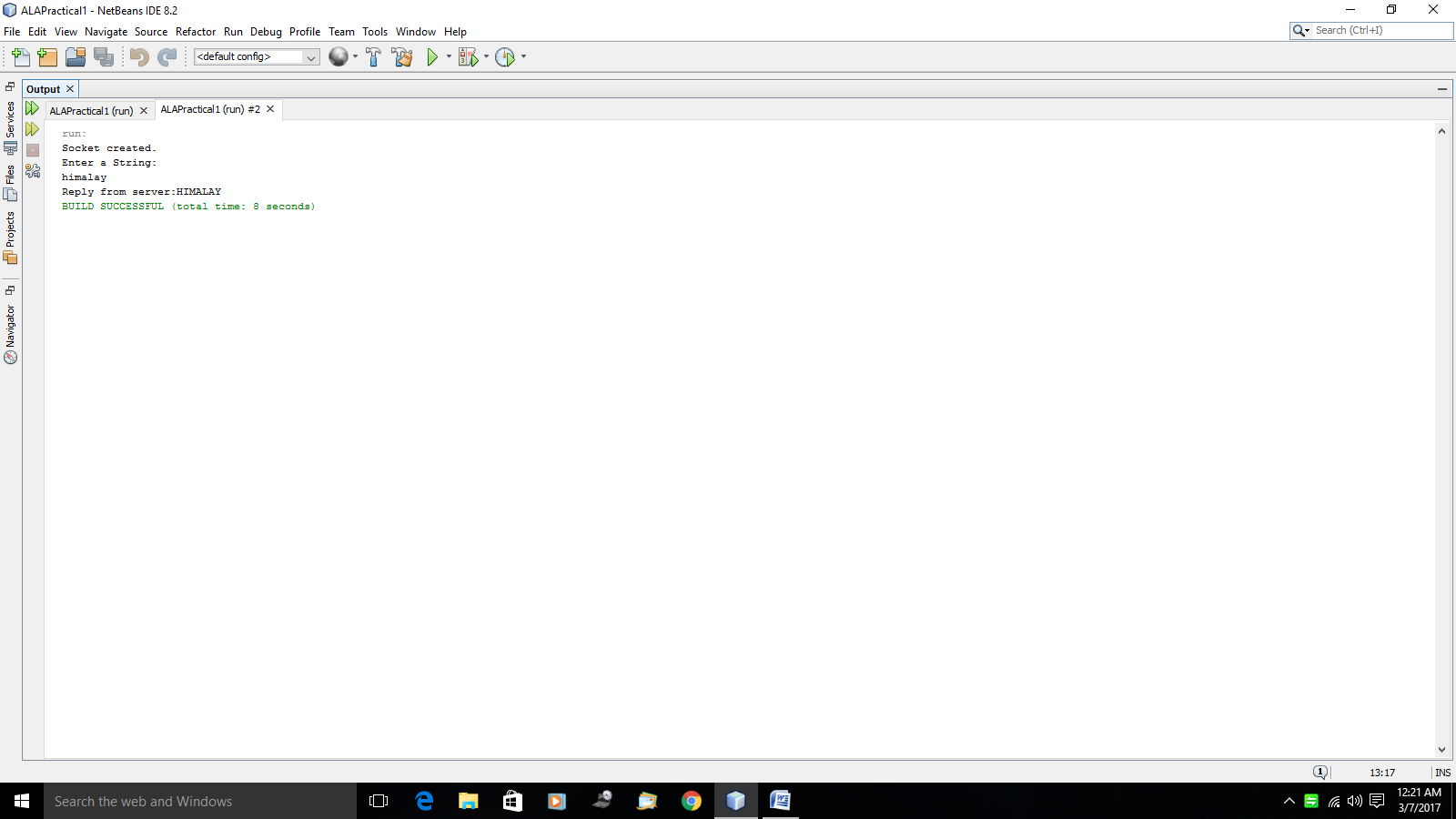
{ System.out.println(e); }

}

}

### ALA Practical 1 Output



****

# ALA PRACTICAL 2A

## AIM : Learn how to process the database metadata.

package ALAPractical2;

import java.sql.\*;

public class alapractical2a

{ public static void main(String[] args)

{ try

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

String username = "root";

String password = "himalaypatel";

Connection c = DriverManager.getConnection("jdbc:mysql://localhost:3306/himalay",username,password);

DatabaseMetaData dmd = c.getMetaData();

System.out.println("The Database Product Name : "+dmd.getDatabaseProductName());

System.out.println("The Username : "+dmd.getUserName());

System.out.println("The Catalogs are : ");

ResultSet rs = dmd.getCatalogs();

while(rs.next())

{ System.out.println(rs.getString(1)); }

String catalog = null;

String schema = null;

String table = null;

String[] types = {"TABLE"};

ResultSet rs1 = dmd.getTables(catalog, schema, table, types);

System.out.println("Table names : ");

while(rs1.next())

{ System.out.println(rs1.getString(3)); }

}

catch (ClassNotFoundException e)

{ e.printStackTrace(); }

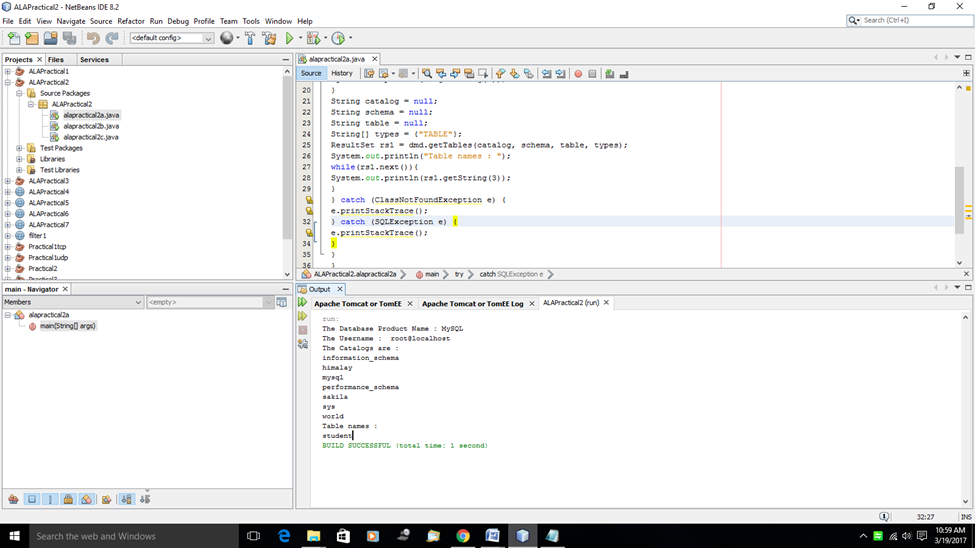
catch (SQLException e)

{ e.printStackTrace(); }

}

}

### ALA PRACTICAL 2A OUTPUT



# ALA PRACTICAL 2B

## AIM : Learn how to get metadata of a query parameter.

package ALAPractical2;

import java.sql.\*;

public class alapractical2b {

public static void main(String[] args)

{ try

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

String username = "root";

String password = "himalaypatel";

Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/himalay",username,password);

ParameterMetaData pmd;

String sql = "select \* from Student where Semester=? ";

PreparedStatement ps = conn.prepareStatement(sql);

pmd = ps.getParameterMetaData();

int count = pmd.getParameterCount();

System.out.println("The Parameter Count is : "+count);

}

catch (ClassNotFoundException e)

{ e.printStackTrace(); }

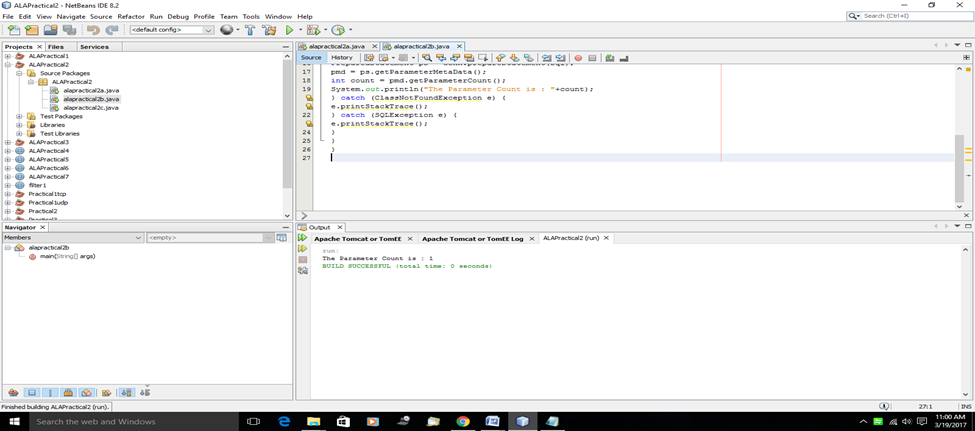
catch (SQLException e)

{ e.printStackTrace(); }

}

}

### ALA PRACTICAL 2B OUTPUT



# ALA PRACTICAL 2C

## AIM : Learn how to get metadata from the results of a query.

package ALAPractical2;

import java.sql.\*;

public class alapractical2c

{ public static void main(String[] args)

{ try

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

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/himalay","root","himalaypatel");

String query = "select \* from Student";

Statement s = con.createStatement();

Query(query);

java.sql.ResultSetMetaData rmd = rs.getMetaData();

int noCol = rmd.getColumnCount();

Number of columns is : "+noCol);

String ColName[] = new String[noCol];

for(int i=0;i<noCol;i++)

{ ColName[i] = rmd.getColumnName(i+1); }

String ColType[] = new String[noCol];

for(int i=0;i<noCol;i++)

{ ColType[i] = rmd.getColumnTypeName(i+1); }

for(int i=0; i<noCol;i++)

{ System.out.print(ColName[i]+" | "); }

System.out.println("\n");

for(int i=0; i<noCol;i++)

{ System.out.print(ColType[i]+" | "); }

}

catch (ClassNotFoundException e)

{ e.printStackTrace(); }

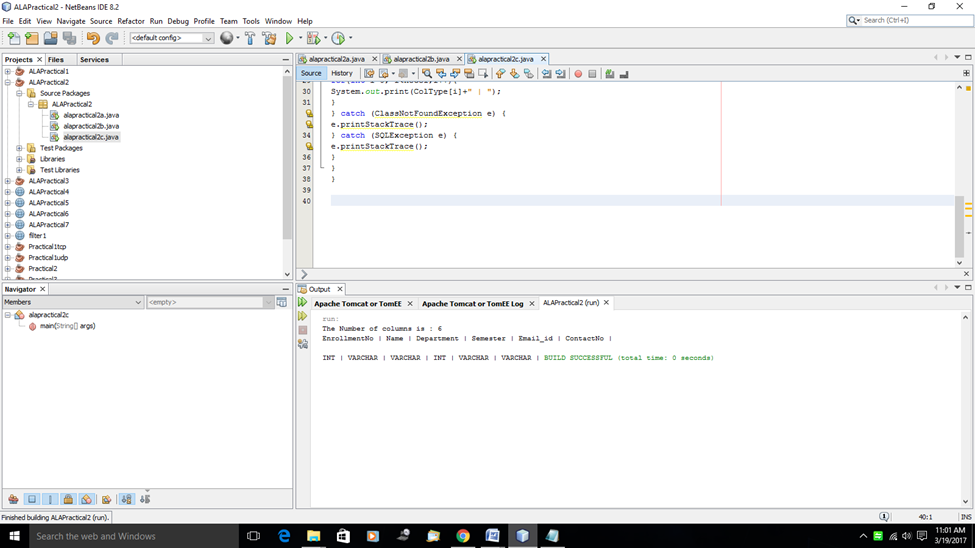
catch (SQLException e)

{ e.printStackTrace(); }

}

}

### ALA PRACTICAL 2C OUTPUT



# ALA PRACTICAL 3

## AIM :Write a servlet that shows all request headers and response headers.

### index.html

<html>

<head>

<title>Request and Response Headers</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

</head>

<body>

<a href="Servlet1">Get Headers</a>

</body>

</html>

### Servlet1.java (for displaying Request Headers)

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

import java.util.\*;

public class Servlet1 extends HttpServlet

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

{ doPost(request,response); }

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

{ response.setContentType("text/html");

PrintWriter out=response.getWriter();

try

{ String title = "HTTP Request Headers";

String docType = "<!doctype html public \"-//w3c//dtd html 4.0 " +

"transitional//en\">\n";

out.println(docType + "<html>\n" + "<head><title>" + title + "</title></head>\n"+ "<body bgcolor=\"orange\">\n" + "<h1 align=\"center\">" + title + "</h1>\n" + "<table width=\"100%\" border=\"1\" align=\"center\">\n" + "<tr bgcolor=\"yellow\">\n" + "<th>Header Name</th><th>Header Value(s)</th>\n"+ "</tr>\n");

}

catch(Exception e)

{ e.printStackTrace(); }

Enumeration headerNames = request.getHeaderNames();

while(headerNames.hasMoreElements())

{ String paramName = (String)headerNames.nextElement();

out.print("<tr><td>" + paramName + "</td>\n");

String paramValue = request.getHeader(paramName);

amValue + "</td></tr>\n");

}

response.setContentType("text/html");

out.println("</table>");

out.println("<a href="+"Servlet3"+">Get Response Headers</a>");

out.println("\n</body></html>");

}

}

### Servlet2.java (for displaying Response Headers)

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

import java.util.\*;

public class Servlet2 extends HttpServlet

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

{ doPost(request,response); }

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

{ response.setContentType("text/html");

PrintWriter out=response.getWriter();

try

{ out = response.getWriter();

String title = "HTTP Response Headers";

String docType = "<!doctype html public \"-//w3c//dtd html 4.0 " + "transitional//en\">\n";

out.println(docType + "<html>\n" + "<head><title>" + title + "</title></head>\n"+ "<body bgcolor=\"green\">\n" + "<h1 align=\"center\">" + title + "</h1>\n" + "<table width=\"100%\" border=\"1\" align=\"center\">\n" + "<tr bgcolor=\"purple\">\n" + "<th>Header Name</th><th>Header Value(s)</th>\n"+ "</tr>\n");

}

catch(Exception e)

{ e.printStackTrace(); }

Collection<String> respHeader= response.getHeaderNames();

<String> headerNames = respHeader.iterator();

while(headerNames.hasNext())

{ String paramName = headerNames.next();

out.print("<tr><td>" + paramName + "</td>\n");

String paramValue = response.getHeader(paramName);

out.println("<td> " + paramValue + "</td></tr>\n");

}

out.println("</table>");

out.println("<a href="+"Servlet1"+">Get Request Headers</a>");

out.println("\n</body></html>");

}

}

### Servlet3.java (for adding Response headers)

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

import java.util.\*;

public class Servlet3 extends HttpServlet

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

{ doPost(request,response); }

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

{ response.setHeader("Himalay", "Patel");

response.setHeader("BrowserUsed", "Chrome");

RequestDispatcher dispatcher = getServletContext().getNamedDispatcher("Servlet2");

dispatcher.forward(request, response);

}

}

### web.xml

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

<web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_3\_1.xsd">

<servlet>

<servlet-name>Servlet1</servlet-name>

<servlet-class>Servlet1</servlet-class>

</servlet>

<servlet>

<servlet-name>Servlet2</servlet-name>

<servlet-class>Servlet2</servlet-class>

</servlet>

<servlet>

<servlet-name>Servlet3</servlet-name>

<servlet-class>Servlet3</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>Servlet1</servlet-name>

<url-pattern>/Servlet1</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>Servlet2</servlet-name>

<url-pattern>/Servlet2</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>Servlet3</servlet-name>

