

# Assignment No. 1 Div - B

Page No.	1
Date	

## Title :- TOMCAT SERVER installation

Objectives :- Understand how to install tomcat server and set up environment variables.

Problem Statement :-

Installation of TOMCAT server and configuration of it.

Software & Hardware Requirements :-

1. Java 7 or later
2. Apache Tomcat Server.

Theory :-

Web application :

A web application runs over the internet. Ex. eBay, Amazon, Google etc.

A webapp contains five components:

1. HTTP Server: Examples are - Google web server, Apache HTTP Server, Apache Tomcat Server, Microsoft Internet Information Server (IIS) etc.

2. HTTP client (Web Browser) :

Examples are : Internet Explorer, Firefox, Google chrome, Safari, etc.

3. Database : Examples are - MySQL, Apache Derby, MySQL Server, SQLite, PostgreSQL, commercial oracle, IBM DB2, SAP Sybase, MS Access etc.

4. Client Web Server :- It can be written in HTML form, VBScript, Javascript, etc.

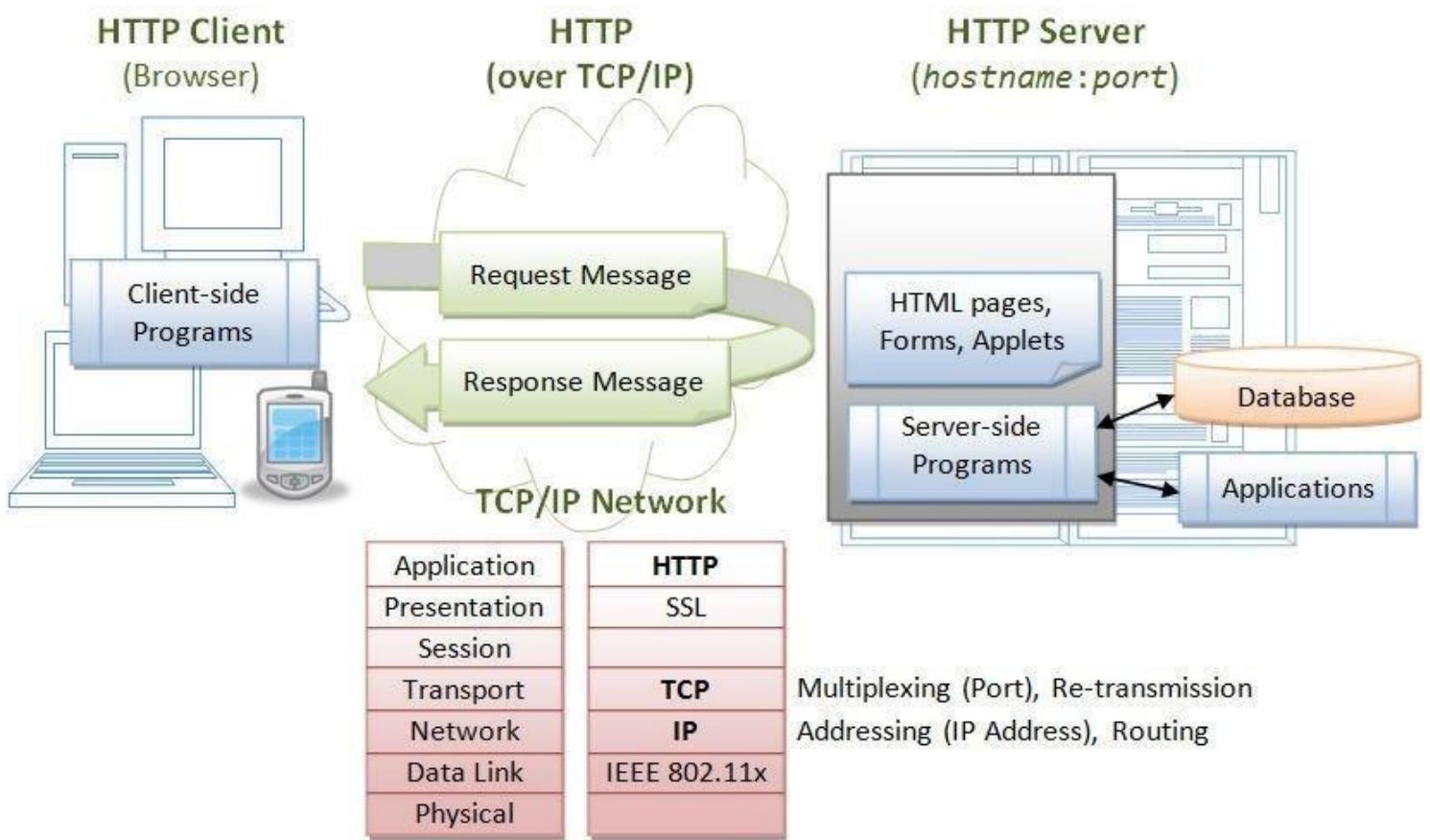
5. Server Side Programs :- could be written in Java, Servlet/JSP, ASP, PHP, Perl, Python, CGI and others.

