**Practical No.9.9 : Write jsp script of exception handling.**

**Exception.jsp**

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

<!DOCTYPE html>

<html>

<head>

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

<title>JSP Page</title>

</head>

<body>

<h1>Hello World!</h1>

<%

try{

int i=10;

int a;

a=i/0;

out.println(" "+a);

}

catch(Exception e)

{

out.println("ERROR caught "+e.getMessage());

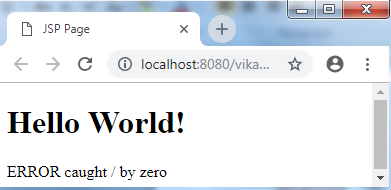
}

%>

</body>

</html>

OUTPUT



**Practical No.9.10 : Write program of cookies.**

**Cookies.html**

<html>

<head>

<title>cookie</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width">

</head>

<body>

<div>cookie</div>

<form action="page1" method="post">

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

password : <input type = "password" name ="pwd">

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

</form>

</body

</html>Bottom of Form

Bottom of Form

**Page1.servlet**

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.http.Cookie;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html;charset=UTF-8");

try (PrintWriter out = response.getWriter()) {

/\* TODO output your page here. You may use following sample code. \*/

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

out.println("welcome "+name);

Cookie ck=new Cookie("name",name);

response.addCookie(ck);

out.println("<!DOCTYPE html>");

out.println("<html>");

out.println("<head>");

out.println("<title>Servlet page1</title>");

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

out.println("<body>");

out.println("<form action='page2' method='post'>");

out.println("<input type='submit' value='submit'>");

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

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

out.println("</html>"); }

}

}

**Page2.servlet**

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.http.Cookie;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html;charset=UTF-8");

try (PrintWriter out = response.getWriter()) {

/\* TODO output your page here. You may use following sample code. \*/

Cookie ck[]=request.getCookies();

out.println("Get Cookies "+ck[0].getValue());

out.println("<!DOCTYPE html>");

out.println("<html>");

out.println("<head>");

out.println("<title>Servlet page1</title>");

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

out.println("<body>");

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

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

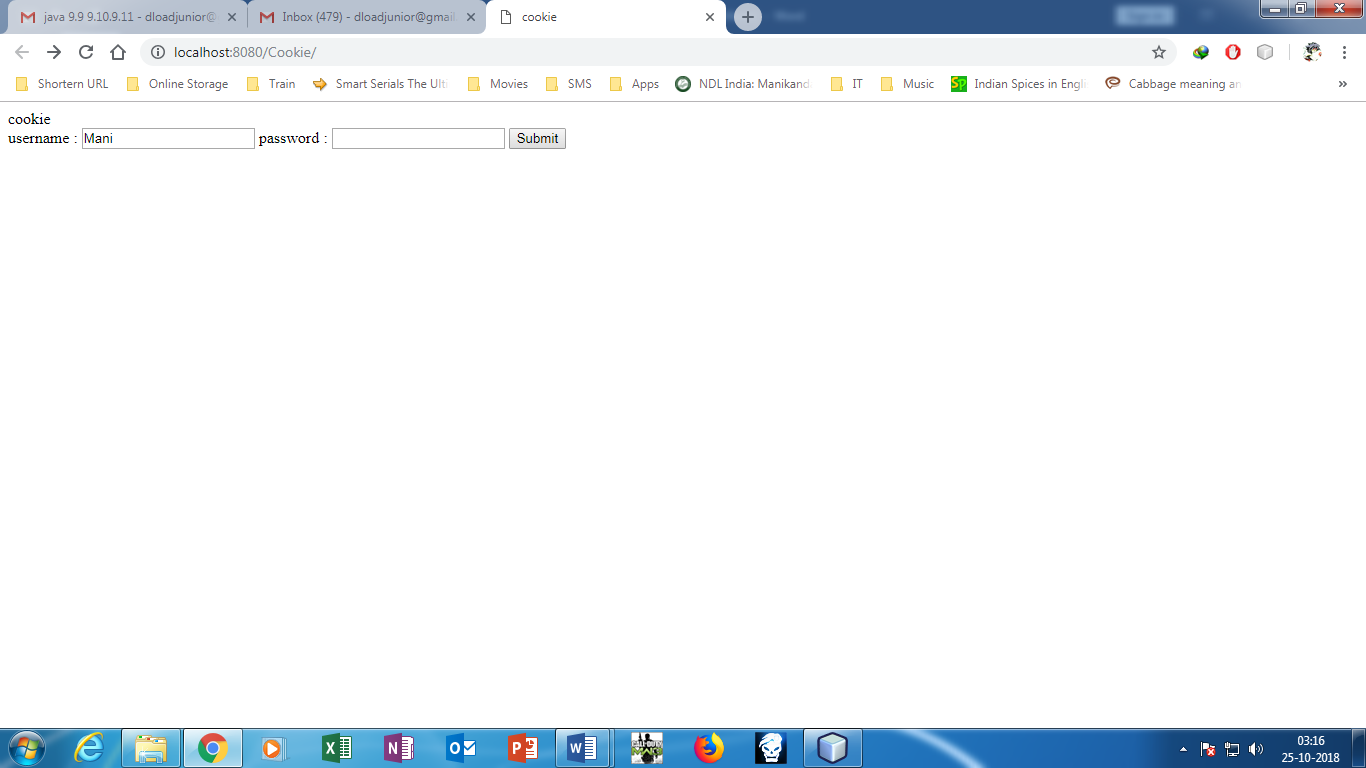
out.println("</html>");