<url-pattern>/Servlet3</url-pattern>

</servlet-mapping>

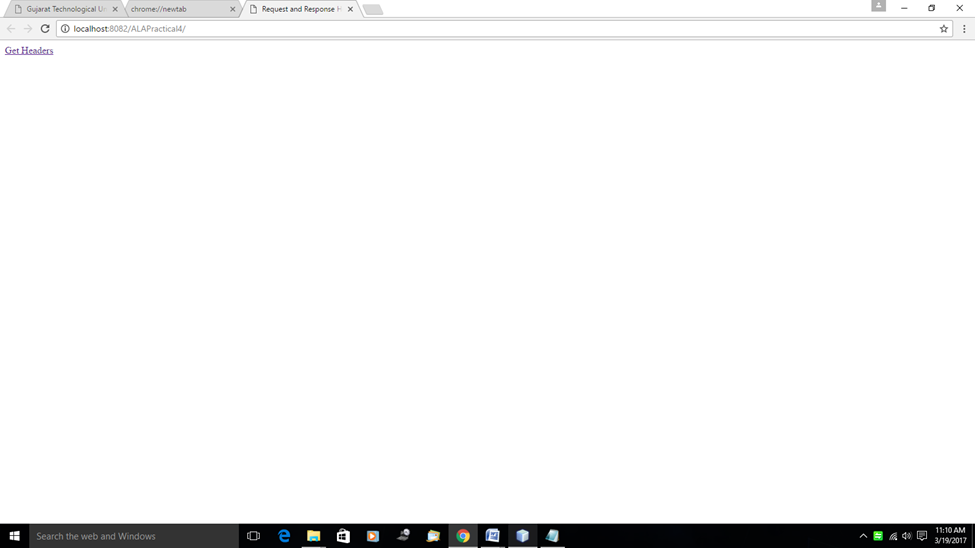
<welcome-file-list>

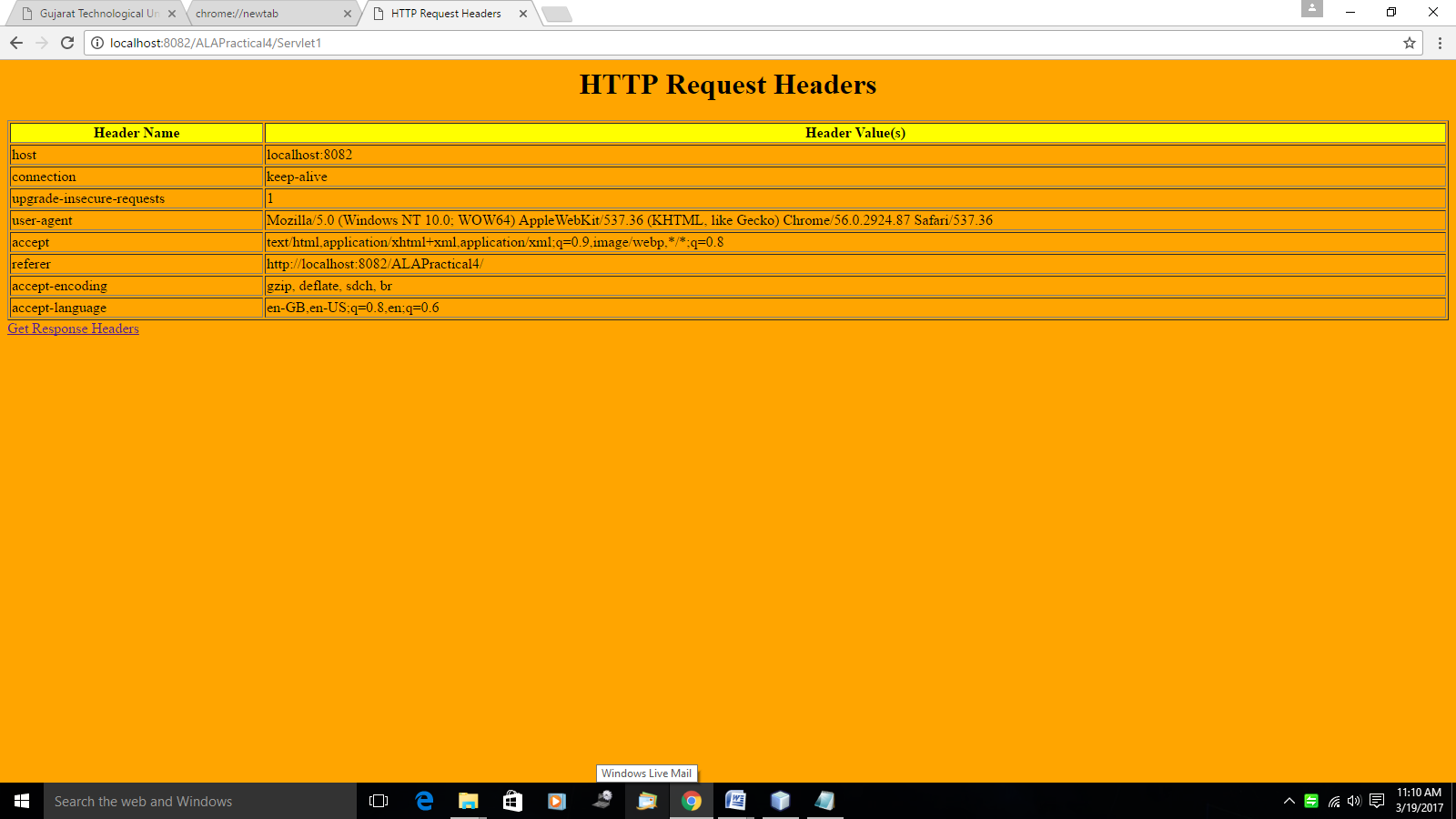
<welcome-file>index.html</welcome-file>

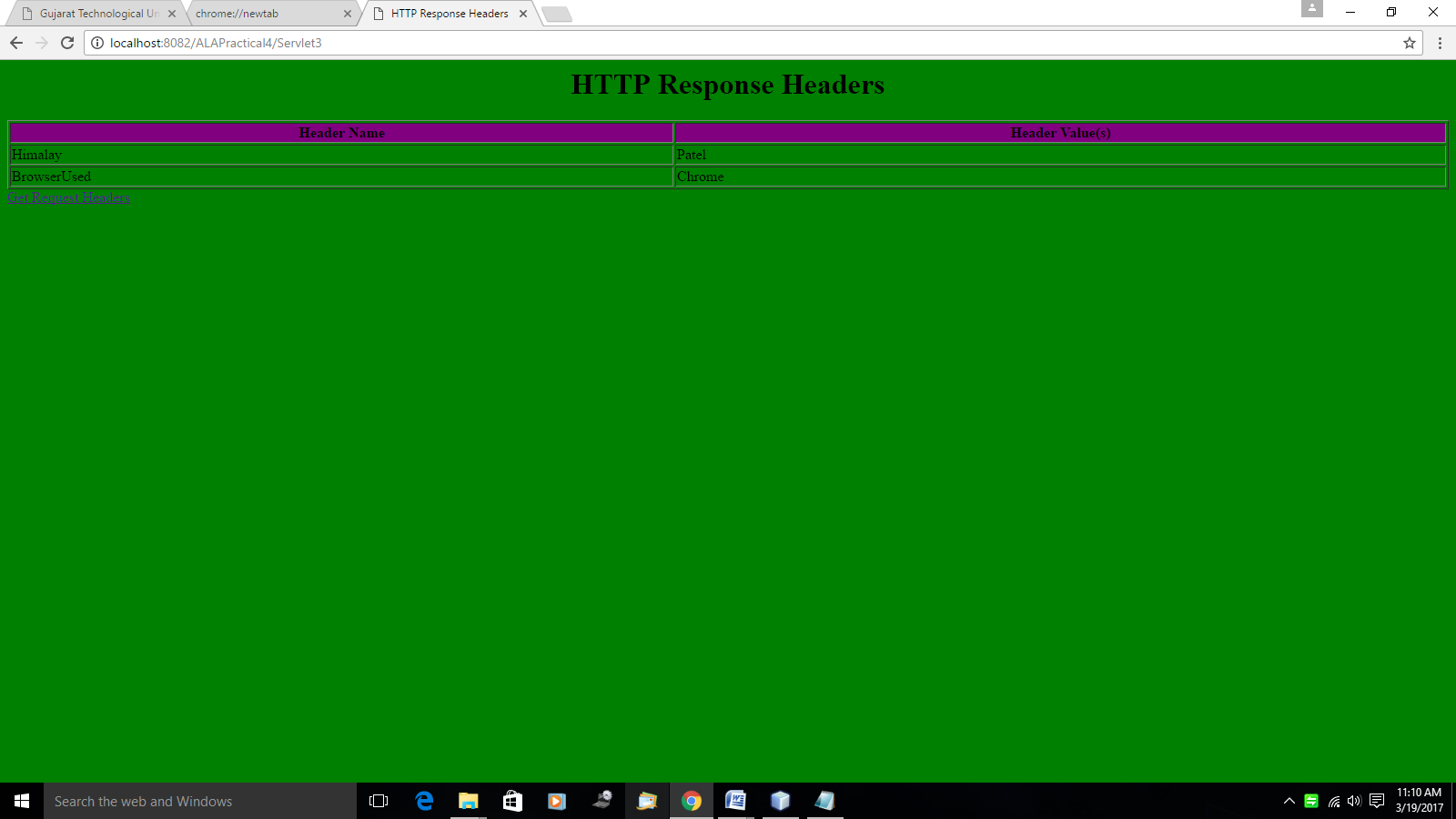
</welcome-file-list>

</web-app>

### ALA PRACTICAL 3 OUTPUT



****

****

# ALA PRACTICAL 4

## AIM : Write a servlet that instructs the browser to reconnect every five seconds. Display the time (print new java.util.Date()) on each connection. (response header)

### index.html

<html>

<head>

<title>Time</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

</head>

<body>

<a href="Servlet1">Time</a>

</body>

</html>

### Servlet1.java (for getting and displaying time)

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

import java.util.\*;

public class Servlet1 extends HttpServlet {

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

doPost(request,response);

}

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setIntHeader("Refresh", 5);

response.setContentType("text/html");

Calendar calendar = new GregorianCalendar();

String am\_pm;

int hour = calendar.get(Calendar.HOUR);

int minute = calendar.get(Calendar.MINUTE);

int second = calendar.get(Calendar.SECOND);

if(calendar.get(Calendar.AM\_PM) == 0)

am\_pm = "AM";

else

am\_pm = "PM";

String CT = hour+":"+ minute +":"+ second +" "+ am\_pm;

PrintWriter out = response.getWriter();

String title = "Auto Refresh Webpage";

String docType = "<!doctype html public \"-//w3c//dtd html 4.0 " +

"transitional//en\">\n";

out.println(docType +

"<html>\n" +

"<head><title>" + title + "</title></head>\n"+

"<body bgcolor=\"orange\">\n" +

"<h1 align=\"center\">" + title + "</h1>\n" +

"<p>Current Time is: " + CT + "</p>\n");

}

}

### web.xml

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

<web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_3\_1.xsd">

<servlet>

<servlet-name>Servlet1</servlet-name>

<servlet-class>Servlet1</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>Servlet1</servlet-name>