A web app is 3 tier client-server database application which run over the Internet.

1. To start a webapp, A user, issues a URL request via a web browser (HTTP client) to HTTP server.
2. The HTTP server returns an HTML form (client side program), which is loaded into the client's browser.
3. The user fills up the query data inside the form and submits that form.
4. The client-side program sends the query parameters to a server-side program.
5. The client side program displays the query result on the browser.
6. The process repeats for the next request.

Apache tomcat :

Tomcat is an open source project, under the "Apache Software Foundation" which also provides the most user,



open source, industrial-strength Apache HTTP server). The main site for Tomcat is <http://tomcat.apache.org>.  
is <http://tomcat.apache.org>.

Alternatively you can find Tomcat via the Apache main site <http://www.apache.org>.

### Conclusion :-

Hence, we have learned how to install and configure Tomcat Server.

# Assignment No. 2

Name - Bhushan Nikumbhe  
Roll No. 22  
Div - B

Page No.	1
Date	

Title : HTML, CSS, XML

- Objectives :-
1. Understand about basic concepts of html
  2. Understand the basic concept of XML
  3. Understand the basic concept of CSS

Problem Statement :- Write a program to design registration form for students by using HTML and CSS.

Software : Notepad, Browser

Theory :-

HTML :- HTML is a standard markup language for creating Web pages.

- HTML Stands for Hyper Text Markup language
- HTML describes the structure of Web pages using markup
- HTML elements are the building blocks of HTML pages.
- HTML elements are represented by tags
- HTML tags label piece of content such as "heading", "paragraph", "table", and so on
- Browsers do not displays the HTML tags but use them to render the content of the page.

CSS:

CSS stands for cascading style sheet. It is nothing but design language intended to simplify the process of making web pages presentable. CSS handles the feel and look part of web page. By using CSS, one can control the color of text, style of fonts, spacing between paragraphs, layout design.

Advantages of CSS:-

It saves the time, Pages load faster, Easy maintenance, Superior styles to HTML, Multiple device compatibility, Global web standards, offline Browsing, Platform independent.

CSS3 Modules

CSS3 Modules are having old CSS specification as well as extension features.

- Box model
- Selectors
- Background
- Border
- Image Values and Replacement Content
- Text Effects
- Animations
- 2D / 3D Transformation
- Multiple column Layout
- User Interface

CSS can be added to HTML elements in 3 ways:

- **Inline** - by using the style attribute in HTML elements. An inline CSS is used to apply a unique style to a single HTML element.

Ex. `<h1 style="color:blue;">This is a blue Heading</h1>`

- **Internal** - by using a `<style>` element in the `<head>` section. An internal CSS is used to define a style for a single HTML page. An inline CSS is defined in the `<head>` section of an HTML page, within a `<style>` element.

Example: `<style>`

```
body {background-color: powderblue;}
h1 {color: blue;}
p {color: red;}
</style>
```

- **External** - by using an external CSS file. An external style sheet is used to define the style for many HTML pages. With an external style sheet, you can change the look of an entire website, by changing one file! To use an external style sheet, add a link to it in the `<head>` section of an HTML page.

