

# Assignment No 7

Page No.	
Date	

Title

Tomcat Server Installation

Problem Statement

Installation of Tomcat Server and Configuration.

Theory:

Web Application:

A Web Application runs over the Internet  
Ex. Amazon, Google etc.

A webapp contains five component

- ① HTTP Server: Apache + HTTP server
- ② Do HTTP Client: Google Chrome
- ③ Database: MySQL
- ④ Client Side program: HTML, JS
- ⑤ Server Side program: JSP, PHP etc.

A web application is 3 tier client server database application which run over the Internet as shown in following diagram.

Apache Tomcat:

Tomcat is an open source project under the Apache Software foundation, the mother site for Tomcat is <http://tomcat.apache.org>.

# Tomcat Directories

## Tomcat Directories

- **bin** : It contains the binaries, startup script, shutdown script, as well as other binaries and scripts.
  - **conf** : It contains the system-wide configuration files such as server.xml, web.xml.
  - **lib** : It contains the tomcat system-wide JAR files which can be accessible by all webapps.
  - **log** : It contains the log files of tomcat.
  - **webapps** : It contains the webapps to be deployed.
  - **work** : It is Tomcat's working directory used by JSP, for JSP-to-servlet conversion.
  - **temp** : Temporary files are stored.
- Conclusion:** Hence, we have learned how to install and configure tomcat server.

# Assignment No - 02

Page No.	
Date	

Title : HTML , CSS , XML

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

Theory

HTML: HTML is the standard markup language for creating web pages.

HTML stands for HyperText markup language.

HTML describes the structure of web pages using markup.

HTML elements are the building blocks of HTML pages.

HTML Represented by tags

CSS:

CSS stand for Cascading style sheet. It is nothing but design language intended to simplify the process of making web pages presentable.

Advantages of CSS.

- It save the time; Pages load faster
- Easy maintenance, Superior style to HTML
- Multiple Device Compatibility, Global web standards.

## CSS Modules

- Box Model
- Selectors
- Background
- Border
- Image valued & Replaced content
- Text effects
- Animations
- 2D/3D Transformation
- Multiple Column Layout
- User Interface

CSS can be added to HTML in three ways

- Inlined - by using `<style>` attribute in HTML element
- Internal - by using `<style>` element in the `<head>` section,
- External - by using external.css file

Conclusion :

- Here we have designed static webpages using HTML and CSS.

**NAME: UJJWAL ARVIND LADE**

**Roll No: 06**

**Div: TE-B**

**ASSIGNMENT NO : 02**

Write a program to design a registration form for students by using html and css.

```
<!-------FILE ! (.html)----->

<html>
  <head>
    <title>Student Registration Form Using Table in HTML</title>
    <link rel="stylesheet" href="ass2.css">
  </head>
  <body>
    <h3>Assignment No : 2
      Write a program to design a registration form for students by using html and css.</h3>
    <h2>Student Registration Form Using HTML and CSS</h2>
    <table >
      <!------- First Name ----->
      <tr>
        <td>First Name</td>
        <td><input type="text" name="FirstName" maxlength="50" />
      </td>
      </tr>
      <!------- Last Name ----->
      <tr>
        <td>Last Name</td>
        <td><input type="text" name="LastName" maxlength="50" />
      </td>
      </tr>
      <!------- Email ID ----->
      <tr>
```

```
<td>Email ID</td>
<td><input type="email" name="EmailID" maxlength="100"
/></td> </tr>
<!-- Mobile Number -->
<tr>
<td>Mobile Number</td>
<td>
<input type="text" name="MobileNumber" maxlength="10" placeholder="10 digit number"/>
</td>
</tr>
<!-- Gender -->
<tr>
<td>Gender</td>
<td>
<input type="radio" name="Gender" value="Male" />
Male
<input type="radio" name="Gender" value="Female" />
Female
</td>
</tr>
<!--Date Of Birth-->
<tr>
<td>Date of Birth(DOB)</td>
<td>
<input type="date" name="" id="">
</td>
</tr>
<!-- Address-->
<tr>
<td>Address<br /><br /><br /></td>
<td><textarea name="Address" rows="10" cols="50"></textarea></td>
```

```
</tr>
<!-- City -->
<tr>
<td>City</td><td><input type="text" name="City" maxlength="50" placeholder="Your City Name"/>
</td>
</tr>
<!-- State -->
<tr>
<td>State</td>
<td><input type="text" name="State" maxlength="50" placeholder="Maharashtra"/>
</td>
</tr>
<!-- Hobbies -->
<tr>
<td>Hobbies <br /><br /><br /></td>
<td>
<input type="checkbox" name="HobbyDrawing" value="Drawing" /> Drawing
<input type="checkbox" name="HobbySinging" value="Singing" /> Singing
<input type="checkbox" name="HobbyDancing" value="Dancing" /> Dancing
<input type="checkbox" name="HobbyCooking" value="Cooking" />
Sketching
<br />
<input type="checkbox" name="HobbyOther" value="Other"> Others
<input type="text" name="Other_Hobby" maxlength="50" placeholder="Ex-Teaching" />
</td>
</tr>
<!-- Submit and Reset -->
<tr>
<td colspan="2" align="center">
<input type="submit" value="Submit">
<input type="reset" value="Reset">