<url-pattern>/Servlet1</url-pattern>

</servlet-mapping>

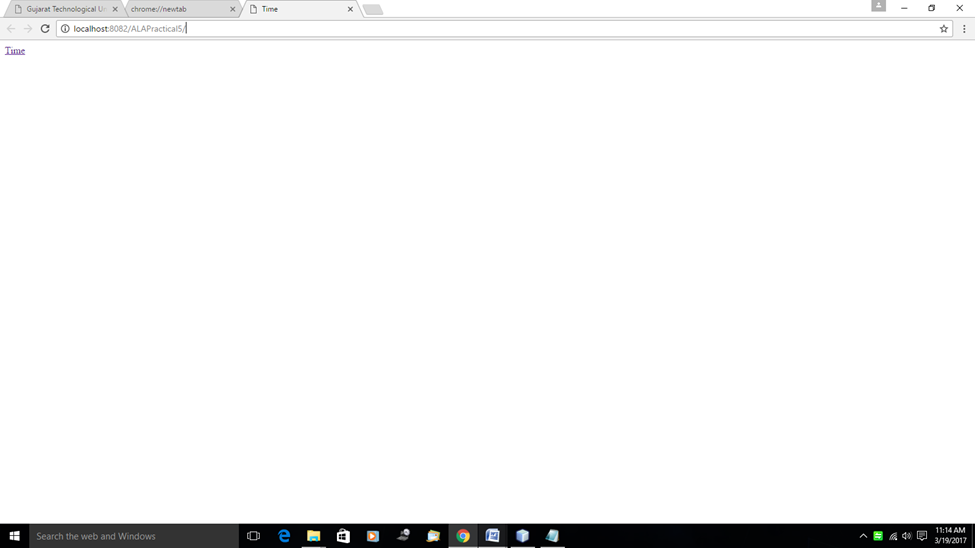
<welcome-file-list>

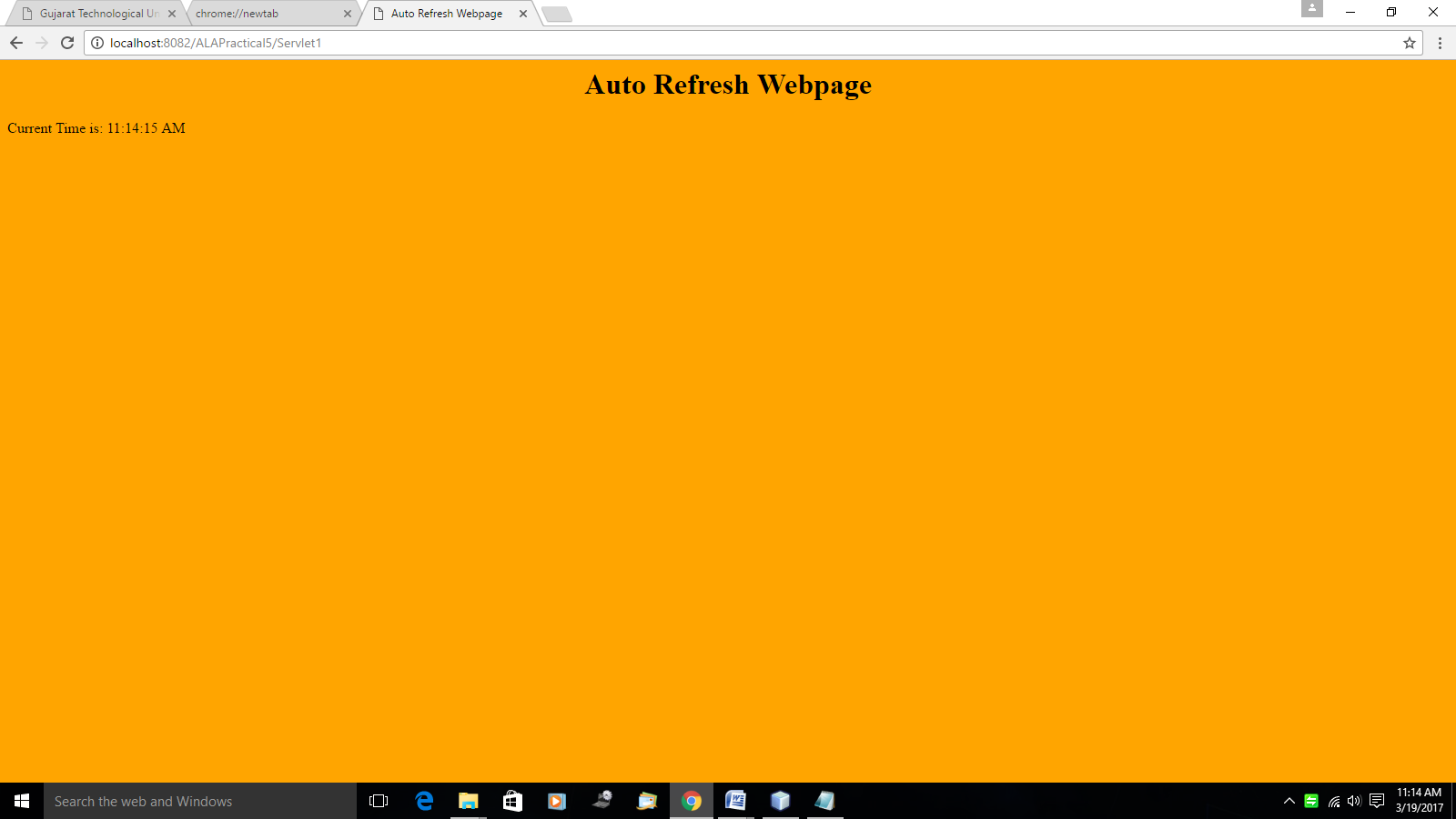
<welcome-file>index.html</welcome-file>

</welcome-file-list>

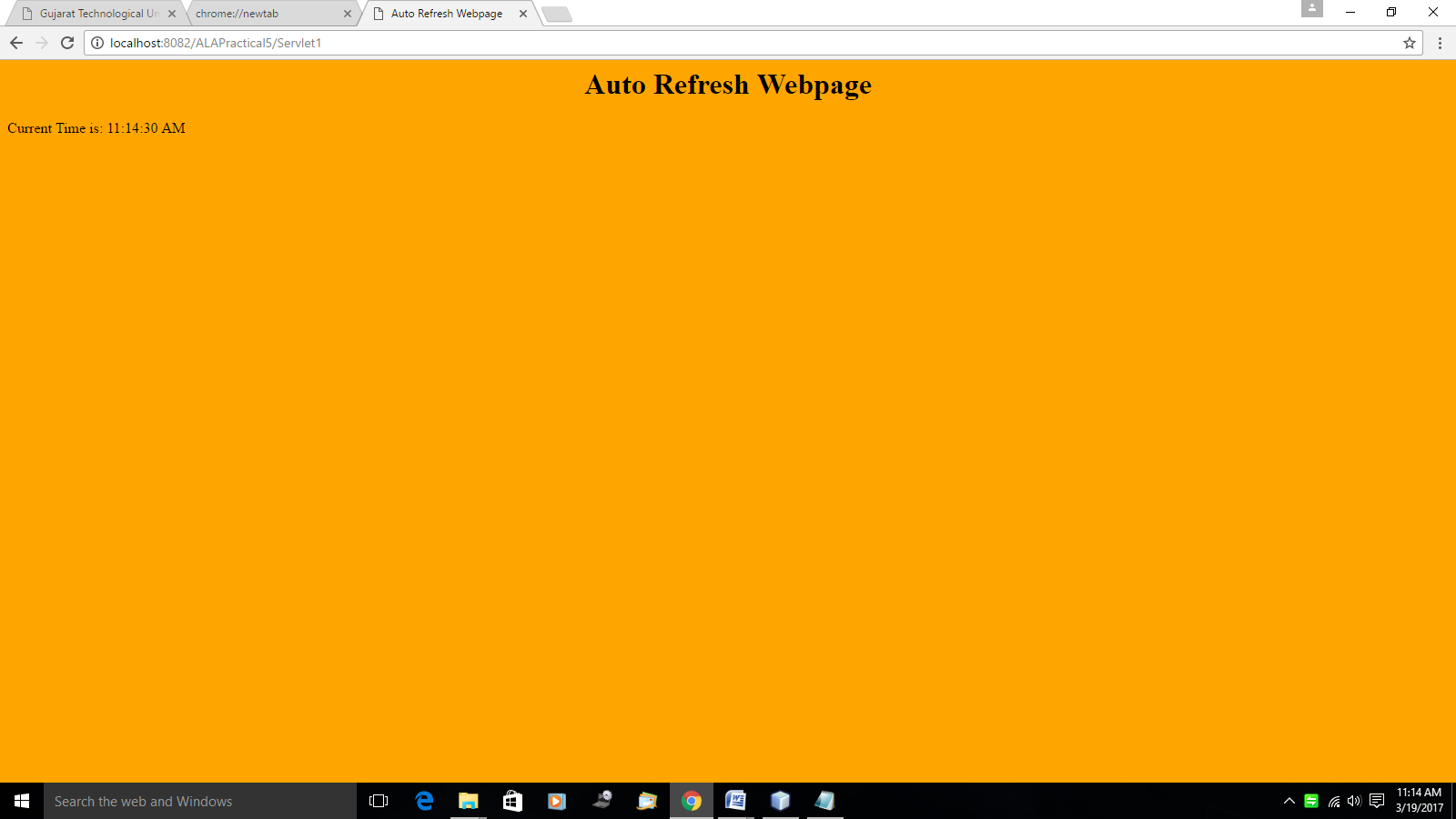
</web-app>

### ALA PRACTICAL 4 OUTPUT

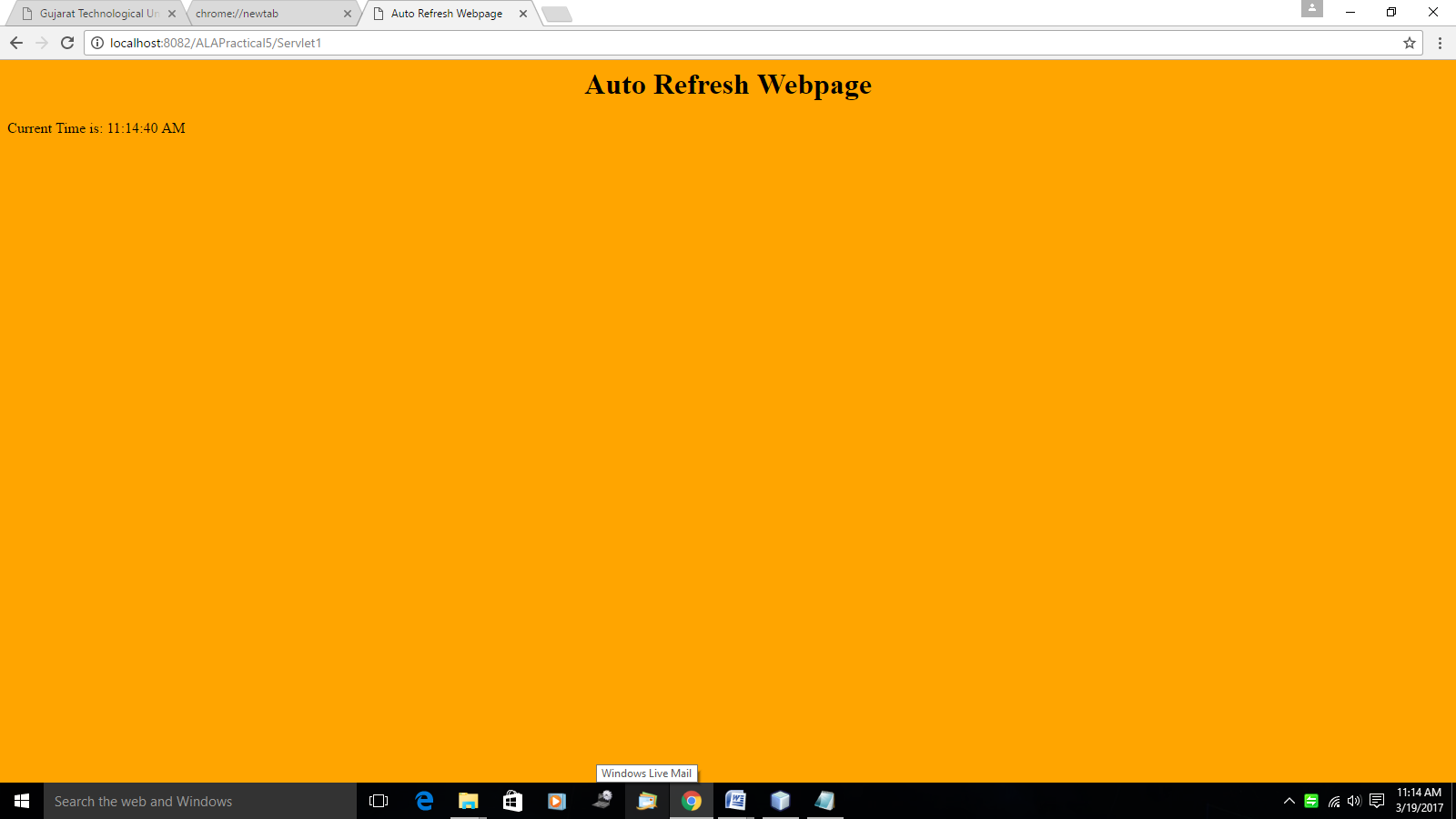


****

#### After 15 seconds-

****

#### After 25 seconds-

****

# ALA PRACTICAL 5

## AIM : Invoke the method from within template text using JSP elements (example o/p - Sum of 3 and 4 is 7.) Print above line to console as well using JSP elements.

### index.html

<html>

<head>

<title>Addition</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

</head>

<body>

<form action="AddJSP.jsp" method="post">

Number 1: <input type="text" name="n1"><br>

Number 2: <input type="text" name="n2"><br>

<input type="submit" value="Add">

</form>

</body>

</html>

### AddJSP.jsp

<%--

Document : AddJSP

Created on : Mar 24, 2017, 1:11:52 PM

Author : Himalay

--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Addition of 2 numbers</title>

</head>

<body>

<%

int x= Integer.parseInt(request.getParameter("n1"));

int y=Integer.parseInt(request.getParameter("n2"));

%>

<%!

int add(int a,int b)

{

return a+b;

}

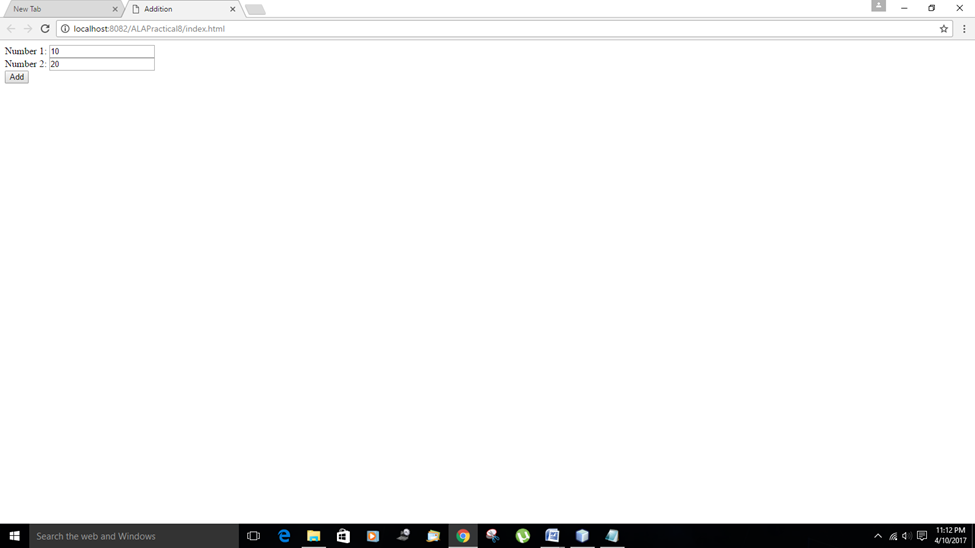
%>

<%= "Addition of " + x + " and " + y + " is " + add(x,y)%>

</body>

</html>

### ALA PRACTICAL 5 OUTPUT



# ALA PRACTICAL 6

## AIM : Write a servlet that sends the first 9 requests to Welcome Page, the next request to Reject Page, and then repeats. That is, every tenth request should get sent to Reject Page and the rest should get sent to Welcome Page.

### index.html

<html>

<head>

<title>Multiple Requests</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

</head>

<body>

<a href="Servlet1">Start</a>

</body>

</html>

### Servlet1.java (to redirect/dispatch requests)

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class Servlet1 extends HttpServlet {

public int count;

public void init()

{

count=1;

}

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

doPost(request,response);

}

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html");

PrintWriter out=response.getWriter();

if(count!=10){

out.println("<html><body><h2>Count is " +count+ " </h2></body></html>");

RequestDispatcher rd= request.getRequestDispatcher("welcome.html");

rd.include(request,response);

count++;

}

else{

out.println("<html><body><h2>Count is "+count+" </h2></body></html");

RequestDispatcher rd= request.getRequestDispatcher("reject.html");

rd.include(request,response);

count=1;

}

out.println("<html><body><a href=\"Servlet1\">Again</a></body></html");

}

}

### welcome.html

<html>

<head>

<title>Welcome</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

</head>

<body>

<h2>Welcome Page</h2>

</body>

</html>

### reject.html

<html>

<head>

<title>Reject</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

</head>

<body>

<h2>Reject Page</h2>

</body>

</html>

### web.xml

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

<web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee <http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd>">

<servlet>

<servlet-name>Servlet1</servlet-name>

<servlet-class>Servlet1</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>Servlet1</servlet-name>