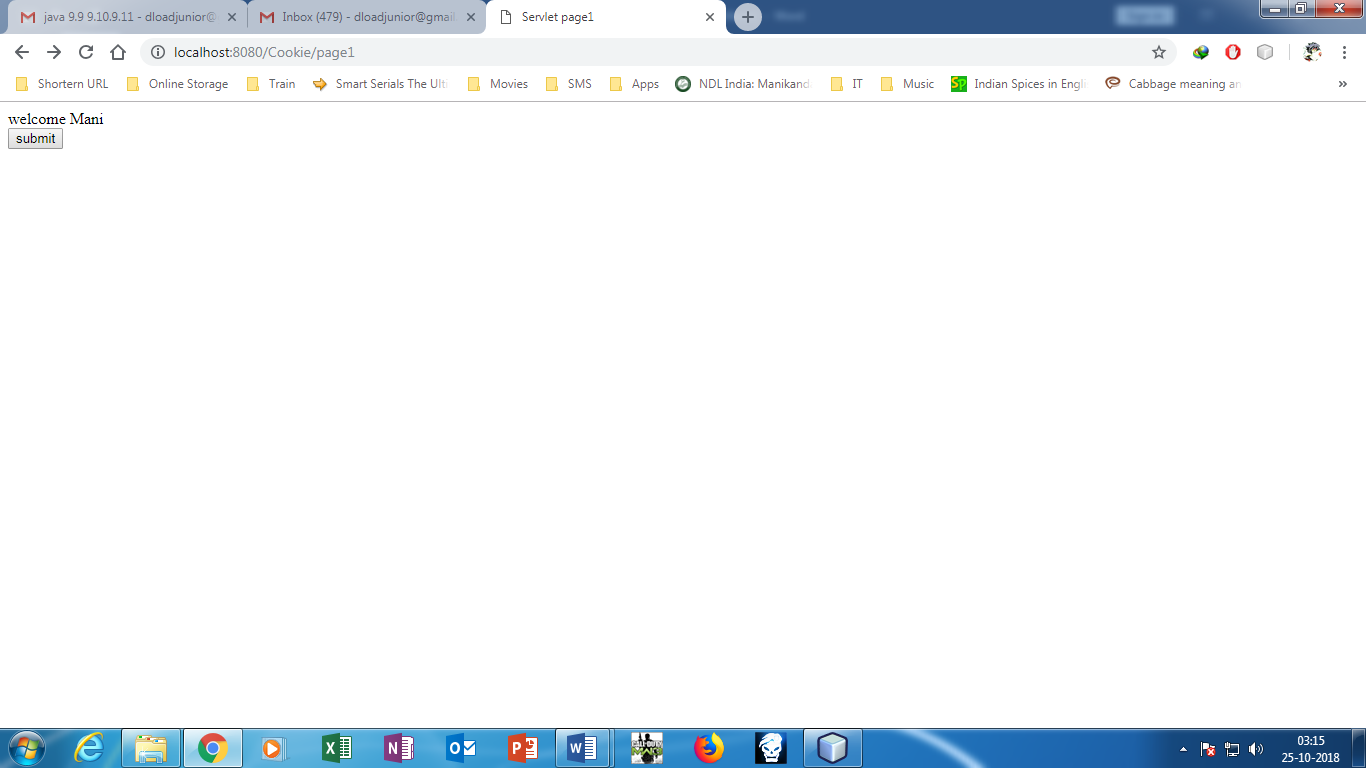
}

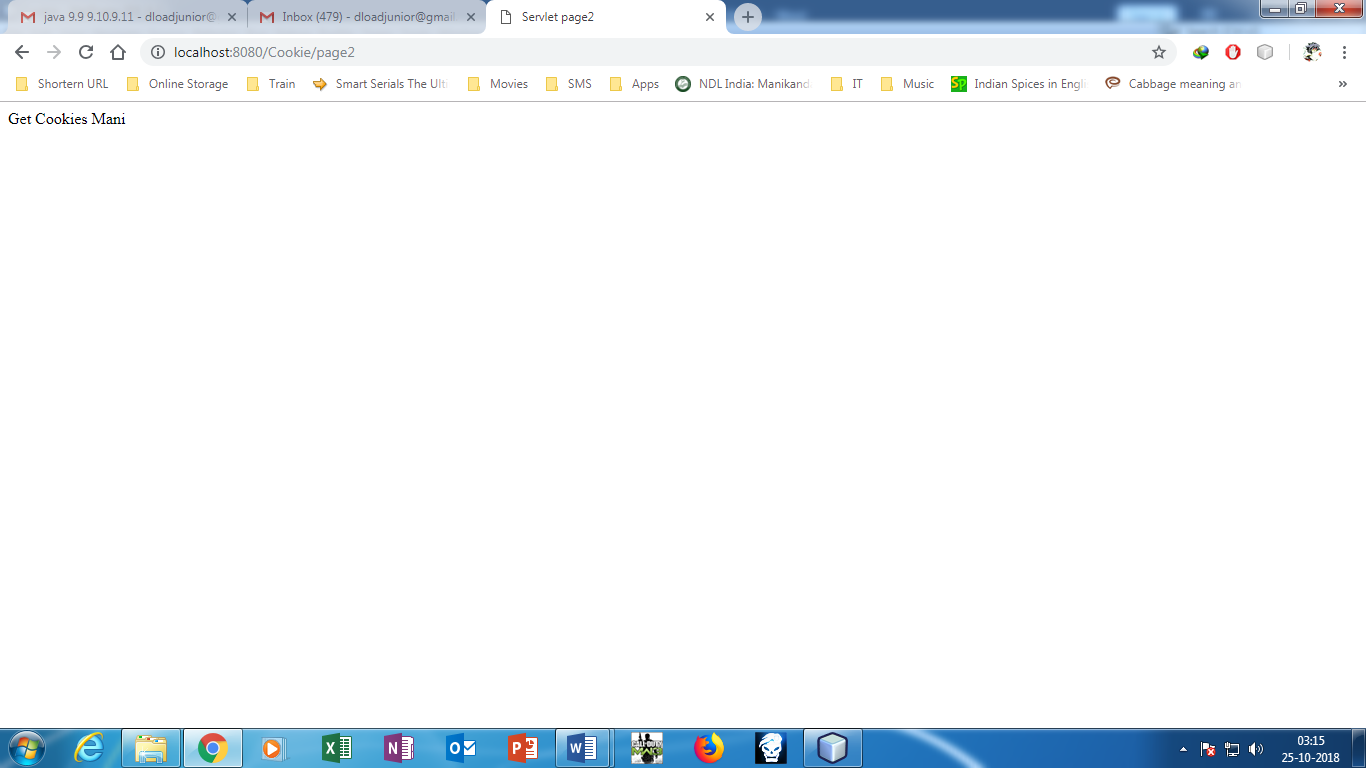
}

}

**OUTPUT :**







**Practical 9.11 : write a program of spring framework.**

**Newjsp.jsp**

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

<!DOCTYPE html>

<html>

<head>

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

<title>JSP Page</title>

</head>

<body>

<h1>SPRING FRAMEWORK</h1>

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

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

</form>

</body>

</html>

**Redirect.jsp**

<% response.sendRedirect("index.htm"); %>

**Index.jsp**

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

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"

"http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

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