Example:

```
<link rel="stylesheet" href="styles.css">
```

- Use the HTML <head> element to store <style> and <link> elements.
- Use the CSS color property for text color.
- Use the CSS font-family property for text font.
- Use the CSS font-size property for text size.
- Use the CSS border property for borders.
- Use the CSS padding property for space inside the border.
- Use the CSS margin property for space outside the border.

Following steps are used to create and execute web applications,

1. Write the HTML code in notepad and save it with .html extension.
2. Write the CSS code in notepad and save with .css extension.
3. Import CSS file in HTML page.
4. Open HTML page in the browser.

Conclusion :-

Hence, we have designed static web pages using HTML and CSS.

## Registration.html

```
<!DOCTYPE html>
<html lang="en" dir="ltr">

<head>
    <meta charset="utf-8">
    <title>regform</title>
    <link href="/p2/samp.css" rel="stylesheet">
</link>
</head>

<body>
    <div class="con">
        <form action="/p2/display.html" method="get" align="center">
            <div class="con2">
                </img>
            </div>
            <div class="main">
                <table align="center">
                    <tr>
                        <td><b><br>Welcome to Registration
                            Page<br><br></b></td>
                    </tr>
                </table>
                <div class="con1">
                    <table align="center">
                        <tr>
                            <td>Full Name:</td>
                            <td><input type="text" value=""></input></td>
                        </tr>
                        <tr>
                            <td>Class</td>
                            <td><input type="text" value=""></input></td>
                        </tr>
                        <tr>
                            <td>Department</td>
                            <td><input type="text" value=""></input></td>
                        </tr>
                        <tr>
                            <td>address:</td>
                            <td><textarea rows="3" columns="2"></textarea></td>
                        </tr>
                        <tr>
                            <td>email:</td>
                            <td><input type="text" value=""></input></td>
                        </tr>
                        <tr>
                            <td>phone:</td>
                            <td><input type="text" value=""></input></td>
                        </tr>
                    </table>
                </div>
            </div>
        </form>
    </div>
</body>
```

```

        </tr>
        <tr></tr>
        <tr></tr>
    </table>
    <table align="center">
        <tr>
            <td><input type="submit" value="SUBMIT"></td>
            <td><input type="reset" value="CANCEL"></td>
        </tr>
    </table>
</div>
</div>
</form>
</div>
</body>

</html>

```

## Samp.css

```

.con {
    width: 400px;
    height: 400px;
    background-color: blue;
    color: black;
    padding: 0px 10px 0px 30px;
    margin: auto;
}

.main {
    width: 400px;
    height: 80px;
    float: left;
    padding: 60px 10px 10px 10px;
    margin: auto;
}

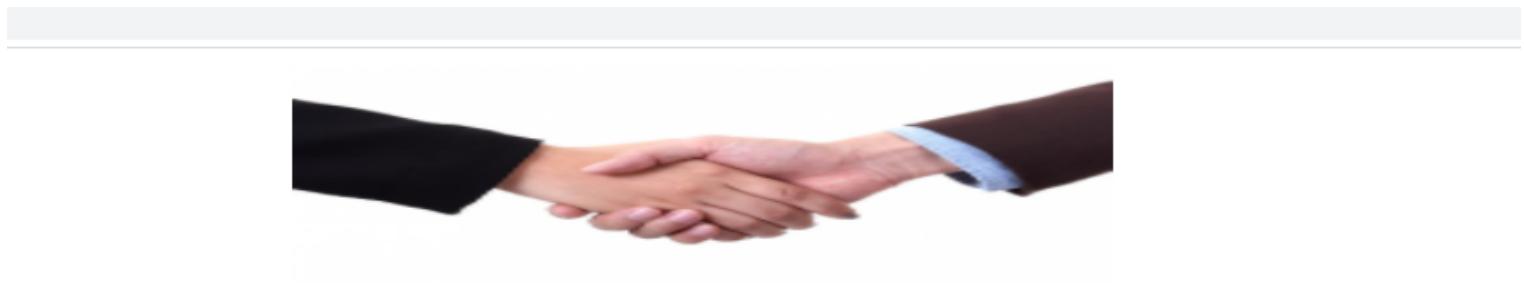
.con1 {
    width: 200px;
    height: 30px;
    float: left;
    margin: auto;
    padding: 0px 0px 0px 0px;
}

.con2 {
    width: 150px;
    height: 30px;
    float: left;
    margin: 10px 10px 10px 10px;
    padding: 0px 0px 0px 0px;
}