<url-pattern>/Servlet1</url-pattern>

</servlet-mapping>

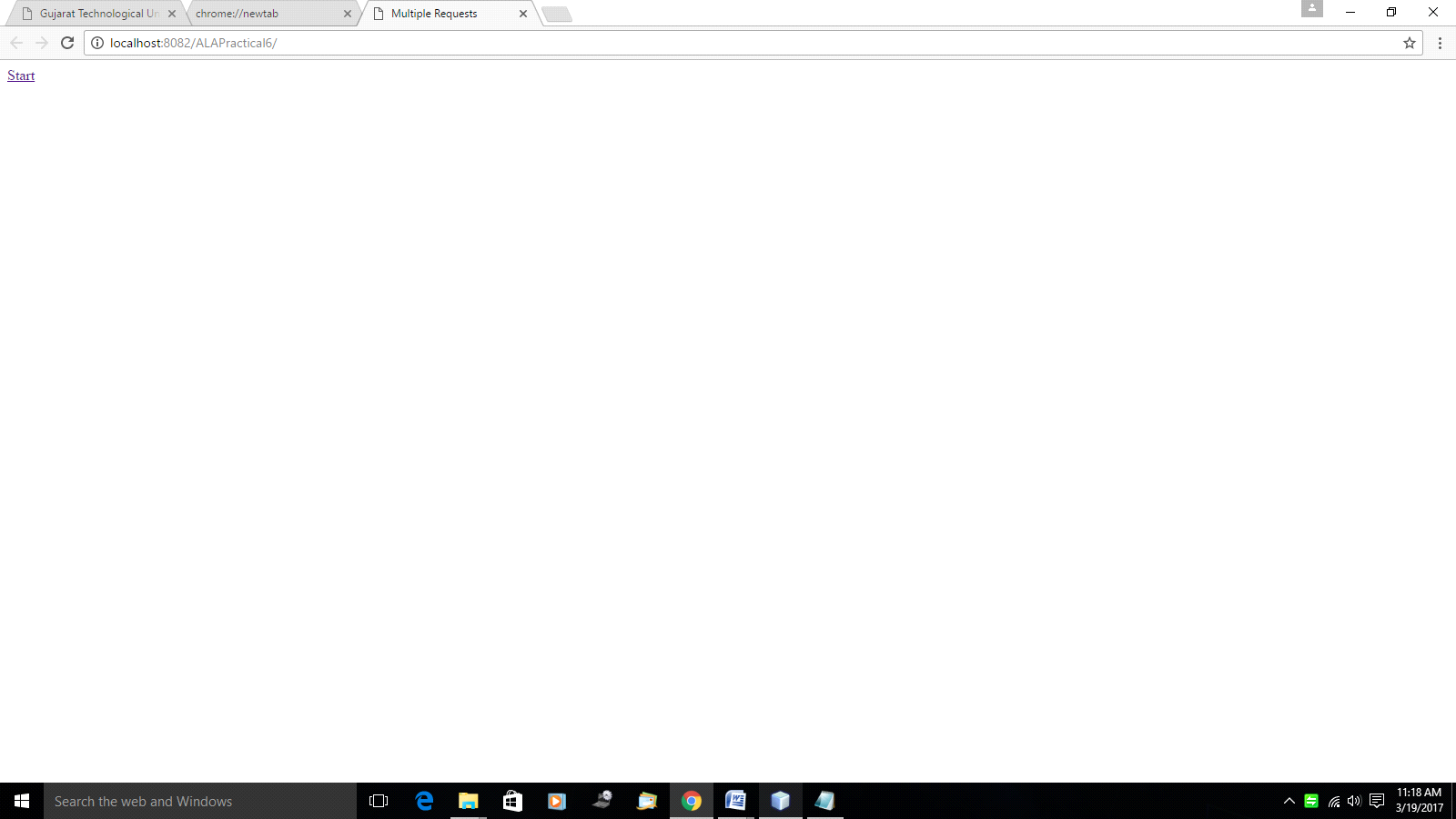
<welcome-file-list>

<welcome-file>index.html</welcome-file>

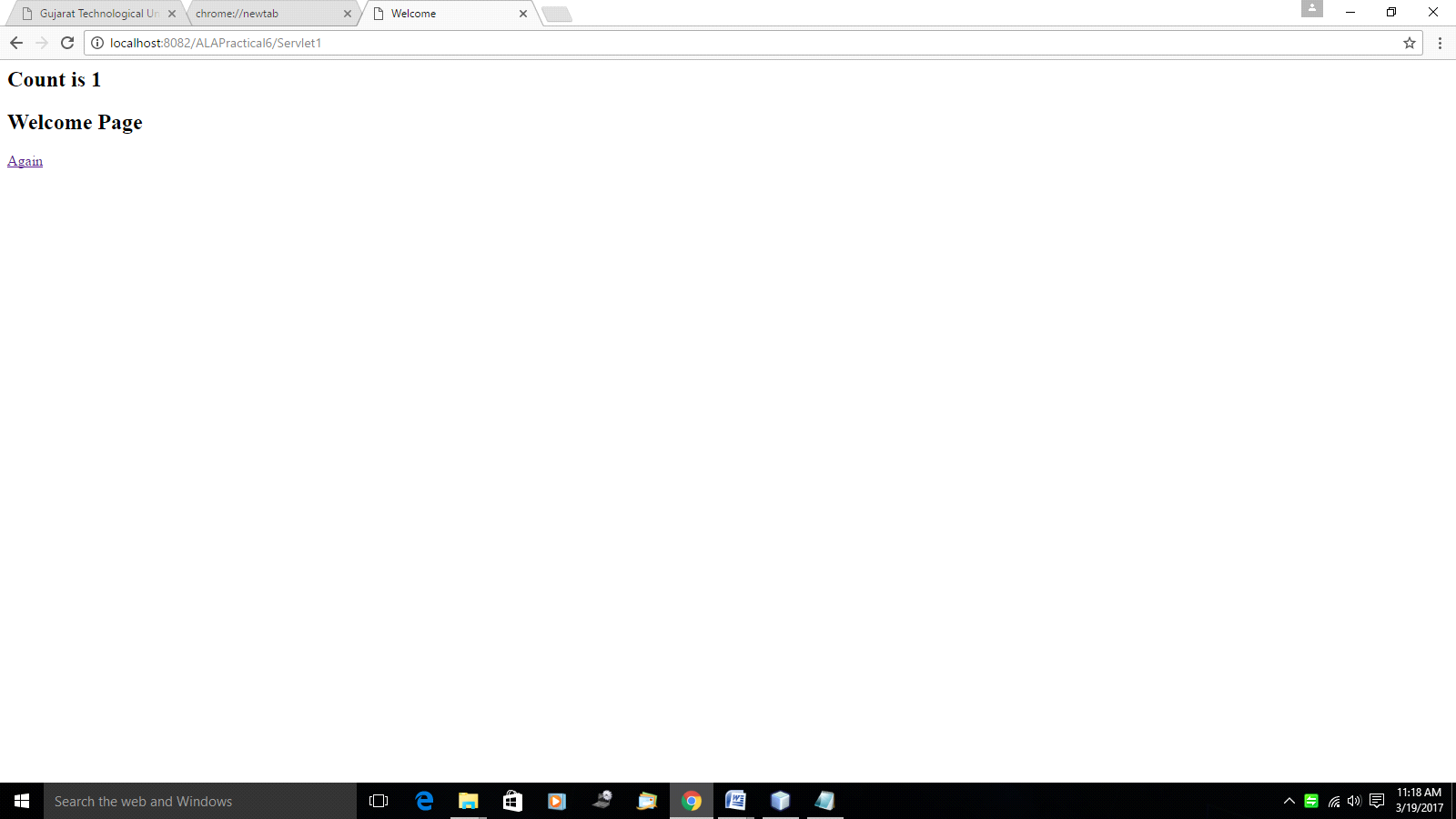
</welcome-file-list>

</web-app>

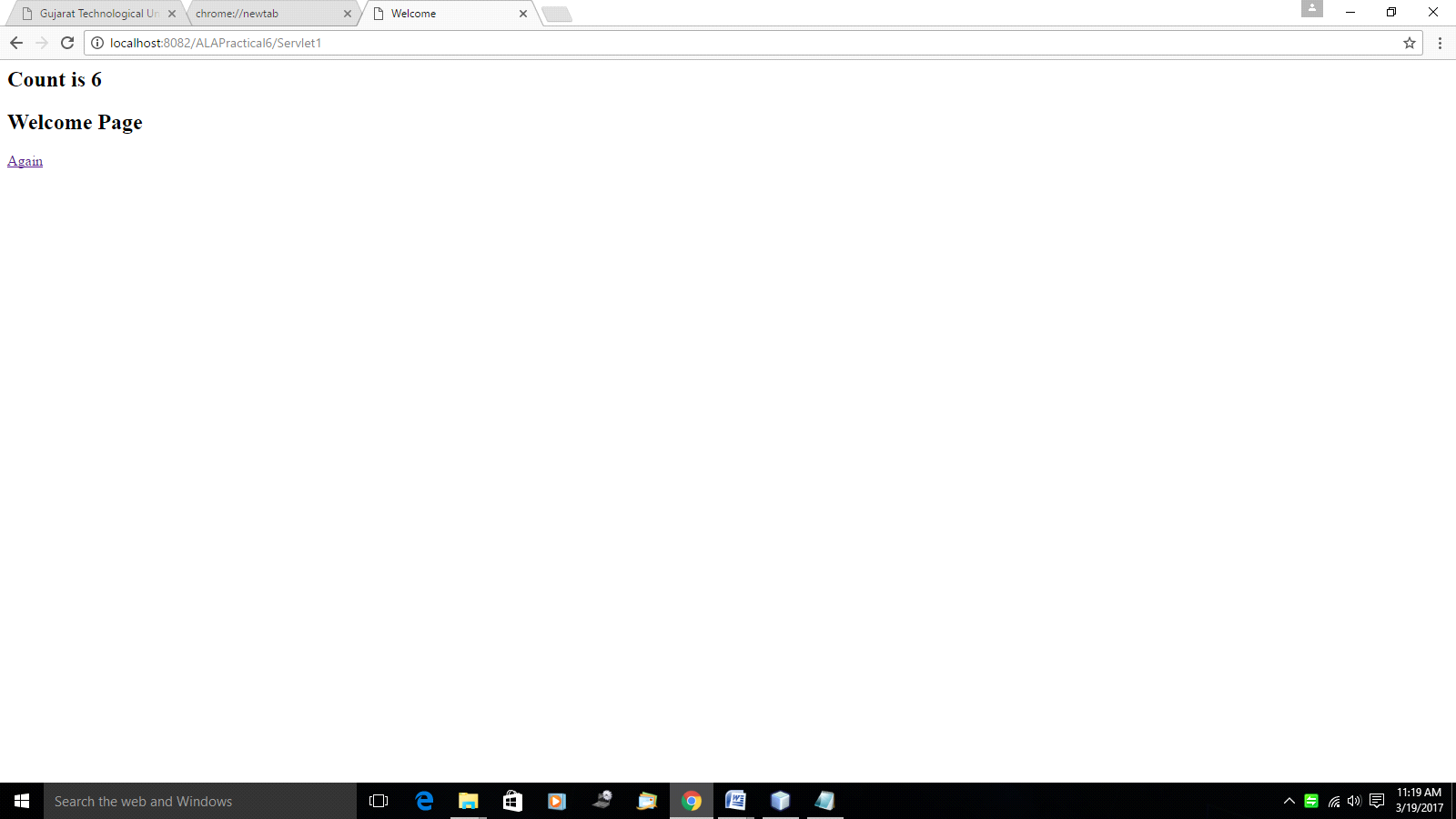
### ALA PRACTICAL 6 OUTPUT



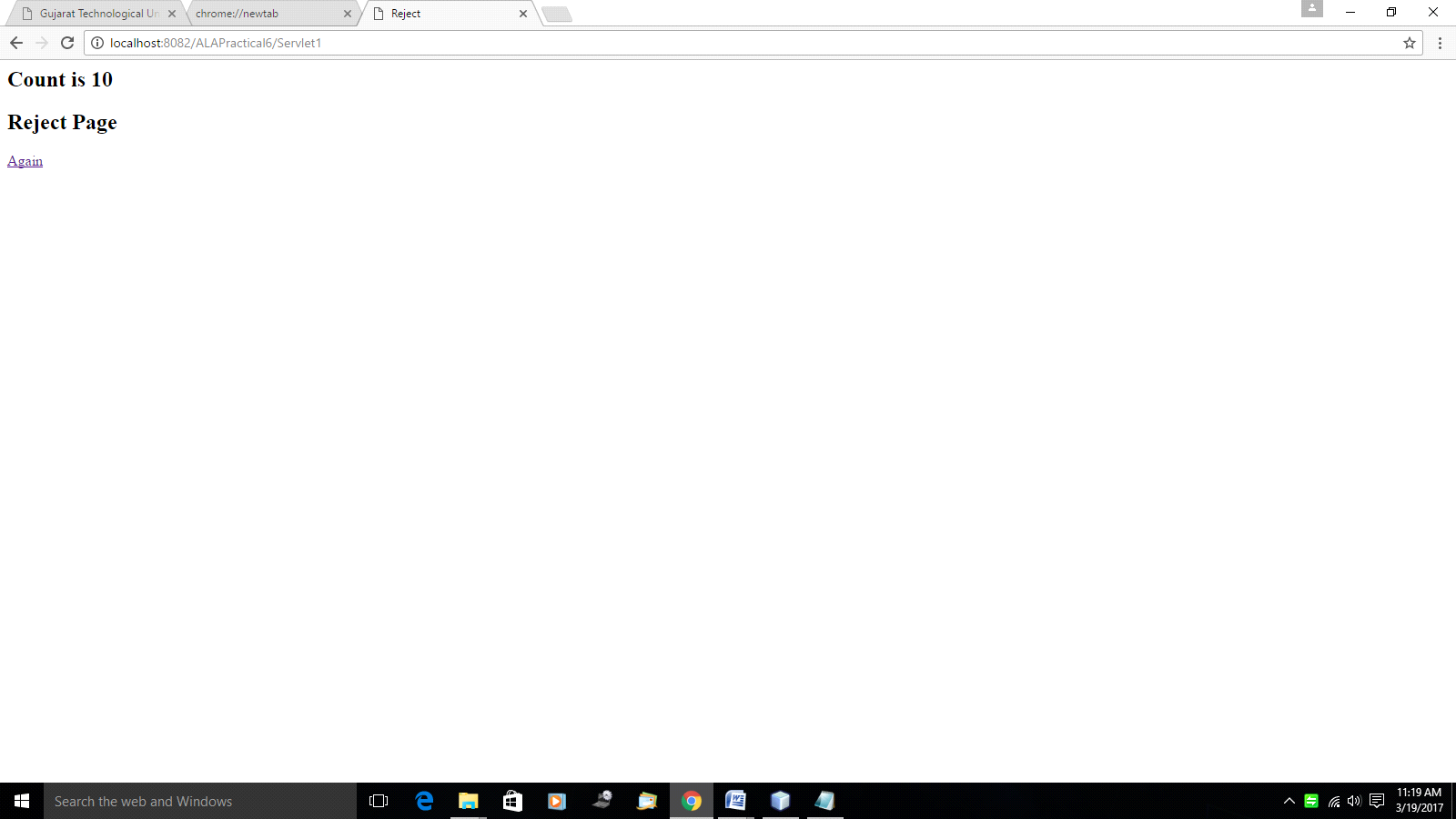
### Initially, count=1



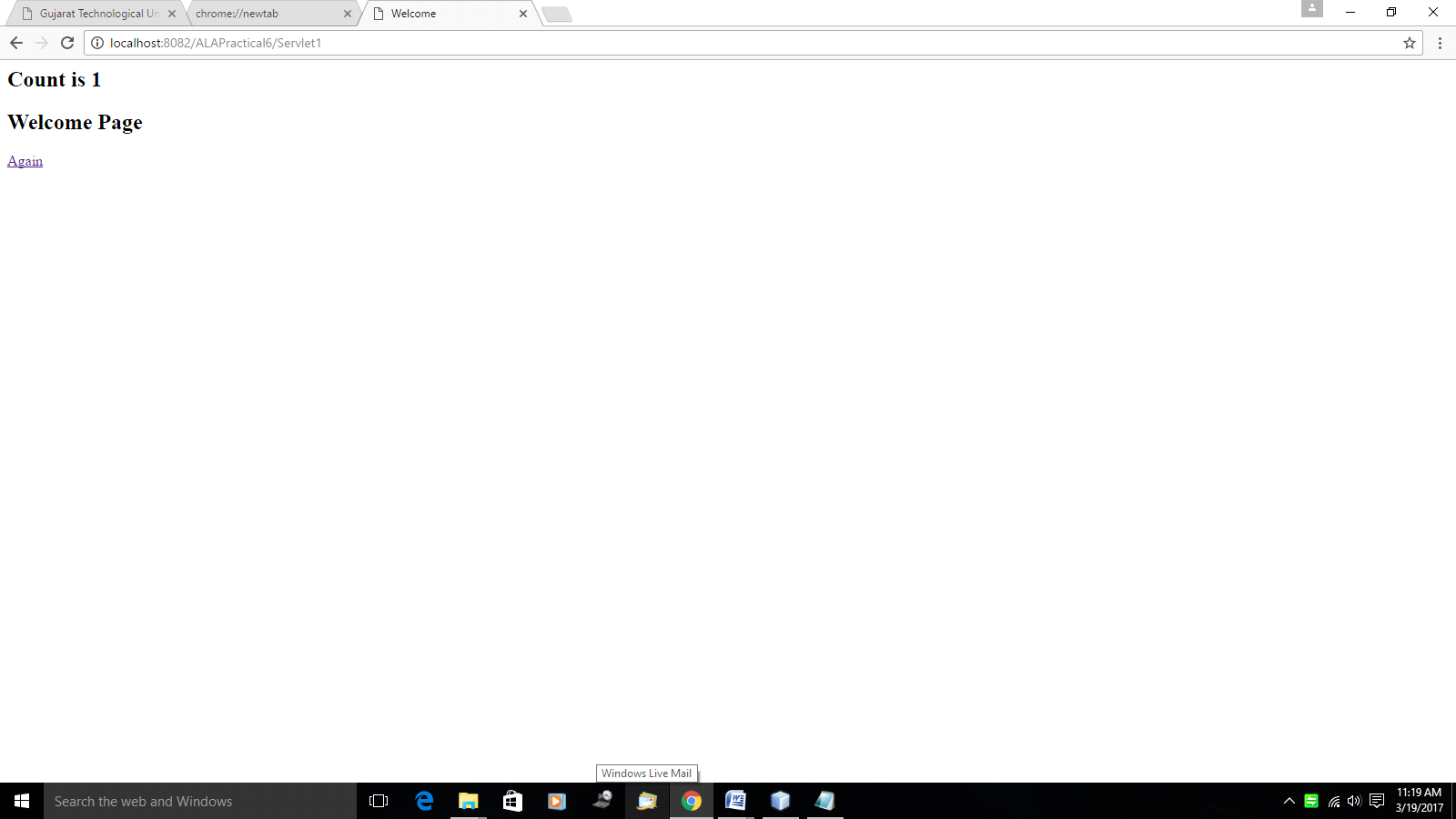
### After 6 requests-



### 10th request (Reject page)-



### After 10 requests (again to Welcome page)



# ALA PRACTICAL 7

## AIM :JSP Scripting Elements

## Do the following with only JSPs (and no servlets) and using JSP scripting elements

## Create a method that take two int numbers and return their sum as int.

## Invoke the method from within template text using JSP elements (example o/p - Sum of 3 and 4 is 7.) Print above line to console as well using JSP elements.

### index.html

<html>

<head>

<title>Addition</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

</head>

<body>

<form action="AddJSP.jsp" method="post">

Number 1: <input type="text" name="n1"><br>

Number 2: <input type="text" name="n2"><br>

<input type="submit" value="Add">

</form>

</body>

</html>

### AddJSP.jsp

<%--

Document : AddJSP

Created on : Mar 24, 2017, 1:11:52 PM

Author : Himalay

--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Addition of 2 numbers</title>

</head>

<body>

<%

int x= Integer.parseInt(request.getParameter("n1"));

int y=Integer.parseInt(request.getParameter("n2"));

%>

<%!

int add(int a,int b)

{

return a+b;

}

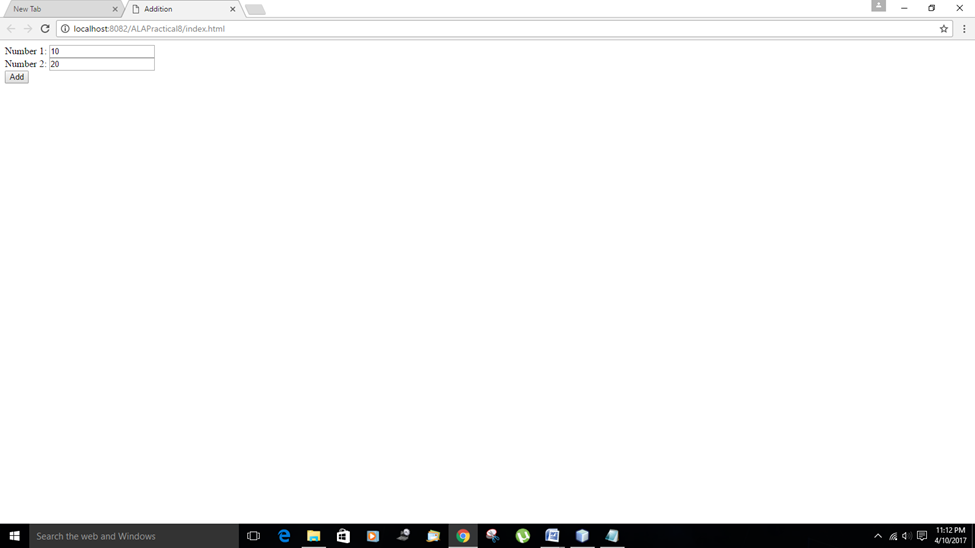
%>

<%= "Addition of " + x + " and " + y + " is " + add(x,y)%>

</body>

</html>

### ALA PRACTICAL 7 OUTPUT



# ALA PRACTICAL 8

## AIM : JSP Forms and Standard Actions

## Create a JSP web form to submit the name and age of an employee.

## Form should be submitted to a JSP file.

## Create a bean class called Employee for storing the name and age of the employee

## Within the JSP to which form is submitted, do the following using only standard actions-

## Declare an instance of the Employee JavaBean

## Populate the Employee object with values from form submission

## Retrieve the values and display it with proper template text:

## Employee name is John

## Employee age is 30

### index.jsp

<%--

Document : index

Created on : Mar 24, 2017, 1:59:58 PM

Author : Himalay

--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Employee form</title>

</head>

<body>

<form action="nextJSP.jsp">

Name: <input type="text" name="name">

Age: <input type="text" name="age">

<input type="submit" value="Go">

</form>

</body>

</html>

### nextJSP.jsp

<%--

Document : nextJSP

Created on : Mar 24, 2017, 2:01:48 PM

Author : Himalay

--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Welcome</title>

</head>

<body>

<jsp:useBean id="emp" class="test.Employee" scope="request">

<jsp:setProperty name="emp" property="\*" />

<jsp:getProperty property="name" name="emp"/><br>

<jsp:getProperty property="age" name="emp"/><br>

</jsp:useBean>

</body>

</html>

### test/Employee.java

package test;

public class Employee {

public Employee(){}

private String name;

private int age;

public void setName(String str){

name=str;

}

public void setAge(int x){

age=x;

}

public String getName(){

return name;