<title>Welcome to Spring Web MVC project</title>

</head>

<body>

The Model-View-Controller (MVC) is an architectural pattern that separates an application into three main logical components: the model, the view, and the controller. Each of these components are built to handle specific development aspects of an application. MVC is one of the most frequently used industry-standard web development framework to create scalable and extensible projects.

MVC Components

Following are the components of MVC −

Model View Controller

Model

The Model component corresponds to all the data-related logic that the user works with. This can represent either the data that is being transferred between the View and Controller components or any other business logic-related data. For example, a Customer object will retrieve the customer information from the database, manipulate it and update it data back to the database or use it to render data.

View

The View component is used for all the UI logic of the application. For example, the Customer view will include all the UI components such as text boxes, dropdowns, etc. that the final user interacts with.

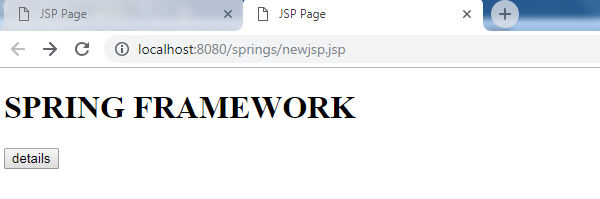
Controller

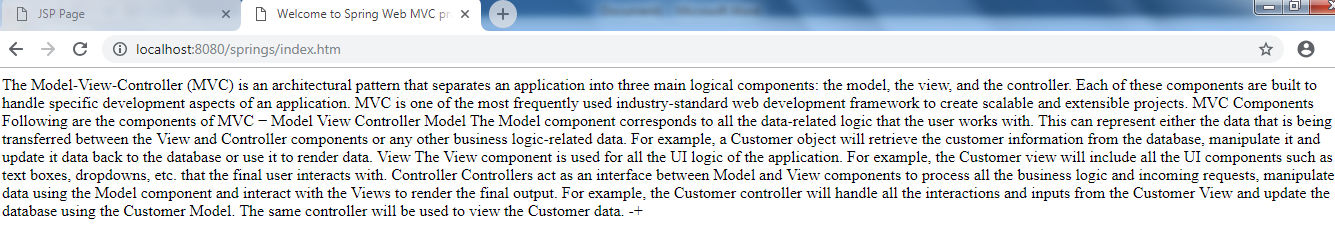
Controllers act as an interface between Model and View components to process all the business logic and incoming requests, manipulate data using the Model component and interact with the Views to render the final output. For example, the Customer controller will handle all the interactions and inputs from the Customer View and update the database using the Customer Model. The same controller will be used to view the Customer data.

</body>-+

</html>

**Output :**

****

****