```

```
</td>  
</tr>  
</table>  
</form>  
</body>  
</html>
```

## CSS Code

```
h2{  
    font-family: Sans-serif;  
    font-size: 24px;  
    font-weight: bold;  
    color: rgb(0, 0, 0);  
    text-align: center;  
  
}  
  
table{  
    font-family: verdana;  
    color:rgb(0, 0, 0);  
    font-size: 16px;  
    font-style: normal;  
    font-weight: bold;  
    background: #ffcc99;  
    border-collapse: collapse;  
    border-radius: 20px;  
    box-shadow: 2px 2px 1px 2px black;  
}  
  
table.inner{  
    border: 10px  
}  
  
input[type=text], input[type=email], input[type=number]{
```

```

width: 90%;

padding: 6px 12px;

margin: 5px 0;

box-sizing: border-box;

}

input[type=submit], input[type=reset]{

width: 100%;

padding: 8px 12px;

margin: 5px 0;

box-sizing: border-box;

background-color: rgb(52, 93, 226);

color: #ddd;

}

input[type=submit]:hover, input[type=reset]:hover{

background-color: rgb(19, 68, 226);

}

```

Student Registration Form Using HTML and CSS

Assignment No : 2 Write a program to design a registration form for students by using html and css.

The screenshot shows a web browser displaying a student registration form. The form is styled with an orange header and a blue footer. It contains fields for personal information like name, email, and address, as well as a date picker for birthdate. There are also sections for hobbies and a note field, both with checkboxes. The 'Submit' and 'Reset' buttons are located at the bottom of the form.

# Assignment No 08

Page No. \_\_\_\_\_  
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

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

Characteristics

- 1) XML is extensible - XML allow you to ~~use~~ create your own self descriptive tags or languages that suit your application.
- 2) XML carries the data does not present it → XML allows you to store the data irrespective of how it will be presented.

## XML - Attributes:

listing or Using a name / value pair on attributes specifies a single property for an element. An XML - element can have one or more attributes. For example

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

(1a)

Here href is the attribute name and and http://www.google.com is attribute value.

Conclusion:

Hence we have designed static web pages using XML and CSS

## ASSIGNMENT NO-03

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

File !1= books.xml

```
<?xml version="1.0" encoding="UTF-8"?> <?xmlstylesheet type="text/css" href="books.css"?>
<books>

<heading>Welcome </heading>

<book>

<title>Title -: Web Programming</title>
<author>Author -: Chrisbates</author>
<year>Year -: 1997</year>
<price> Price -: 300</price>
</book>

<book>

<title>Title -: Internet world-wide-web</title>
<author>Author -: Ditel</author>
<year>Year -: 2004</year>
<price>Price -: 400</price>
</book>

<book>

<title>Title -: Computer Networks</title>
<author>Author -: Forouzan</author>
<year>Year -: 1996</year>
<price>Price -: 700</price>
</book>

<book>

<title>Title -: DBMS Concepts</title>
<author>Author -: Navath</author>
<year>Year -: 1980</year>
<price>Price -: 600</price>
</book>

<book>
```

```
<title>Title :- Linux Programming</title>
<author>Author :- Subhitab Das</author>
<year>Year :- 1995</year>
<price>Price :- 300</price>
</book>
</books>
```