}

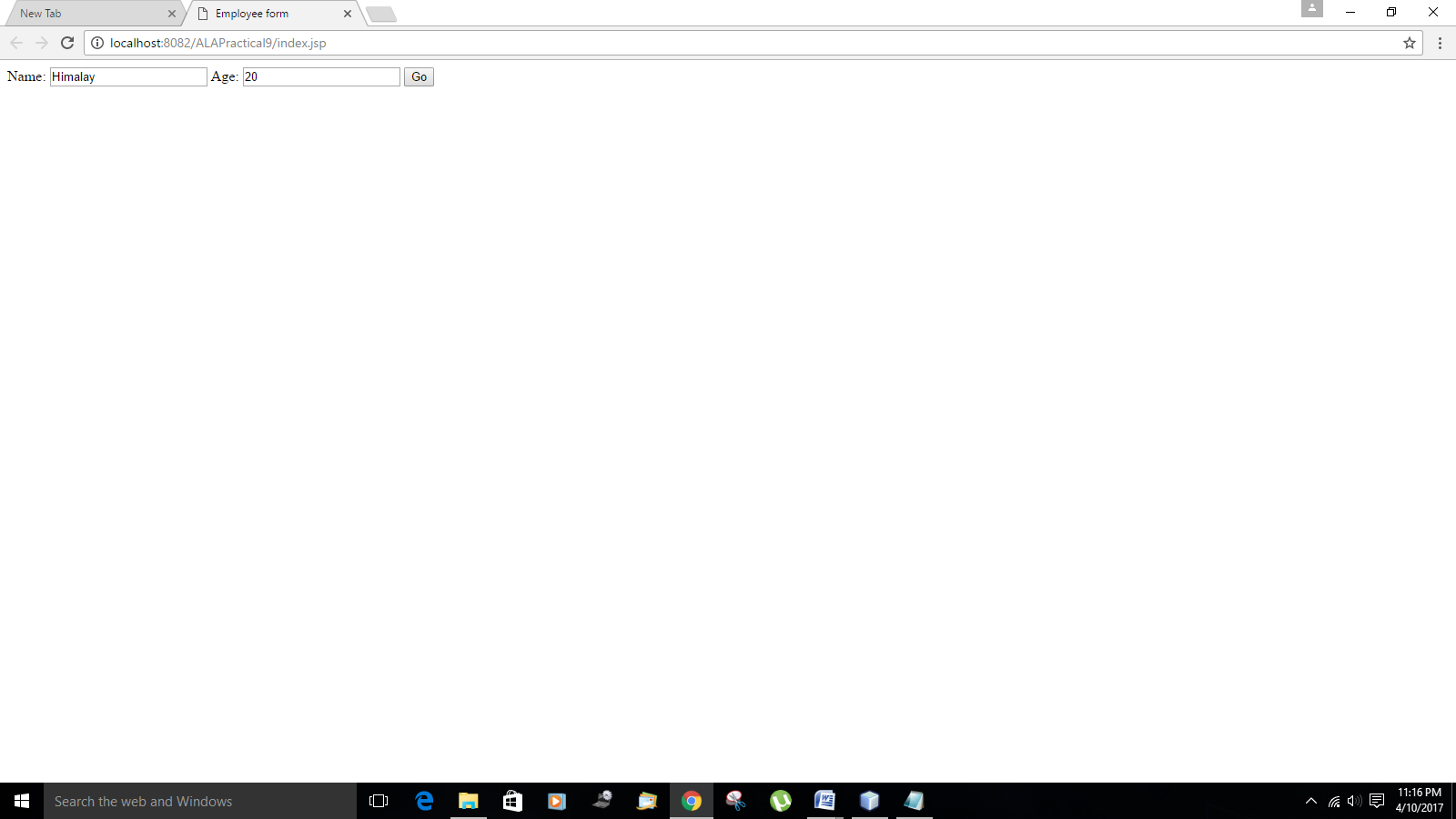
public int getAge(){

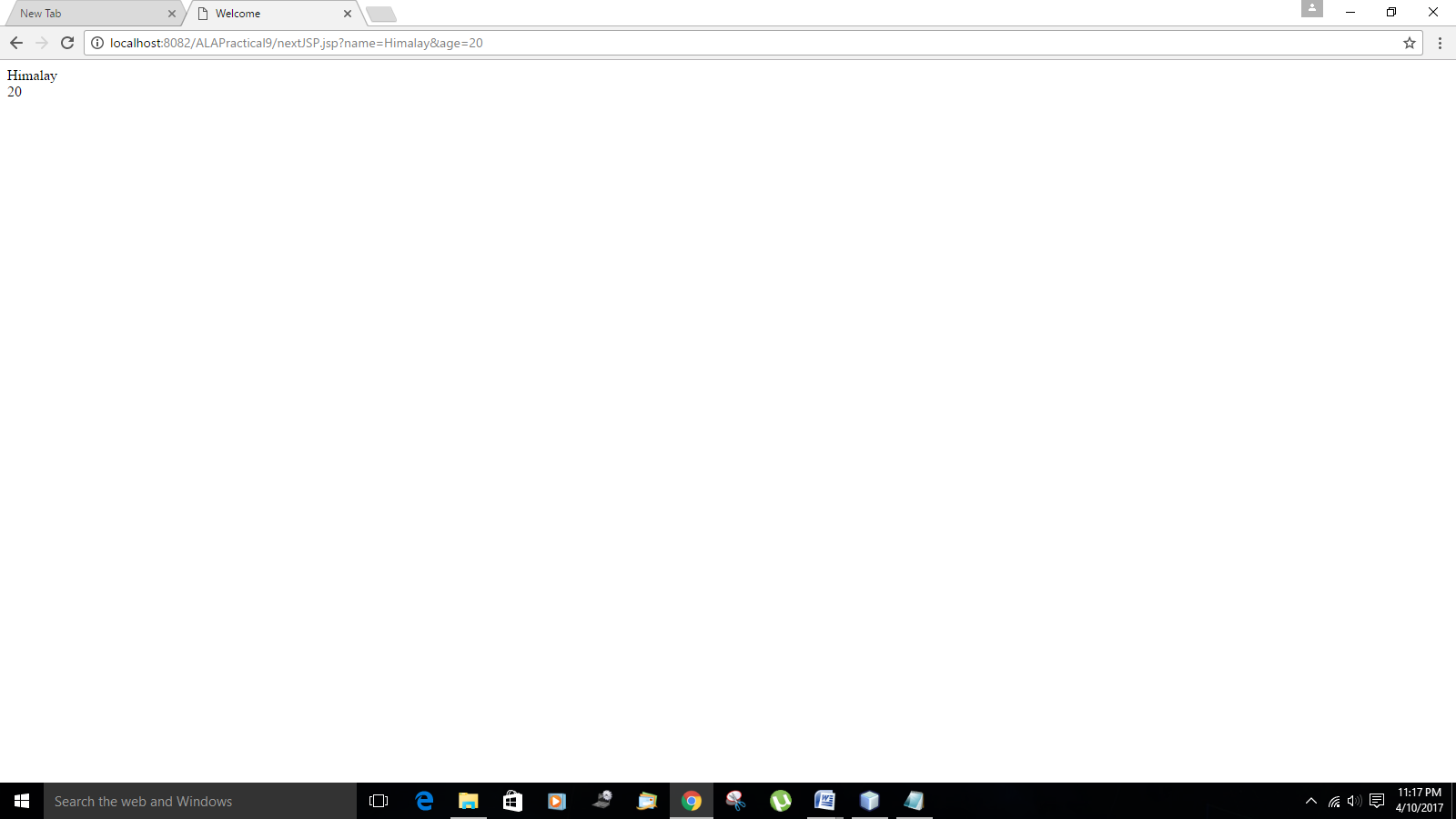
return age;

}

}

### ALA PRACTICAL 8 OUTPUT





# ALA PRACTICAL 9

## AIM : Create two simple JSP files.

## A JSP web form that will submit data to the second JSP

## Second JSP may simply print the values of the form submission

### index.jsp

<%--

Document : index

Created on : Apr 4, 2017, 11:19:23 AM

Author : Himalay

--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>JSP web form</title>

</head>

<body>

<form name="ff" method="post" action="show.jsp">

First Name- <input type="text" name="fName" /><br>

Last Name- <input type="text" name="lName" /><br>

<input type="radio" name="gender" value="Male"/>Male<br>

<input type="radio" name="gender" value="Female"/>Female<br>

Date of Birth- <input type="text" name="dob" /><br>

College- <input type="text" name="college" /><br>

Home town- <input type="text" name="home" /><br>

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

</form>

</body>

</html>

### show.jsp

<%--

Document : show

Created on : Apr 4, 2017, 11:24:09 AM

Author : Himalay

--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Show page</title>

</head>

<body>

<%

String fname=request.getParameter("fName");

String lname=request.getParameter("lName");

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

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

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

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

%>

<%= "First Name is " + fname + "<br>" %>

<%= "Last Name is " + lname + "<br>" %>

<%= "Gender is " + gender + "<br>" %>

<%= "Date of Birth is " + dob + "<br>" %>

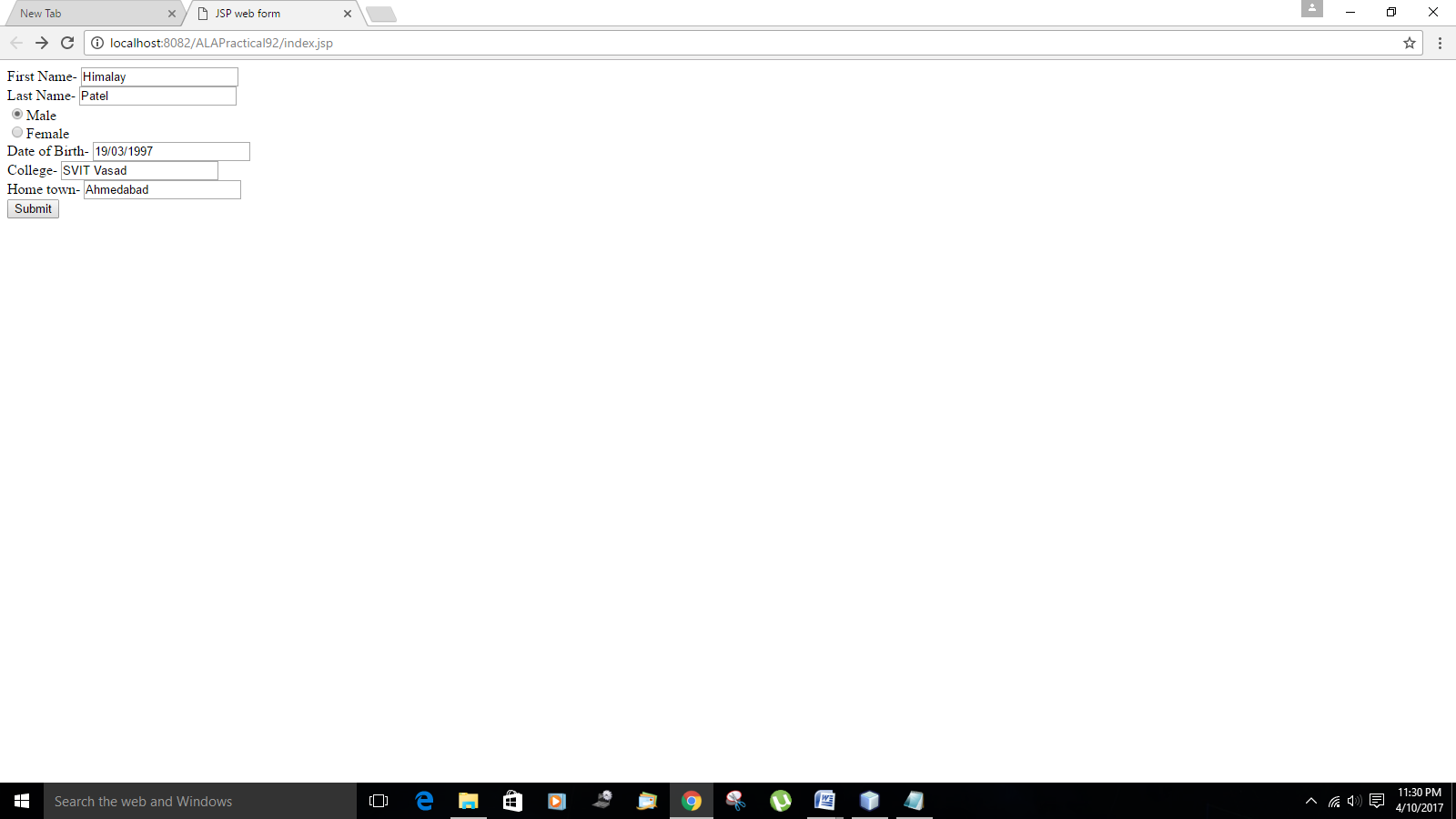
<%= "College is " + college + "<br>" %>

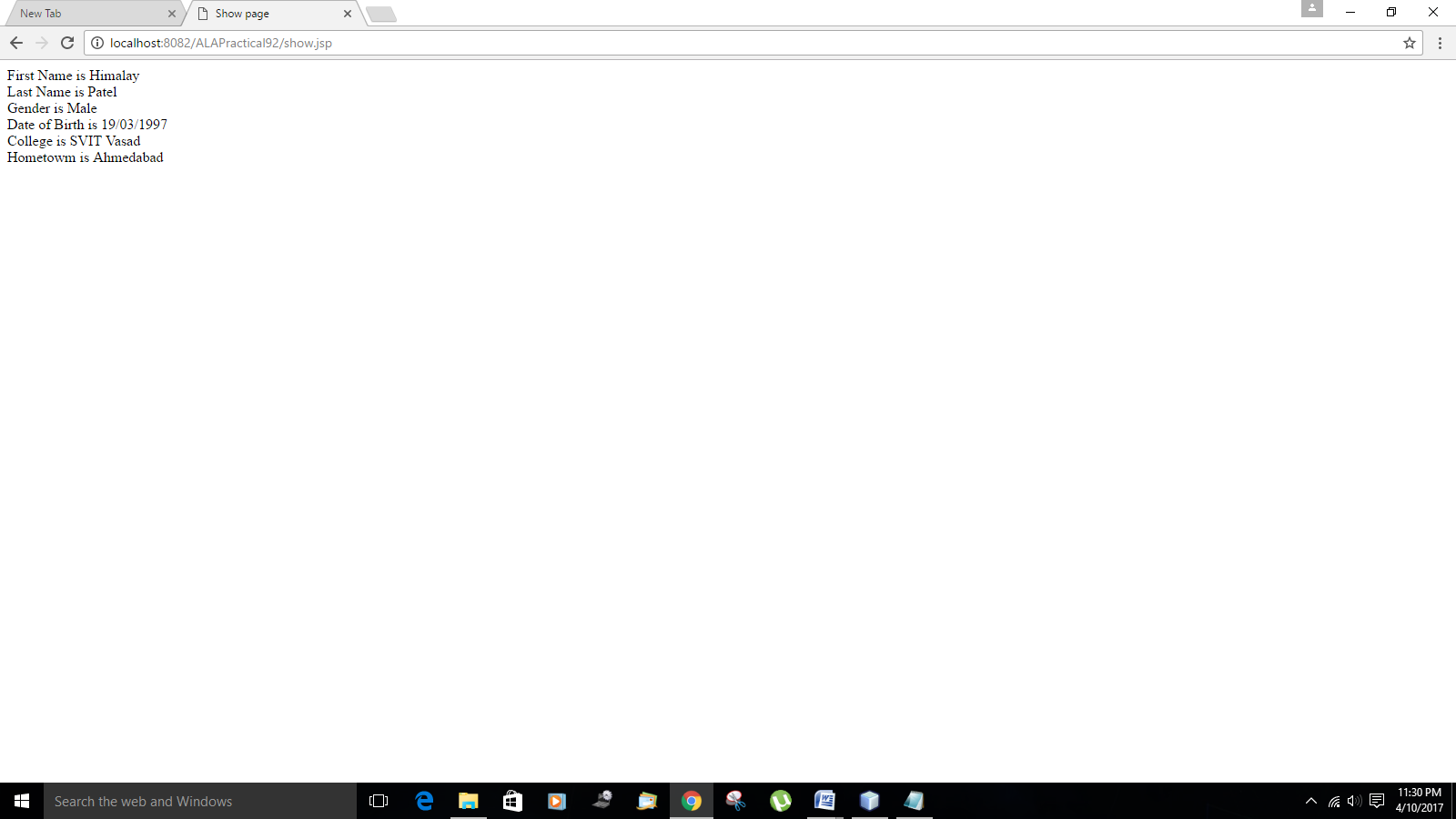
<%= "Hometowm is " + home + "<br>" %>

</body>

</html>

### ALA PRACTICAL 9 OUTPUT





# ALA PRACTICAL 10 (Use JSTL SQL taglib)

## AIM : Write a JSP which insert the details of a customer who register with the web site by using registration form.

### insert.html

<!DOCTYPE html>

<html>

<head>

<title>Registration form</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

</head>

<body>

<form name="f1" method="post" action="insert.jsp">

Name- <input type="text" name="name" /><br>

Phone number- <input type="text" name="phone" /><br>

Address- <textarea rows="5" name="address"></textarea><br>

<input type="submit" name="submit" />

<input type="reset" name="reset" />

</form>

</body>

</html>

### insert.jsp

<%--

Document : insert

Created on : Apr 4, 2017, 8:06:37 AM

Author : Himalay

--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix='c'%>

<%@ taglib uri="http://java.sun.com/jsp/jstl/sql" prefix='sql'%>

<sql:setDataSource var="dataSource" driver="com.mysql.jdbc.Driver"

url="jdbc:mysql://localhost:3306/himalay" user="root" password="himalaypatel" />

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Insert details</title>

</head>

<body>

<%

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

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

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

%>

<sql:update dataSource="${dataSource}" var="result">

insert into customer(name,phone,address) values (?,?,?);