```

```
<html>
  <head>
    <title> display</title>
  </head>
  <body>
    <h1 style="color:blue; margin-left:30px;"> You have submitted the form successfully....</h1>
  </body>
</html>
```

Output-



Welcome to Registration Page

Full Name:

Class

Department

address:

email:

phone:

# Assignment No. 3

Name - Bhushan Nikumbhe  
Roll No. 22  
Div - B  
Page No. 1  
Date

Title : XML and CSS

Problem statement :- Write a program to design book catalog by using XML and CSS to display title, author, price and year of the book.

Software : Notepad, Any browser

Theory :-

XML stands for Extensible Markup Language. It is nothing but the text-based markup language which is derived from Standard Generalized Markup Language (SGML).

XML tags identify data and are used to store and organize the data, rather than specifying how to display it like HTML tags, which are used to display the data.

There are three important characteristics of XML that makes it useful in a variety of systems

- XML is extensible - XML allows you to create your own self-descriptive tags, or language, that suits your application.
- XML carries the data, does not present it - XML allows you to store the data irrespective of how it will be presented.

• XML is a public standard - XML was developed by an organization called World Wide Web Consortium (W3C) and is available as an open standard.

Technology / Tool :-

The XML document have an XML declaration, but it is optional, and it is written as -

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

Where version is nothing but the version of an XML document and UTF specifies the character encoding used in the document.

An XML document can have only one root element.

```
<root>
  <x> ... </x>
  <y> ... </y>
</root>
```

XML Attributes :-

Using a name/value pair, an attribute specifies a single property of element. An XML element can have one or more attributes.

For example:

```
<a href="http://www.google.com">  
    XML tutorial</a>
```

Conclusion :-

Hence, we have designed Static web pages using XML and CSS.

## Book.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/css" href="book_catalog.css"?>
<CATALOG>
  <BOOK>
    <TITLE>Database Management System</TITLE>
    <AUTHOR>Korth</AUTHOR>
    <PRICE>500</PRICE>
    <YEAR>1985</YEAR>
  </BOOK>
  <BOOK>
    <TITLE>Computer Network</TITLE>
    <AUTHOR>Tenenbaum</AUTHOR>
    <PRICE>600</PRICE>
    <YEAR>1985</YEAR>
  </BOOK>
  <BOOK>
    <TITLE>Software Engineering and project Management</TITLE>
    <AUTHOR>Roger Pressman</AUTHOR>
    <PRICE>600</PRICE>
    <YEAR>1985</YEAR>
  </BOOK>
</CATALOG>
```

## Book\_catalog.css

```
BOOK {
  display: block;
  margin-left: 0;
  margin-bottom: 30pt;
}

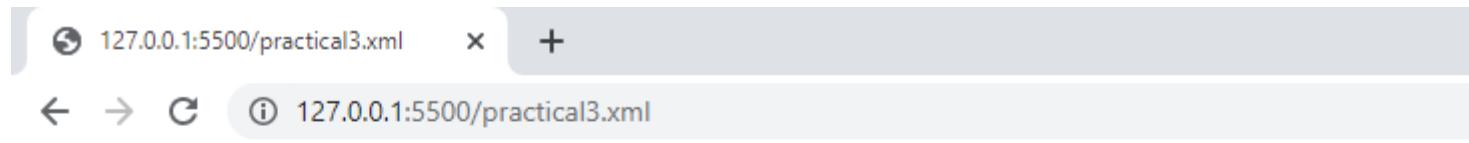
CATALOG {
  width: 100%;
  background-color: #ffffff;
}

TITLE {
  color: ff0000;
  display: block;
  font-size: 20pt;
}

AUTHOR {
  display: block;
  color: #0000ff;
  font-size: 20pt;
```

```
}  
  
YEAR, PRICE {  
    Color: #000000;  
    Display: block;  
    Margin-Left: 20pt;  
}  
  