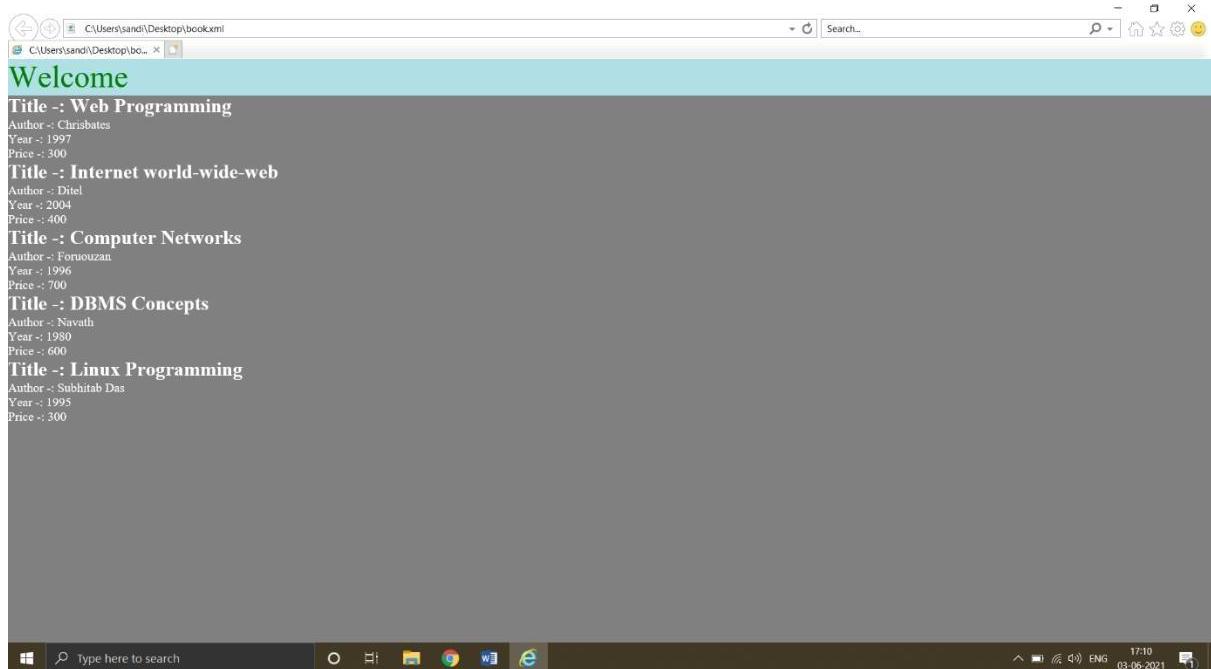
## File 2 – Books.css

```
books {
color: white;
background-color : gray;
width: 100%; }

heading {
color: green;
font-size : 40px;
background-color : powderblue;
}

heading, title, author, publisher, edition, price {
display : block;
}

title {
font-size : 25px;
font-weight : bold;
}
```



# Assignment No-04

Page No. \_\_\_\_\_  
Date. \_\_\_\_\_

Title : HTML , Java Script

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

Theory:- JavaScript is a programming language of HTML as well as web . It is preferred for creating network centric application. It is integrated and complementary with Java . As JavaScript is integrated in HTML it is very easy to implement . It is open as well as cross platform.

Advantages of using JavaScript are -

- It requires less server interaction.
- Immediate feedback to visitors
- Increase Interactivity.
- Richer interface.

## Validation

When client enters the all necessary data and press the Submit button form Validation is done at server side if data entered by a 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 field are filled in.

Data format validation - secondly the data that is entered must be checked for correct format and its value.

### - Technology / Tools :

Java Scripts can be implemented using Java Script statement that are placed within the <Script>

The Script tag takes two important attributes:

Language - this attribute specifies what scripting language you are using.

Type - this attribute is what is now recommended to indicate the scripting language (in case) and its value should be set to "text/javascript".

Test

### Conclusion:

Hence we have our own applied validation for date using Java Scripts.

Name: Ujjwal Lade

Roll No: 06

Div: TE-B

Assignment No : 4

Write a program to design a registration form for students by using html , css & javascript and perform following validation : all fields are mandatory , phone number and email address validation.