<sql:param value="${param.name}" />

<sql:param value="${param.phone}" />

<sql:param value="${param.address}" />

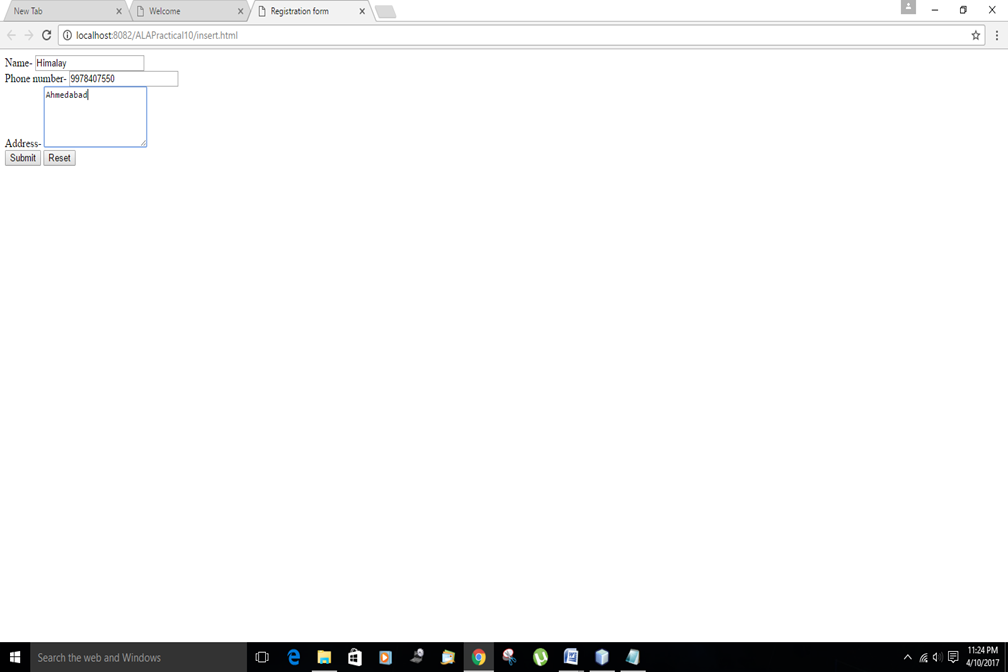
</sql:update>

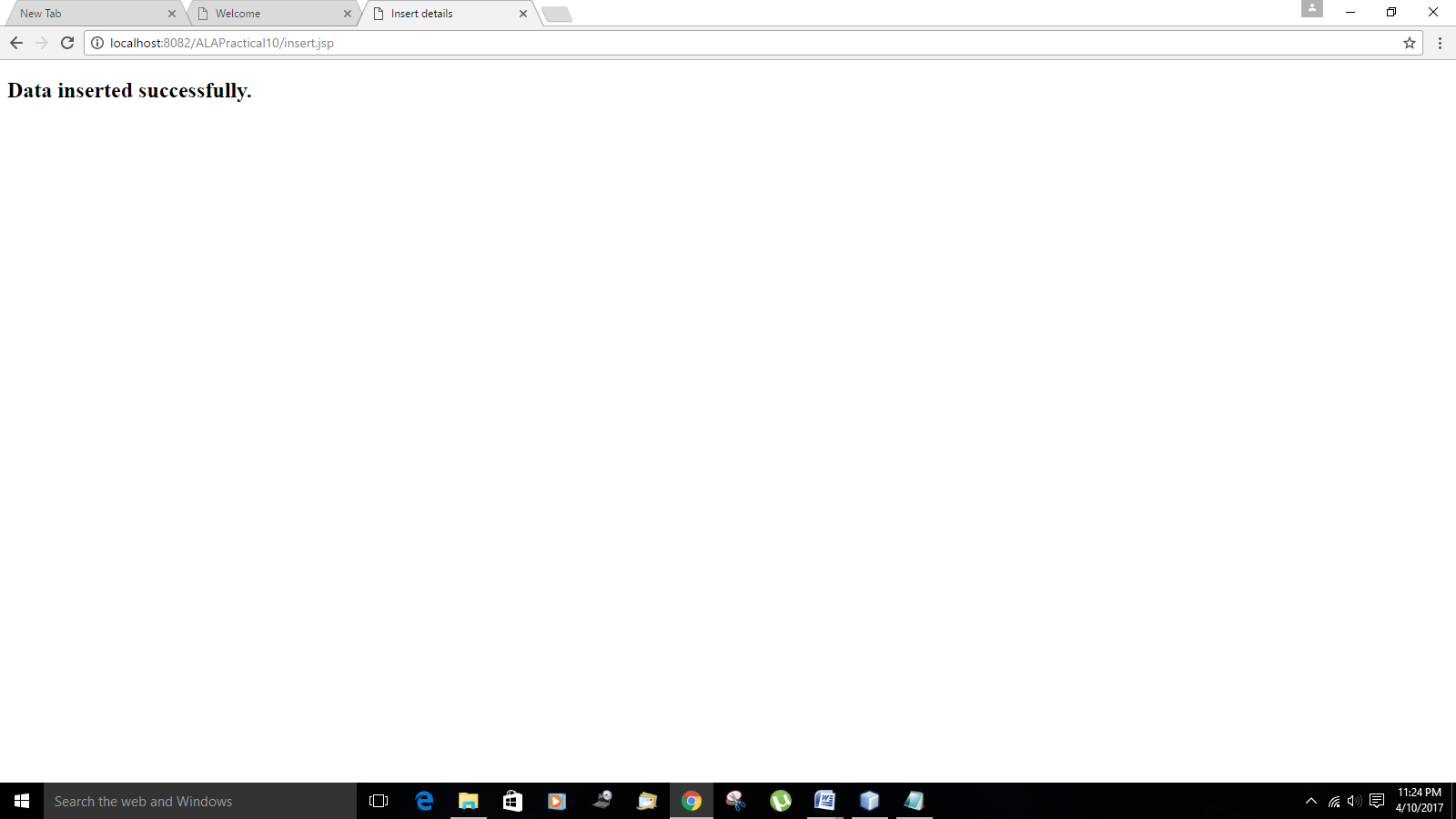
<h2>Data inserted successfully.</h2>

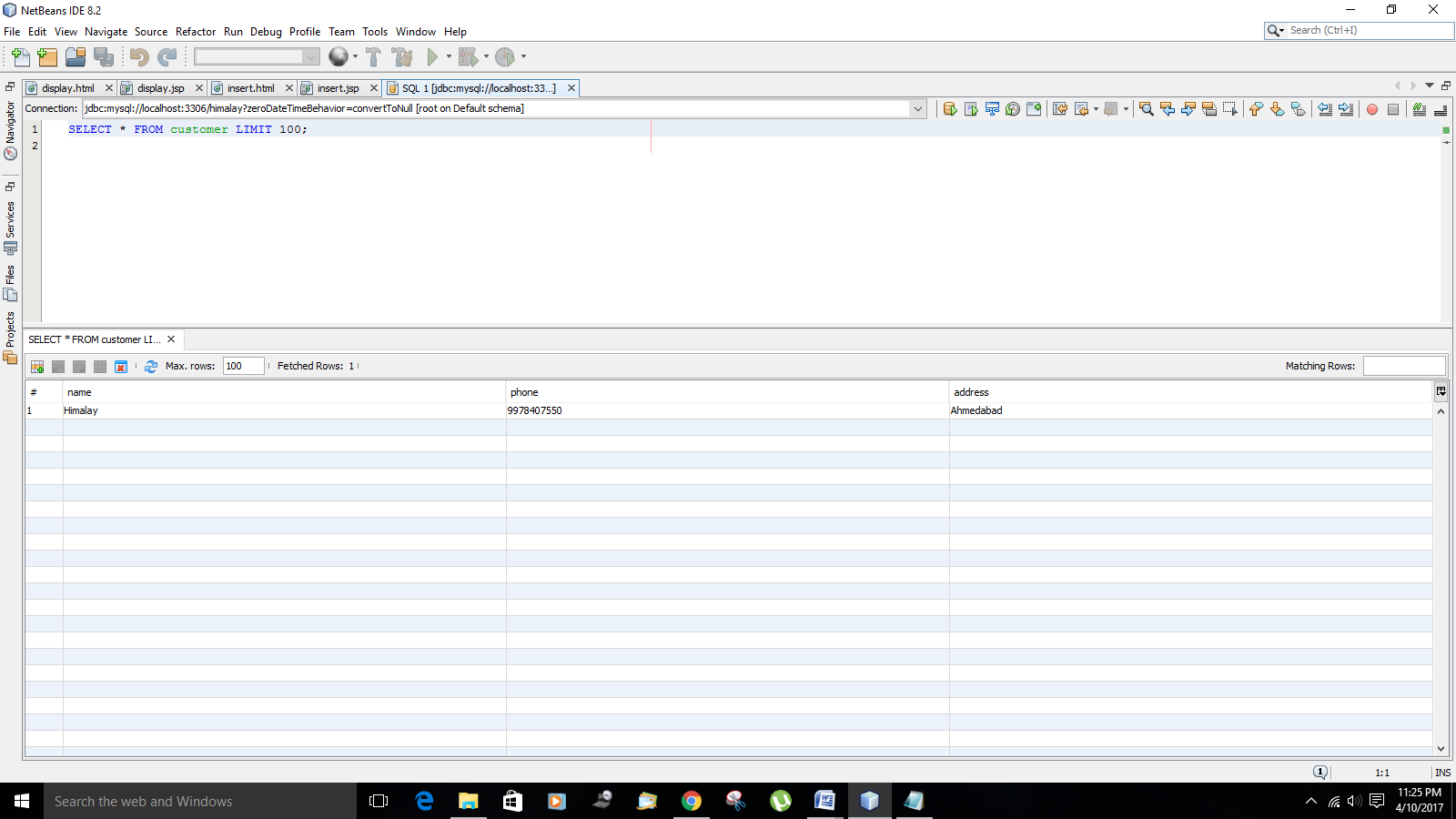
</body>

</html>

### ALA PRACTICAL 10 OUTPUT







# ALA PRACTICAL 11 (Use JSTL SQL taglib)

## AIM :Write a JSP which displays customer details in tabular form by iterating through the database customer table.

### display.html

<html>

<head>

<title>Display details</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

</head>

<body>

<form name="f2" method="post" action="display.jsp">

<input type="submit" name="submit" value="Display details of customer table" />

</form>

</body>

</html>

### display.jsp

<%--

Document : display

Created on : Apr 4, 2017, 8:38:17 AM

Author : Himalay

--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix='c'%>

<%@ taglib uri="http://java.sun.com/jsp/jstl/sql" prefix='sql'%>

<sql:setDataSource var="dataSource" driver="com.mysql.jdbc.Driver"

url="jdbc:mysql://localhost:3306/himalay" user="root" password="himalaypatel" />

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Display details</title>

<style type="text/css">

table{

border-collapse:collapse;

}

</style>

</head>

<body>

<sql:query var="customers" dataSource="${dataSource}">

select \* from customer;

</sql:query>

<center><h2>Customer table</h2></center>

<table border="1" cellspacing="3" cellpadding="5" align="center">

<tr>

<c:forEach var="column" items="${customers.columnNames}" >

<td><c:out value="${column}" /></td>

</c:forEach>

</tr>

<c:forEach var="row" items="${customers.rows}">

<tr>

<td><c:out value="${row.name}" /></td>

<td><c:out value="${row.phone}" /></td>

<td><c:out value="${row.address}" /></td>

</tr>

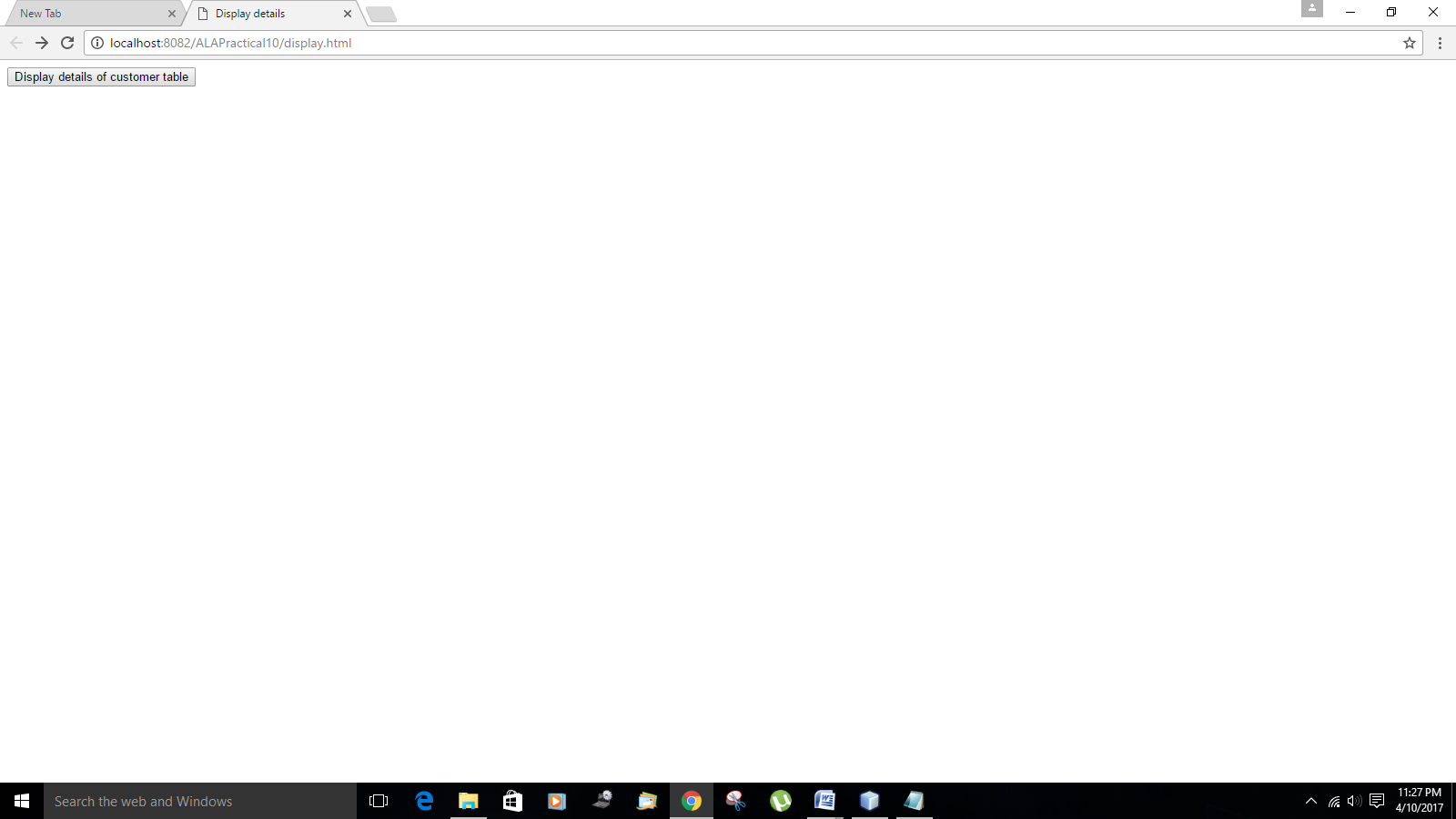
</c:forEach>

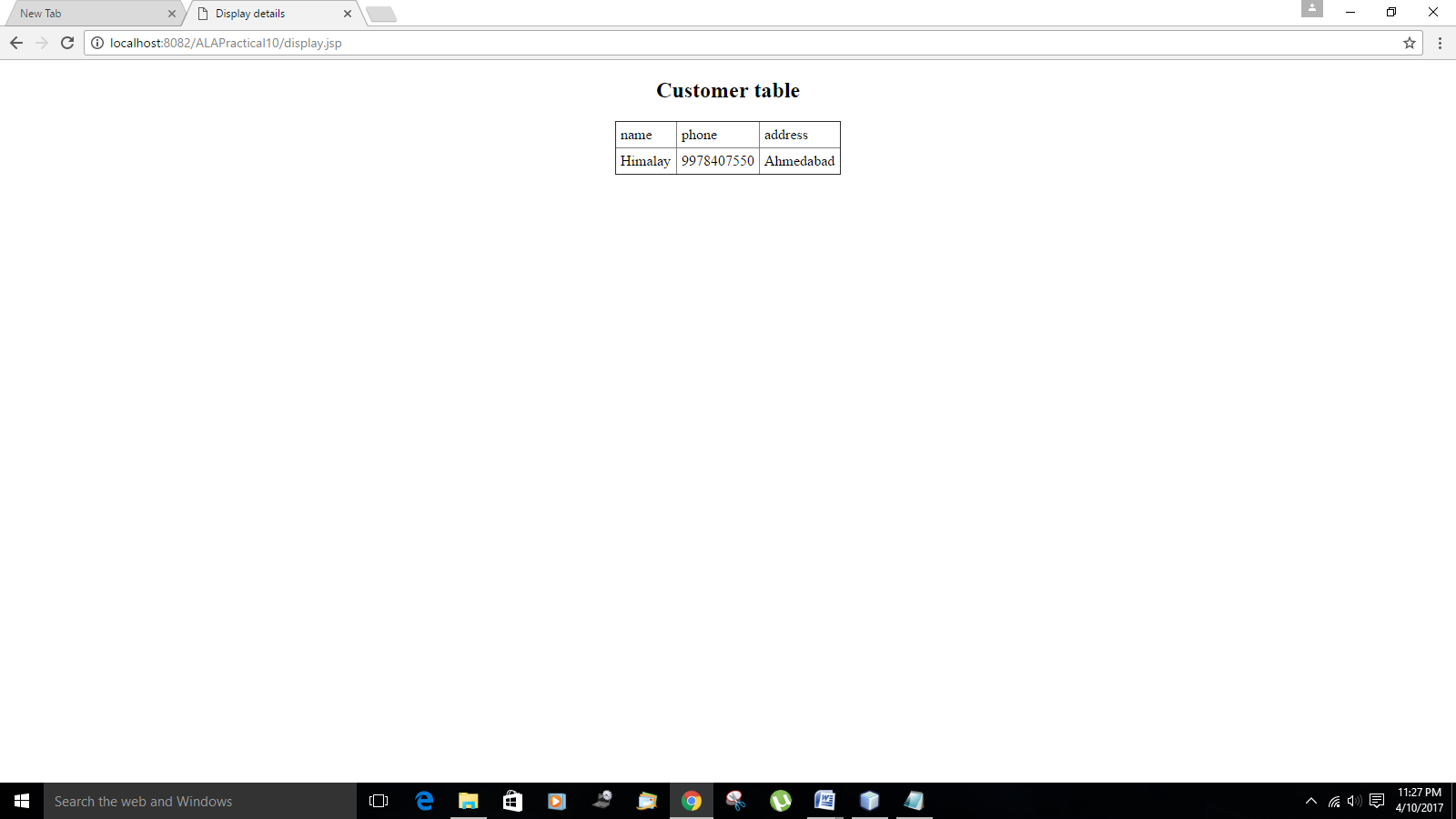
</table>

</body>

</html>

### ALA PRACTICAL 11 OUTPUT

****

****

# ALA PRACTICAL 12

## AIM : Create a servlet that makes an array of Name objects, which have firstName and lastName properties. Have your JSP page make an HTML table with first names in the left table cell and last names in the right table cell. Use the JSP 2.0 expression language as well as JSTL.

### index.html

<html>

<head>

<title>Name objects</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

</head>

<body>

<form name="f" method="post" action="Servlet1">

<button type="submit">Display Names in tabular form</button>

</form>

</body>

</html>

### test/Servlet1.java

package test;

import test.Name;

import java.io.IOException;

import java.io.PrintWriter;

import java.util.\*;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

public class Servlet1 extends HttpServlet {

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

doPost(request,response);

}

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

Name[] n;

n = new Name[3];

for(int i=0;i<3;i++){

Name m=new Name();

n[i]=m;

}

n[0].setFirstName("Himalay");

n[0].setLastName("Patel");

n[1].setFirstName("Parva");

n[1].setLastName("Patel");

n[2].setFirstName("Kaival");

n[2].setLastName("Kothari");

request.setAttribute("names",n);

RequestDispatcher rd= request.getRequestDispatcher("go.jsp");

rd.forward(request,response);

}

}

### test/Name.java

package test;

public class Name {

private String firstName;

private String lastName;

public Name(){ }

public void setFirstName(String fname){

firstName=fname;

}

public void setLastName(String lname){

lastName=lname;

}

public String getFirstName(){

return firstName;

}

public String getLastName(){

return lastName;

}

}

### go.jsp

<%--

Document : go

Created on : Apr 4, 2017, 11:55:28 AM

Author : Himalay

--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix='c'%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Display Names</title>

<style type="text/css">

table{

border-collapse:collapse;