}
```

Output-



Database Management System

Korth

500  
1985

Computer Network

Tenenbaum

600  
1985

Software Engineering and project Management

Roger Pressman

600  
1985

Page No.	1
Date	

# Assignment - 4

Title :- HTML JavaScript

Objective :-

1. Understand about basic concepts of Javascript.
2. Use Javascript for validation of data.

Problem Statement:-

Write a program to design registration form for students by using HTML, CSS & Javascript and perform following validations: all fields mandatory, phone number and email address validation.

Software requirements:-

Notepad, Any Browser

Theory :-

Javascript is a programming language of HTML as well as web. It is preferred for creating network-centric applications. It is integrated and complimentary with Java. As Javascript is integrated with HTML is a very easy to implement. It is open as well as cross-platform.

## Advantages of Javascript:-

- It requires less server interaction.
- Immediate feedback to the visitor.
- Increased interactivity.
- Richer interfaces.

## Validation :

When client enters all necessary data and press the Submit button form validation is done at server side.  
If data entered by client is incorrect or missing, the server needs to send all data back to the client and request for resubmission of form with correct information.

- Basic validation : First of all the form must be checked to make sure all the mandatory fields are filled in. It would require just a loop through each field in the form & check for the data.
- Data Format validation : - Secondly the data that is entered must be checked for correct format & its value. The code must include appropriate logic to test correctness of data.

Technology / Tool :-

Javascript can be implemented using Javascript statements that are placed within the `<script>`

You can place the `<script>` tag, containing your Javascript, anywhere within your web page but it is normally recommended that you should keep it within the `<head>` tag.

The script takes two important attributes:

- Language - This attribute specifies what scripting language you are using. Typically, its value will be javascript.
- Type : This attribute is what is now recommended to indicate the scripting language in use and its value should be set to "text/javascript".

Conclusion :-

Hence, we applied validate the data using Javascript.

## Registration.html

```
<html>

<head>
<title>regform</title>
<link href="samp.css" rel="stylesheet"></link>
<script type="text/javascript">
function Validation() {
    varemailID =document.getElementById('EMail').value;
    varnum = document.getElementById('ph').value;
    atpos = varemailID.indexOf("@");
    dotpos = varnum.lastIndexOf(".");
    if (atpos < 1 || (dotpos - atpos < 2)) {
        alert("Please enter correct email ID")
        document.myForm.Email.focus();
        return false;
    } else if (isNaN(varnum)) {
        document.getElementById("ph");
        alert("Please enter numeric only.")
        document.myForm.ph.focus();
        return false;
    } else {
        return true;
    }
}
</script>
</head>

<body>
<div class="con">
<form name="myForm" action="display.xml" method="get" align="center">
<div class="con2">
    </img>
</div>
<div class="main">
    <table align="center">
        <tr>
            <td><br><br>Welcome to Registration Page<br><br></td>
        </tr>
    </table>
<div class="con1">
    <table align="center">
        <tr>
            <td>Full Name:</td>
            <td><input type="text" id="FirstName" value="" required></input></td>
        </tr>
        <tr>
            <td>Class:</td>
            <td><input type="text" id="Class" value="" required></input></td>
        </tr>
        <tr>
            <td>Department:</td>
```

```

<td><input type="text" id="Department" value="" required></input></td>
</tr>
<tr>
    <td>address:</td>
    <td><textarea rows="3" columns="2" value="" required></textarea></td>
</tr>
<tr>
    <td>email Id:</td>
    <td><input type="text" id="EMail" value="" required></input></td>
</tr>
<tr>
    <td>phone no:</td>
    <td><input type="text" id="ph" value="" required></input></td>
</tr>
<tr></tr>
<tr></tr>
</table>
<table align="center">
    <td><input type="submit" value="SUBMIT" onclick="return(Validation());"></td>
    <td><input type="reset" value="CANCEL"></td>

    </table>
</div>
</div>
</form>
</div>
</body>

</html>

```

## Samp.css

```

.con {
    width: 400px;
    height: 400px;
    background-color: blue;
    color: black;
    padding: 0px 10px 0px 30px;
    margin: auto;
}