File 1 validation.html

```
<!DOCTYPE html>

<html lang="en">
<head>
<meta charset="UTF-8">
<title>JavaScript Form validation</title>
<link rel="stylesheet" href="C:/Users/sandi/Desktop/wt4.css">
<script>

// Defining a function to display error message function printError(elemId, hintMsg) {
document.getElementById(elemId).innerHTML = hintMsg;
}

// Defining a function to validate form function validateForm() {

// Retrieving the values of form elements

var name = document.contactForm.name.value; var email = document.contactForm.email.value; var
mobile = document.contactForm.mobile.value; var country = document.contactForm.country.value;
var gender = document.contactForm.gender.value; var hobbies = [];

var checkboxes = document.getElementsByName("hobbies[]"); for(var i=0; i < checkboxes.length;
i++) { if(checkboxes[i].checked) {

// Populate hobbies array with selected values

hobbies.push(checkboxes[i].value);

}

}

// Defining error variables with a default value

var nameErr = emailErr = mobileErr = countryErr = genderErr = true;

// Validate name if(name == "") {

printError("nameErr", "Please enter your
name"); } else {
```

```
var regex = /^[a-zA-Z\s]+$/; if(regex.test(name) === false) { printError("nameErr", "Please enter a valid name"); }

} else

{ printError("nameErr",
"""); nameErr = false;
}

}

/ Validate email address if(email == "") {

printError("emailErr", "Please enter your email address"); } else {

/ Regular expression for basic email validation var regex = /^$+@\$.+\$+/;

if(regex.test(email) === false) {

printError("emailErr", "Please enter a valid email address");

} else{ printError("emailErr", ""); emailErr = false;
}

}

/ Validate mobile number if(mobile == "") {

printError("mobileErr", "Please enter your mobile number");

} else {

var regex = /^[1-9]\d{9}$/; if(regex.test(mobile) === false) {

printError("mobileErr", "Please enter a valid 10 digit mobile number");

} else{

printError("mobileErr", "");

mobileErr = false;
}

}

/ Validate country if(country == "Select")

{

printError("countryErr", "Please select your country");

} else {

printError("countryErr",
"""); countryErr = false;
}
```

```
}

/ Validate gender if(gender == "") {
printError("genderErr", "Please select your gender");
} else
{ printError("genderErr",
"""); genderErr = false;
}

/ Prevent the form from being submitted if there are any errors if((nameErr || emailErr || mobileErr
|| countryErr || genderErr) == true) {
return false;
} else {

/ Creating a string from input data for preview var dataPreview = "You've entered the following
details: \n" +
"Full Name: " + name + "\n" +
"Email Address: " + email + "\n" +
"Mobile Number: " + mobile + "\n" +
"Country: " + country + "\n" +
"Gender: " + gender + "\n";
if(hobbies.length) {
dataPreview += "Hobbies: " + hobbies.join(", ");
}
}

/ Display input data in a dialog box before submitting the form alert(dataPreview);
}

};

</script>
</head>
<body>

<form name="contactForm" onsubmit="return validateForm()"
action="/examples/actions/confirmation.php" method="post">

<h2>Application Form</h2>

<div class="row">
<label>Full Name</label>
```

```
<input type="text" name="name">
<div class="error" id="nameErr"></div>
</div>

<div class="row">
<label>Email Address</label>
<input type="text" name="email">
<div class="error" id="emailErr"></div>
</div>

<div class="row">
<label>Mobile Number</label>
<input type="text" name="mobile" maxlength="10">
<div class="error" id="mobileErr"></div>
</div>

<div class="row">
<label>Country</label>
<select name="country">
<option>Select</option>
<option>Australia</option>
<option>India</option>
<option>United States</option>
<option>United Kingdom</option>
</select>
<div class="error" id="countryErr"></div>
</div>

<div class="row">
<label>Gender</label>
<div class="form-inline">
<label><input type="radio" name="gender" value="male"> Male</label>
<label><input type="radio" name="gender" value="female"> Female</label>
</div>
<div class="error" id="genderErr"></div>
```

```
</div>

<div class="row">
    <label>Hobbies <i>(Optional)</i></label>
    <div class="form-inline">
        <label><input type="checkbox" name="hobbies[]" value="sports"> Sports</label> <label><input type="checkbox" name="hobbies[]" value="movies"> Movies</label>
        <label><input type="checkbox" name="hobbies[]" value="music"> Music</label>
    </div>
</div>

<div class="row">
    <input type="submit" value="Submit">
</div>
</form>
</body>
</html>
```