}

</style>

</head>

<body>

<center><h2>In JSP file</h2></center>

<center>

<table border="1" cellspacing="3" cellpadding="5">

<tr>

<th>First Name</th>

<th>Last Name</th>

</tr>

<c:forEach var="n" items="${names}">

<tr>

<td><c:out value="${n.firstName}" /></td>

<td><c:out value="${n.lastName}" /></td>

</tr>

</c:forEach>

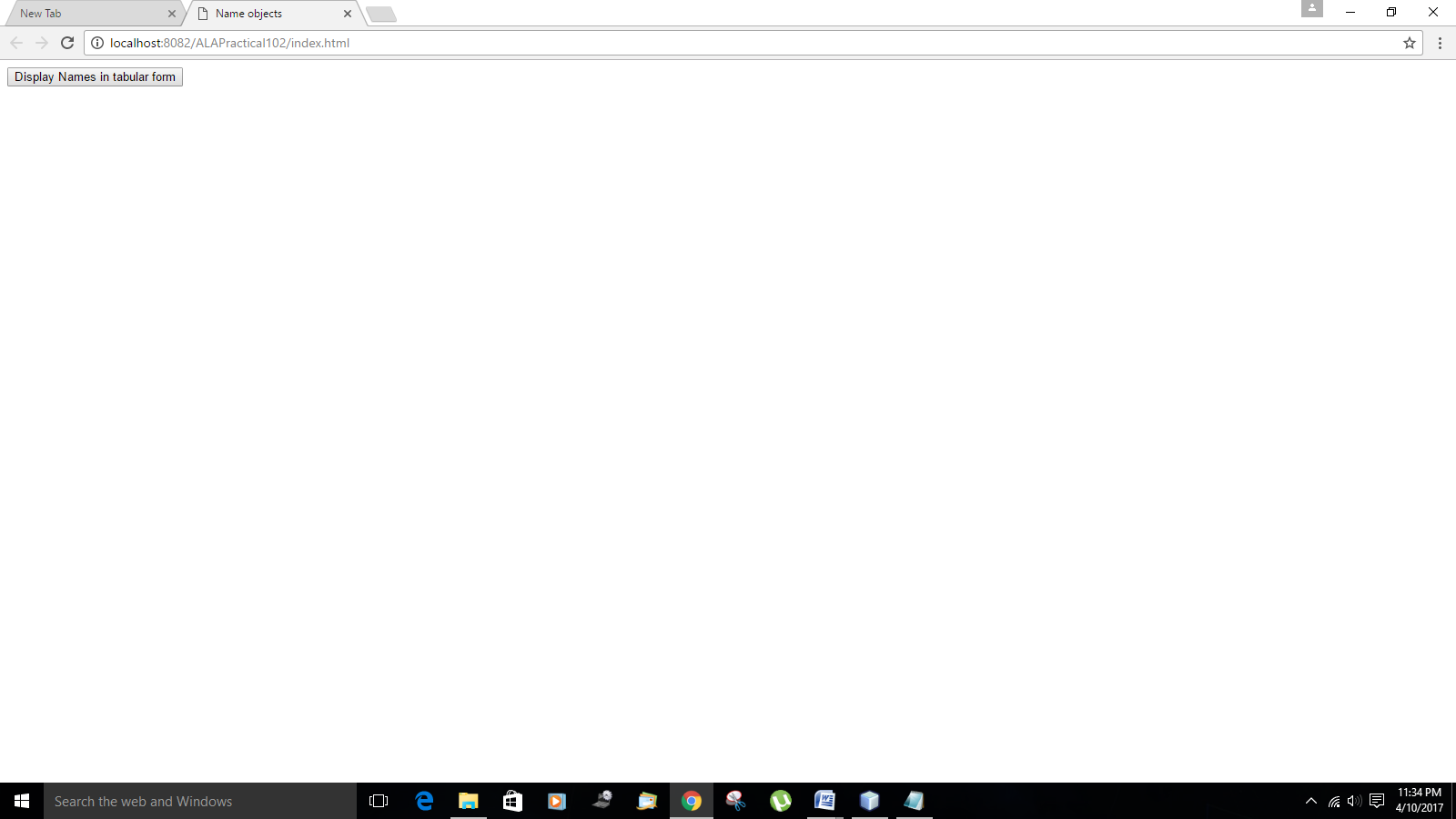
</table>

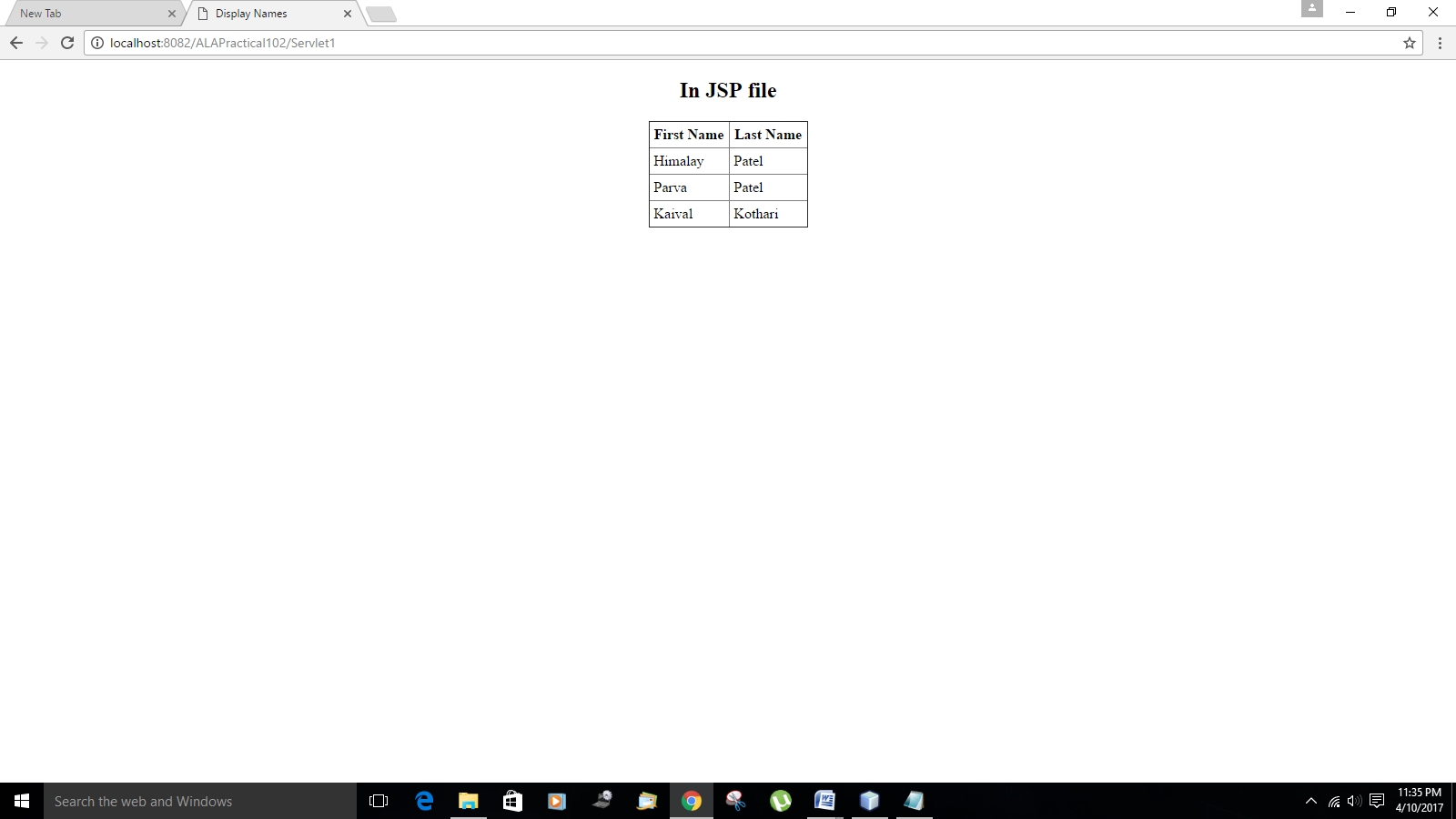
</center>

</body>

</html>

### ALA PRACTICAL 12 OUTPUT

****

****

# ALA PRACTICAL 13

## AIM : Using JSTL XML taglib

## Write a JSP which parses an XML file containing Customer details like Name, Age, phoneno, email.. Displays the result in tabular form.

### student.xml

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

<customers>

<customer>

<name>Himalay</name>

<age>20</age>

<phoneno>9978407550</phoneno>

<email>himalayp8@gmail.com</email>

</customer>

<customer>

<name>Parva</name>

<age>21</age>

<phoneno>9039039022</phoneno>

<email>p@gmail.com</email>

</customer>

<customer>

<name>Kaival</name>

<age>19</age>

<phoneno>9990009990</phoneno>

<email>k@gmail.com</email>

</customer>

<customer>

<name>Rajan</name>

<age>20</age>

<phoneno>1234567890</phoneno>

<email>r@gmail.com</email>

</customer>

<customer>

<name>Tirth</name>

<age>21</age>

<phoneno>9998887770</phoneno>

<email>t@gmail.com</email>

</customer>

</customers>

### ParseXML.jsp

<%--

Document : ParseXML

Created on : Apr 4, 2017, 4:37:27 PM

Author : Himalay

--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix='c'%>

<%@taglib uri="http://java.sun.com/jsp/jstl/xml" prefix='x'%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Parsing XML document</title>

<style type="text/css">

table{

border-collapse:collapse;

}

</style>

</head>

<body>

<c:import var="customerInfo" url="http://localhost:7777/practicals/customer.xml"/>

<x:parse xml="${customerInfo}" var="output"/>

<center><h2>Parsing customer.xml and displaying in tabular form</h2></center>

<center><table border="1" cellspacing="3" cellpadding="5">

<tr>

<th>Name</th>

<th>Age</th>

<th>Phone number</th>

<th>Email id</th>

</tr>

<x:forEach select="$output/customers/customer" var="xx">

<tr>

<td><x:out select="$xx/name" /></td>

<td><x:out select="$xx/age" /></td>

<td><x:out select="$xx/phoneno" /></td>

<td><x:out select="$xx/email" /></td>

</tr>

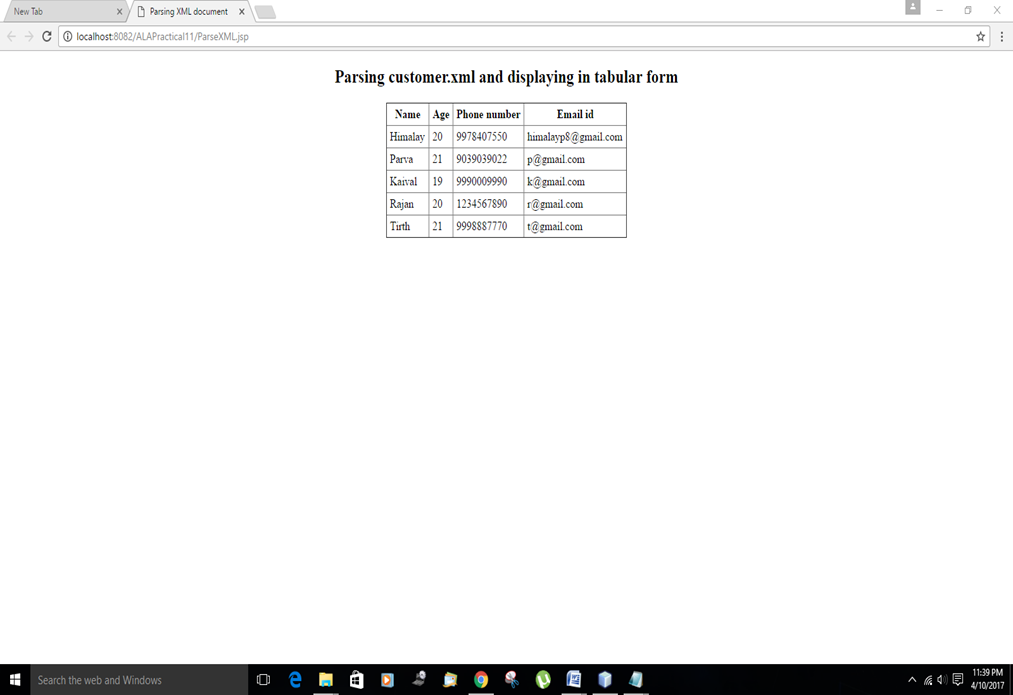
</x:forEach>

</table></center>

</body>

</html>

### ALA PRACTICAL 13 OUTPUT



# ALA PRACTICAL 14

## AIM : Make a JSF page which takes input for two numbers and has command buttons for different operation. When any button is clicked that operation is performed and result will be displayed in a text box. Should show an error message if any number is missing.

### index.xhtml

<?xml version='1.0' encoding='UTF-8' ?>

<!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"

xmlns:h="http://xmlns.jcp.org/jsf/html">

<h:head>

<title>Calculator using JSF</title>

</h:head>

<h:body>

<h2><h:outputText value="Simple Calculator using JSF" /></h2>

<h:form id="f1">

<h:outputText value="Enter a-"/>

<h:inputText value="#{test.a}" /><br></br>

<h:outputText value="Enter b-"/>

<h:inputText value="#{test.b}" /><br></br>

<h:commandButton action="#{test.add}" value="Add" /><br></br>

<h:commandButton action="#{test.subtract}" value="Subtract" /><br></br>

<h:commandButton action="#{test.multiply}" value="Multiply" /><br></br>

<h:commandButton action="#{test.divide}" value="Divide" /><br></br>

<h:outputText value="Answer is-"/>

<h:inputText value="#{test.result}" /><br></br>

<h:outputText value="#{test.error}"/><br></br>

</h:form>

</h:body>

</html>

### bean/test.java

package bean;

import javax.faces.bean.ManagedBean;

import javax.faces.bean.RequestScoped;

@ManagedBean

@RequestScoped

public class test {

private String a;

private String b;

private int result;

private String error;

public test(){ }

public String getA(){

return a;

}

public void setA(String str){

a=str;

}

public String getB(){

return b;

}

public void setB(String str){

b=str;

}

public int getResult(){

return result;

}

public void setResult(int str){

result=str;

}

public String getError(){

return error;

}

public void setError(String str){

error=str;

}

public void add(){

if(a.equals("") || b.equals("")){

error="One of the two numbers is missing.";

result=-9999999;

return;

}

result=Integer.parseInt(a)+Integer.parseInt(b);

}

public void subtract(){

if(a.equals("") || b.equals("")){

error="One of the two numbers is missing.";

result=-9999999;

return;

}

result=Integer.parseInt(a)-Integer.parseInt(b);

}

public void multiply(){

if(a.equals("") || b.equals("")){

error="One of the two numbers is missing.";

result=-9999999;

return;

}

result=Integer.parseInt(a)\*Integer.parseInt(b);

}

public void divide(){

if(a.equals("") || b.equals("")){

error="One of the two numbers is missing.";

result=-9999999;

return;

}

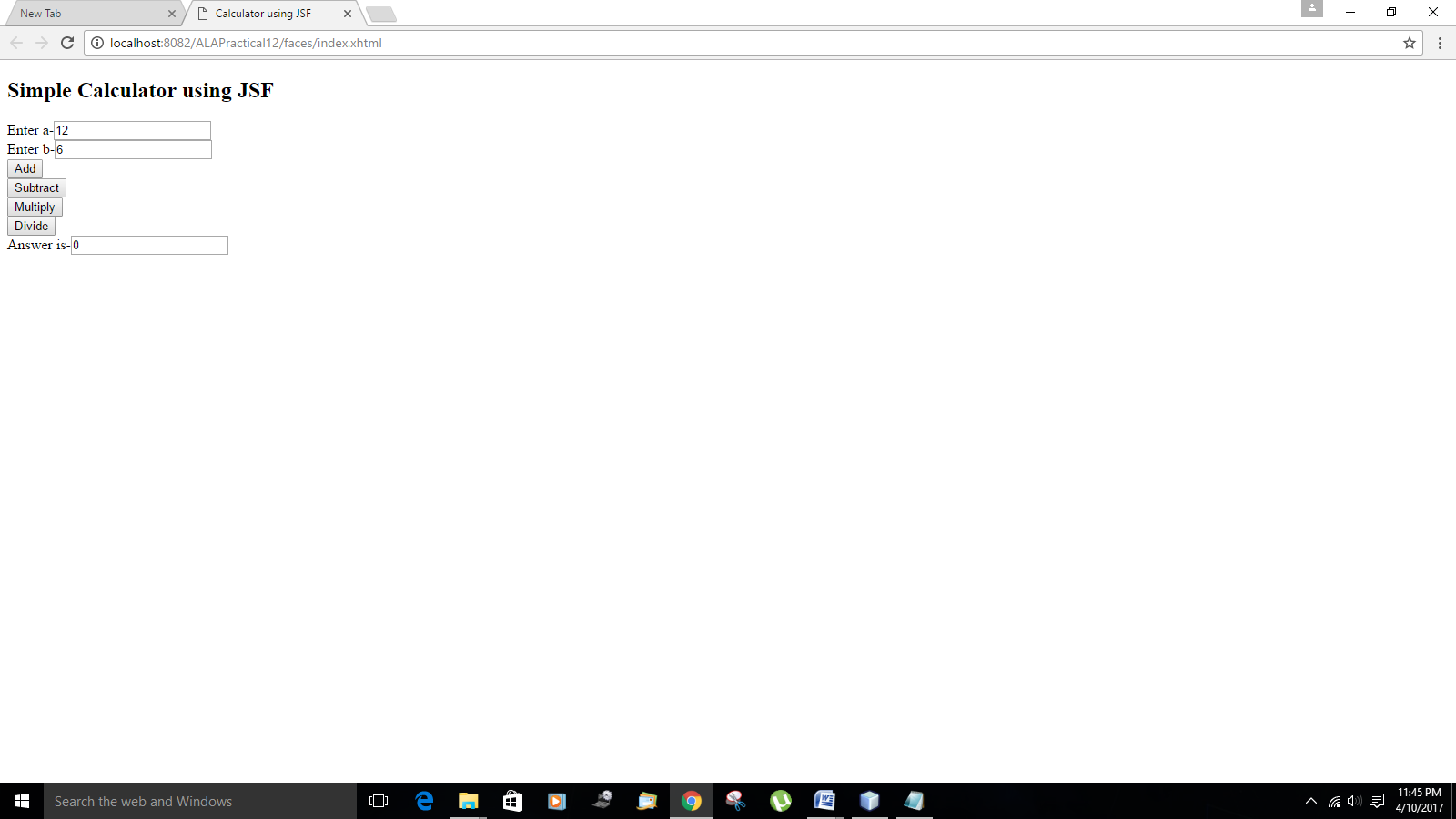
result=Integer.parseInt(a)/Integer.parseInt(b);