.main {
    width: 400px;
    height: 80px;
    float: left;
    padding: 60px 10px 10px 10px;
    margin: auto;
}

.con1 {
    width: 200px;
    height: 30px;
    float: left;
    margin: auto;
    padding: 0px 0px 0px 0px;
}

```

```
.con2 {  
    width: 150px;  
    height: 30px;  
    float: left;  
    margin: 10px 10px 10px 10px;  
    padding: 0px 0px 0px 0px;  
}
```

## Output-

← → ⌂ ⓘ 127.0.0.1:5500/registration.html

Welcome to Registration Page

Full Name: Bhushan

Class:

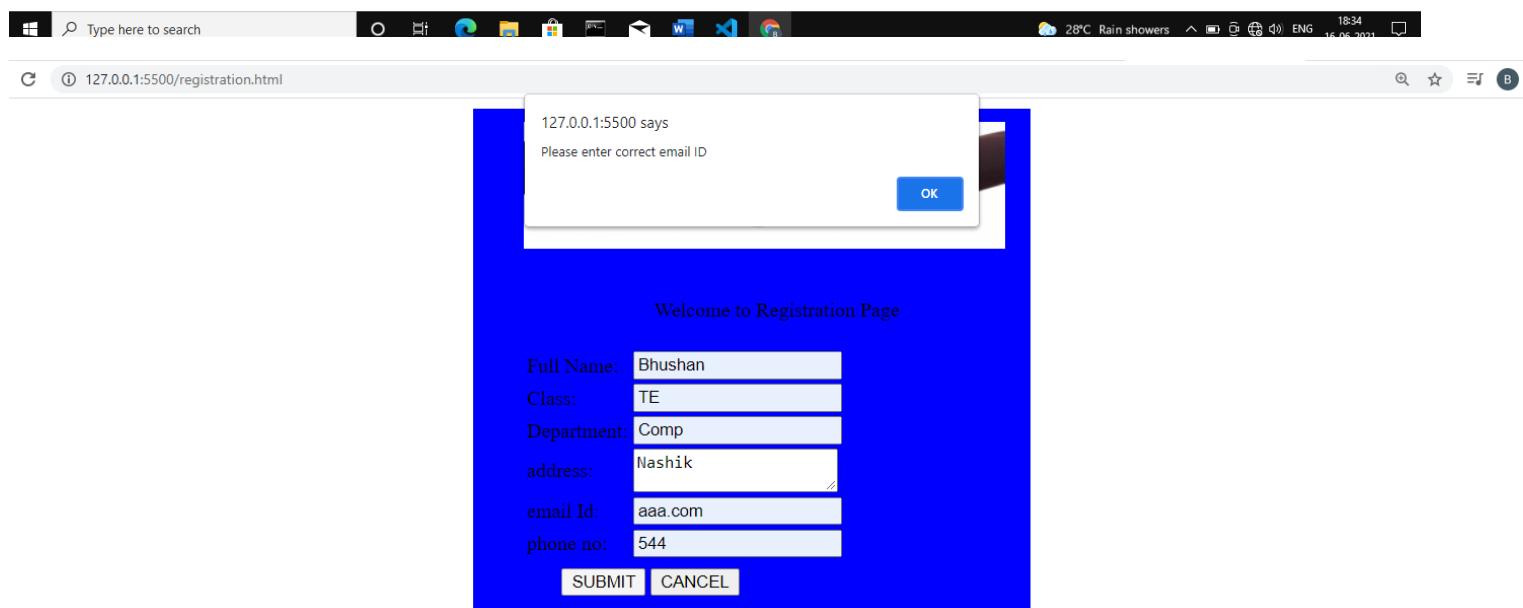
Department:

address:

email Id: aaa.com

phone no: 544

SUBMIT CANCEL



127.0.0.1:5500 says  
Please enter numeric only.

OK

### Welcome to Registration Page

Full Name:	Bhushan
Class:	TE
Department:	Comp
address:	Nashik
email Id:	Bhu@gmail.com
phone no:	aaa

SUBMIT CANCEL

## Assignment - 5

Title - JSP, Servlet and MySQL (Backend)

Problem Statements :-

1. Design and build Student Login Page using JSP, Servlet and MySQL
2. Design and build Employee Login Page using JSP, Servlet and MySQL.

Software Requirements:-

1. Any operating System.
2. JDK 7 or later
3. Editors: Netbeans / Eclipse
4. Web browser
5. Tomcat 7 or later.

Theory -

~~Java Server Pages (JSP)~~ :-

It is a server side programming technology that is used to create dynamic web-based applications. JSP have right to use the complete Java APIs, including the JDBC API to access the databases.

It is a technology that helps software developer to create dynamic web pages based on HTML, XML and other document types.

JSP tags can be used for different purposes, such as retrieving information from a database or registering user.

preferences, accessing JavaBeans components, passing control between pages and sharing information between requests, pages, etc.

### Why we need JSP?

JSP is used for the design of dynamic web page and Servlet is used to code the logic that is present i.e. in the MVC (Model view controller) architecture the servlet is the controller and the JSP is the view.

### Architecture of JSP :

1. The request/response part of a JSP is defined in below architecture
2. The client initiated request for a JSP file using browser.
3. web server invokes the JSP file and interpret the JSP file produce a Java code. The created Java code will be servlet.
4. Once servlet is created, JSP engine compiles the servlet, compilation errors will be detected in this phase.
5. Now Servlet class is loaded by the container and executes it.
6. Engine sends the response back to the client.

## Syntax of JSP:

JSP declarations is used to declare variable and methods, as shown below.

<% text %>

Following is the simple & first example for JSP