File 2 – wt4.css

```
body {
    font-size: 16px;
    background: #f9f9f9;
    font-family: "Segoe UI", "Helvetica Neue", Arial, sans-serif;
}

h2 {
    text-align: center;
    text-decoration: underline;
}

form {
    width: 300px;
    background: #fff;
    padding: 15px 40px 40px;
    border: 1px solid #ccc;
    margin: 50px auto 0;
    border-radius: 5px;
```

```
}

label {
    display: block;
    margin-bottom: 5px
}

label i {
    color: #999;
    font-size: 80%;
}

input, select {
    border: 1px solid #ccc;
    padding: 10px;
    display: block;
    width: 100%;
    box-sizing: border-box;
    border-radius: 2px;
}

.row {
    padding-bottom: 10px;
}

.form-inline {
    border: 1px solid #ccc;
    padding: 8px 10px 4px;
    border-radius: 2px;
}

.form-inline label, .form-inline input {
    display: inline-block;
    width: auto;
    padding-right: 15px;
}

.error {
```

```
color: red;  
font-size: 90%; }  
  
input[type="submit"] {  
font-size: 110%;  
font-weight: 100;  
background: #006dcc;  
border-color: #016BC1;  
box-shadow: 0 3px 0 #0165b6;  
color: #fff;  
margin-top: 10px;  
cursor: pointer; }  
  
input[type="submit"]:hover {  
background: #0165b6;  
}
```

Screenshot of a web browser showing an "Application Form" page. The form fields include:

- Full Name: Input field containing "sanchita kochar". Error message: "Please enter your name".
- Email Address: Input field containing "ssk". Error message: "Please enter a valid email address".
- Mobile Number: Input field containing "7777g". Error message: "Please enter a valid 10 digit mobile number".
- Country: Select dropdown menu set to "India".
- Gender: Radio buttons for "Male" and "Female", with "Female" selected.
- Hobbies (Optional): Checkboxes for "Sports", "Movies", and "Music", with "Sports" and "Movies" checked.

The "Submit" button is at the bottom of the form.

Screenshot of a web browser showing an "Application Form" page. The form fields include:

- Full Name: Input field containing "sanchita kochar".
- Email Address: Input field containing "ssk". Error message: "Please enter a valid email address".
- Mobile Number: Input field containing "7777g". Error message: "Please enter a valid 10 digit mobile number".
- Country: Select dropdown menu set to "India".
- Gender: Radio buttons for "Male" and "Female", with "Female" selected.
- Hobbies (Optional): Checkboxes for "Sports", "Movies", and "Music", with "Sports" and "Movies" checked.

The "Submit" button is at the bottom of the form.

# Assignment No 05

Page No. \_\_\_\_\_  
Date \_\_\_\_\_

Title - JSP servlet and MySQL (Backend)

Problem Statement is

- 1) Design and build Student login page using JSP Servlet and MySQL.
- 2) Design and build employee login page using JSP servlet and MySQL.

Theory:

Java Server Page (JSP): It is server side programming technology that is used to create dynamic web based application. JSP have right to use the complete java APIs including the JDBC API to access the databases.

Jsp tags can be used for different purposes such as retrieving information from a database or registering user preferences, accessing Java Beans components passing control between pages and sharing information between requests (page), etc.

Why we need JSP?

JSP is used for the design of dynamic web pages and servlet is used to code the logic that is present.

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 a servlet.
4. Once servlet is created, JSP engine compiles the servlet. Compilation error 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 variables and methods as shown below-

`<% text %>`

### 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 Java program that extends the capabilities of a server.

### Technology / tool

- 1) JSP and Servlets
- 2) IDE
- 3) Database : MySQL

NetBeans : NetBeans is an IDE used for quickly and easily developing Java desktop, mobile and web applications as well as HTML5.

MySQL : MySQL is a freely available open source Relational Database Management System (RDBMS), It is Structured Query Language (SQL)

### Test cases

Manual testing is used to validate the fields like Username, password, mobile number and email id's of the users entered by user with database.

### Conclusion

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

Name Ujjwal Lade

TE B 06

Assignment 5

Design and Build Student Login Page using JSP, Servlet and MySQL.

Login page (login.jsp)

```
<%@ 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>
```

Login controller (LoginController.java) 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 page (success.html)  
    <!DOCTYPE html>  
    <html> <head>  
        <meta charset="ISO-8859-1">  
        <title>Insert title here</title>  
    </head>  
    <body>  
        Login success  
    </body> </html>  
    error page (error.html)
```

```
<!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>
```