}

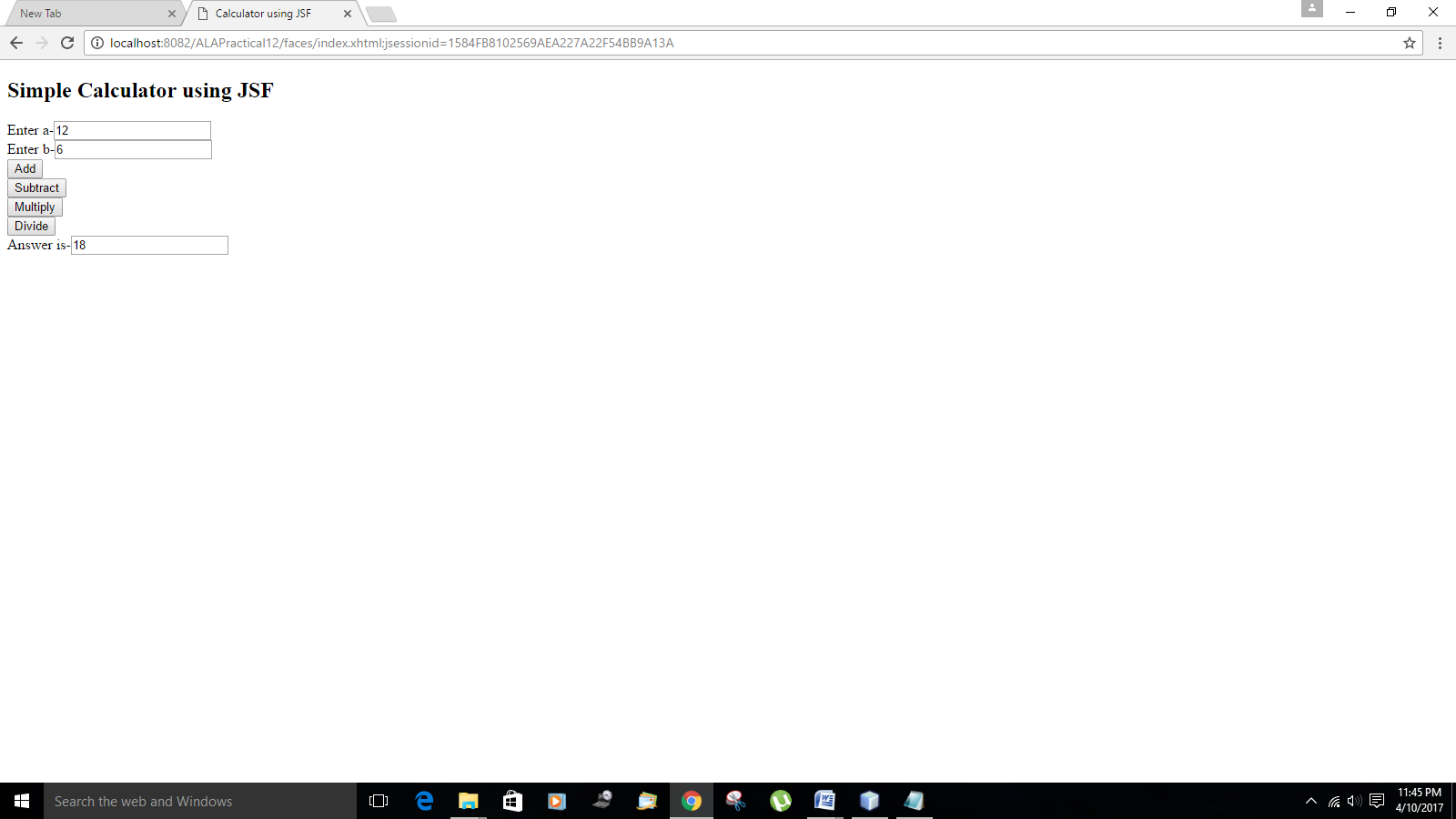
}

### ALA PRACTICAL 14 OUTPUT

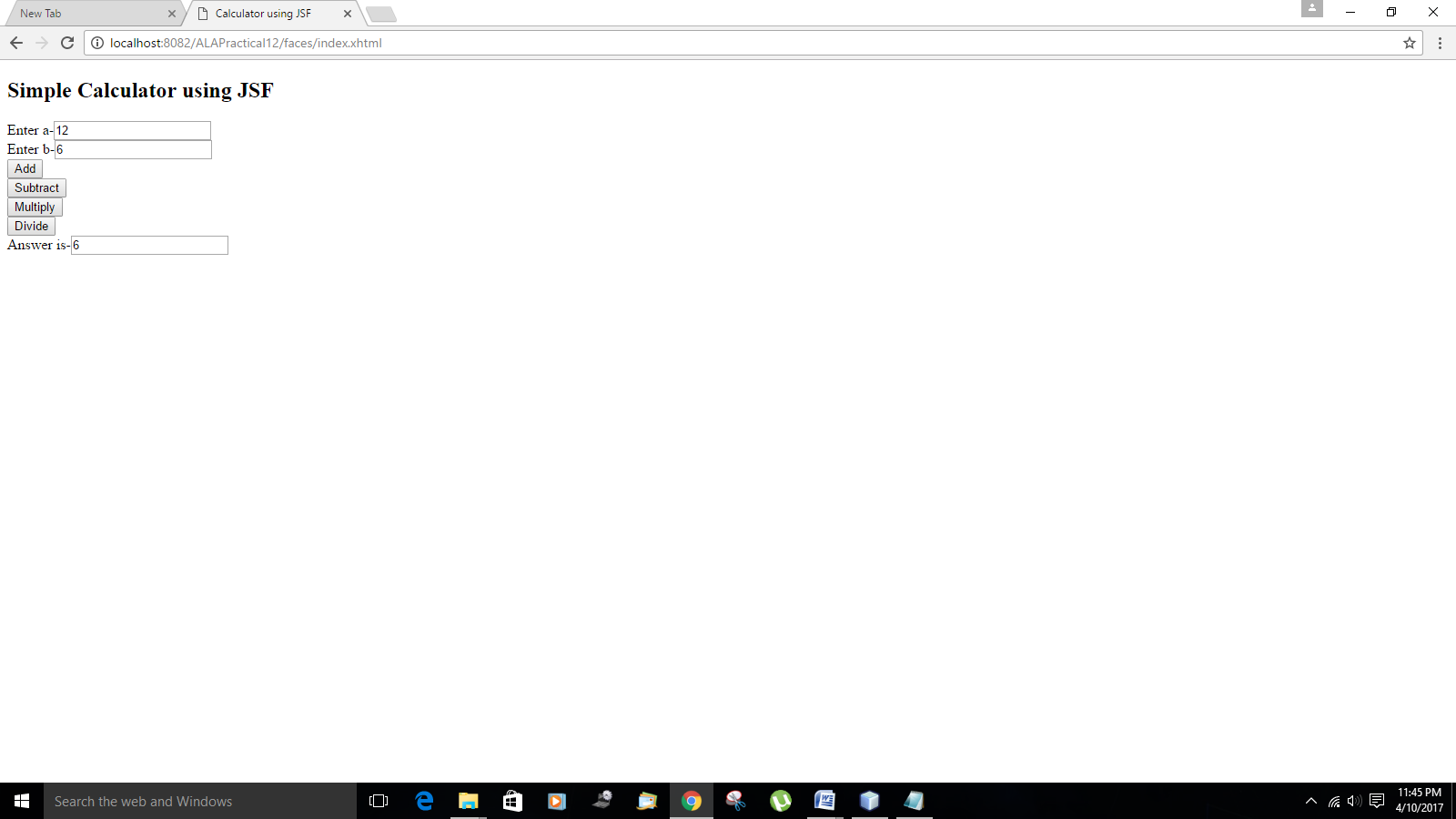
### Initially,

****

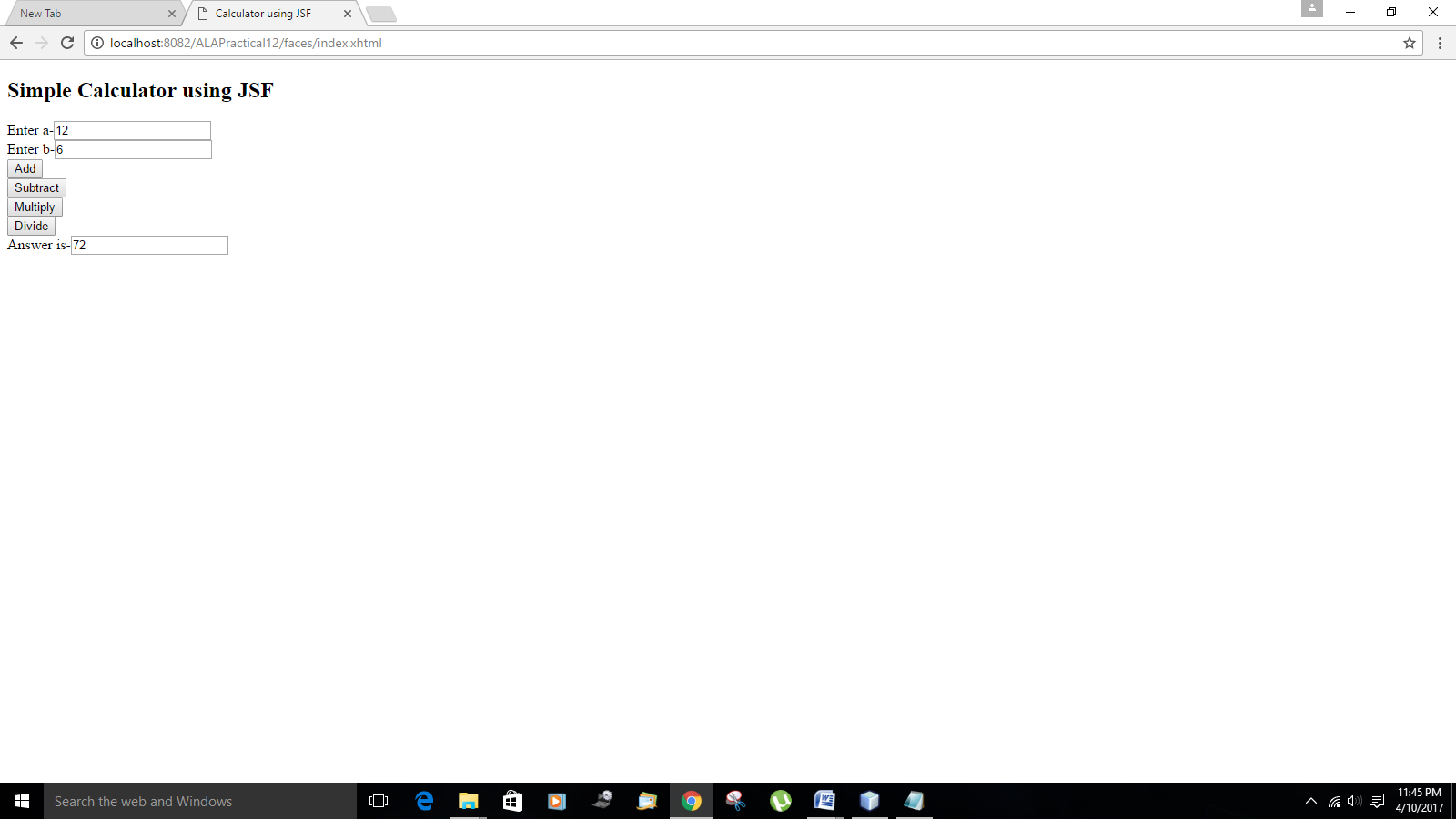
### Add,

****

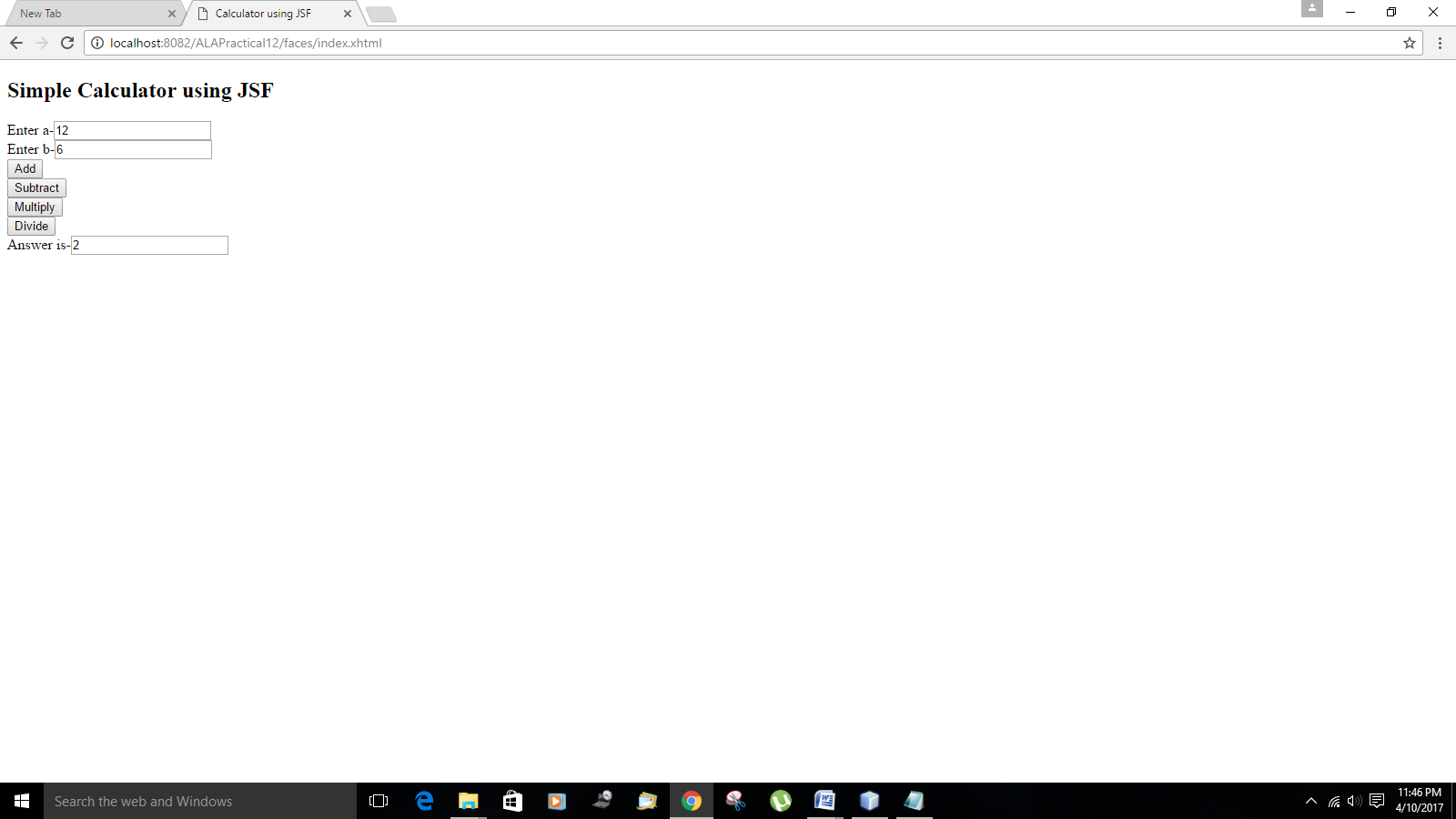
### Subtract,

****

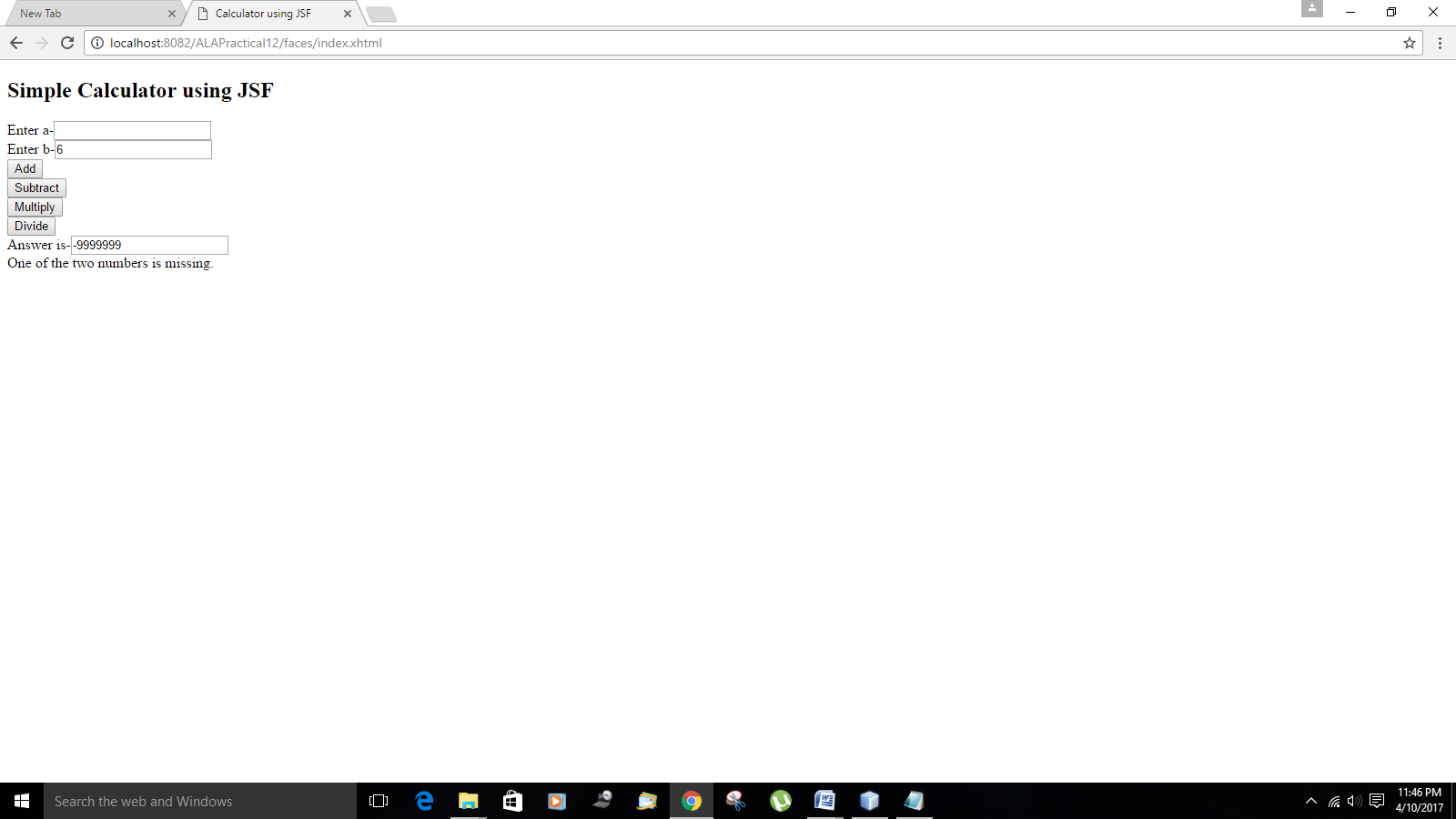
### Multiply,

****

### Divide,

****

### Number missing,

****