```
// Hello.jsp
```

```
<html>
```

```
<head>
```

```
<title> JSP File </title>
```

```
</head>
```

```
<body>
```

```
<%
```

```
out.println("welcome to JSP class");
```

```
%>
```

```
</body>
```

```
</html>
```

## Output:-

welcome to JSP class

## Servlet :-

A Servlet is a server side program and written in Java. Servlet is a web component that is deployed on the server for creating the dynamic web pages. A Java Servlet is a program that extends the capabilities of a server.

Conclusion :-

Hence, we have performed the dynamic web application using JSP, Servlet and MySQL.

## Login.jsp(Login Page)

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1" %>
<!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=ISO-88591">
<title>Insert title here</title>
</head>
<body>
Simple login Example using servlet jsp and mysql(mariadb) database connectivity
<br> Create a test database, student table and insert some user information in it.
<br>
<br>
<form action="LoginController" method="post">
Enter username :<input type="text" name="username"> <br>
Enter password :<input type="password" name="password"><br>
<input type="submit" value="Login">
</form>
</body>
</html>
```

## LoginController.java (Login Controller)

```
package com.candid;
import java.io.IOException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
@WebServlet("/LoginController")
public class LoginController extends HttpServlet {
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
String un = request.getParameter("username");
String pw = request.getParameter("password");
// Connect to mysql(mariadb) and verify username password
try {
Class.forName("org.mariadb.jdbc.Driver");
// loads driver
Connection c = DriverManager.getConnection("jdbc:mariadb://localhost:3306/test", "root", "root");
// gets a new connection
PreparedStatement ps = c.prepareStatement("select userName,pass from student where userName=? and
pass=?"); ps.setString(1, un); ps.setString(2, pw);
ResultSet rs = ps.executeQuery();
while (rs.next()) {
```

```
response.sendRedirect("success.html"); return; }
response.sendRedirect("error.html"); return;
} catch (ClassNotFoundException | SQLException
e) { e.printStackTrace();
}
}
}
```

Success.html (Success page)

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
Login success
</body>
</html>
```

Error.html (error page)

```
<!DOCTYPE html>
<html> <head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
Invalid username or password, Please try again with valid
</body>
</html>
```

Output-

Insert title here - Chromium

Insert title here ×

localhost:8080/login/

Simple login Example using servlet jsp and mysql database connectivity  
Create a test database, student table and insert some user information in it.

Enter username :

Enter password :

Insert title here - Chromium

Insert title here ×

localhost:8080/login/success.html

Login success

Insert title here - Chromium

Insert title here ×

localhost:8080/login/error.html

Invalid username or password, Please try again with valid