STES's

SINHGAD COLLEGE OF ENGINEERING

Vadgaon(Bk), Pune

Department of Computer Engineering

LABORATORY MANUAL

2023-24

WEB TECHNOLOGY LABORATORY

BE-COMPUTER ENGINEERING

SEMESTER-II

INDEX

Web Technology Lab

All 10 assignments and 1 mini project are compulsory.

<table>
<tr><th>Sr. No.</th><th>Assignment Title</th></tr>
<tr><td>1.</td><td>Case study:
Before coding of the website, planning is important, students should visit different websites (Min.5) for the different client projects and note down the evaluation results for these websites, eithergood website or bad website in following format:
<table>
<tr><th>Sr. No.</th><th>Website URL</th><th>Purpose of Website</th><th>Things liked in the website</th><th>Things disliked in the website</th><th>Overall evaluation of the website (Good/Bad)</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>
From the evaluation, students should learn and conclude different website design issues, which should be considered while developing a website.</td></tr>
<tr><td>2.</td><td>Implement a web page index.htm for any client website (e.g., a restaurant website project) using following:
a. HTML syntax: heading tags, basic tags and attributes, frames, tables, images, lists, linksfor text and images, forms etc.
b. Use of Internal CSS, Inline CSS, External CSS</td></tr>
<tr><td>3.</td><td>Design the XML document to store the information of the employees of any businessorganization and demonstrate the use of:
a) DTD
b) XML Schema
And display the content in (e.g., tabular format) by using CSS/XSL.</td></tr>
<tr><td>4.</td><td>Implement an application in Java Script using following:
a) Design UI of application using HTML, CSS etc.
b) Include Java script validation
c) Use of prompt and alert window using Java Script

e.g., Design and implement a simple calculator using Java Script for operations like addition, multiplication, subtraction, division, square of number etc.
a) Design calculator interface like text field for input and output, buttons for numbers and operators etc.
b) Validate input values
c) Prompt/alerts for invalid values etc.</td></tr>
</table>

5.	Implement the sample program demonstrating the use of Servlet. e.g., Create a database table ebookshop (book_id, book_title, book_author, book_price, quantity) using database like Oracle/MySQL etc. and display (use SQL select query) the table content using servlet.
6.	Implement the program demonstrating the use of JSP. e.g., Create a database table students_info (stud_id, stud_name, class, division, city) using database like Oracle/MySQL etc. and display (use SQL select query) the table content using JSP.
7.	Build a dynamic web application using PHP and MySQL. a. Create database tables in MySQL and create connection with PHP. b. Create the add, update, delete and retrieve functions in the PHP web app interacting with MySQL database
8.	Design a login page with entries for name, mobile number email id and login button. Use strutsand perform following validations a. Validation for correct names b. Validation for mobile numbers c. Validation for email id d. Validation if no entered any value e. Re-display for wrongly entered values with message f. Congratulations and welcome page upon successful entries
9.	Design an application using Angular JS. e.g., Design registration (first name, last name, username, password) and login page using Angular JS.
10.	Design and implement a business interface with necessary business logic for any web applicationusing EJB. e.g., Design and implement the web application logic for deposit and withdraw amounttransactions using EJB.
11.	**Mini Project**: Design and implement a dynamic web application for any business functionalityby using web development technologies that you have learnt in the above given assignments.

Assignment No.1

Title: Evaluating Websites.

Problem Statement:

Case study:
Before coding of the website, planning is important, students should visit different websites (Min. 5) for the different client projects and note down the evaluation results for these websites, either good website or bad website (evaluation format given in the end).

From the Evaluation, Students should learn and conclude different websites design issues, which should be considered while developing a websites.

Introduction:

Many of you have done a fair amount of browsing and searching on the Internet. Have you ever stopped to question the content of sites you encounter when you are looking for cold, hard facts? Anyone can publish on the Internet and most of that content is not verified for accuracy, unlike many print journals and other publications. The job of fact verification is left up to you, the user. Expert searching of the Internet for information is a valuable skill, but knowing how to evaluate what you find is something of an art. The same skills that go into evaluating print materials can be applied to evaluating web content. Honing those skills until they become second nature will make the task of wading through the glut of information out there a little easier.

Objectives:

- In a search engine, use basic search strategies to bring back sites with information on a topic.
- Use evaluation techniques to determine authenticity and credibility of web sites.

Evaluation Criteria: - By following criteria we have to evaluate the websites and the evaluation criteria are:

1. **Browser and Operating System Support:**

   As different browsers and their version affects rendering ways. Different versions of HTML also support different sets of tags. So does the Website supports the every browser and Operating system? Or they have some exceptions.

2. **Bandwidth and Cache support**
   As User have different connection speed. So does the website have Low speed- response time more, so that user should move to other site with low speed internet. For this we

need to check if website have cache support. If there is Cache Support then it can store graphics into it. So that it can run with low speed internet.

3. **Display Resolution**

As the Display resolution of different devices are different so the website should Responsive to every resolution so that it will look nice and will fit properly on screen. If it is not responsive then maybe some part of screen will be blank if it have high resolution or maybe it will not fit in screen if it have low resolution.

4. **Website Design**

It decides the overall appearance of the Web site.
Web site theme – emphasizes on the unification of the design.
Eg- logo of company, Color theme for links, buttons, titles, labels, Use of pictures, messages related to subject

Fonts, Graphics and Colors- Different fonts have different readability and it affects the user's psychology. Height and width of the same character is different in different font which affects line ending and boundaries. Maintain consistency in using the font type and size. Select few and use them with different sizes and modes. Use CSS for designing and animations. Consider the availability of fonts on visitor's machine

Graphics- file format- different file format support different level of compression

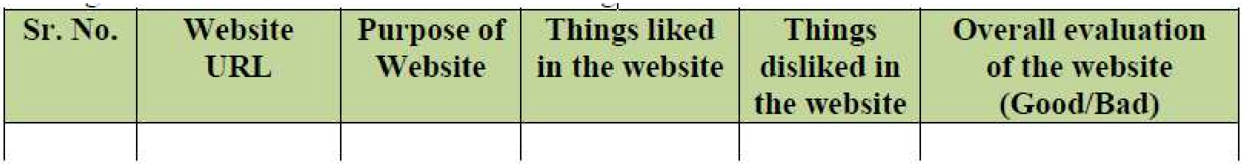
5. **Accessibility**

Is the site consistently available?
How many links are dead ends?
Does it cost money to use the site?
Do you have to register to use the site?

List your findings in following table:

URL	**Browser and Operating System Support**	**Bandwidth and Cache support**	**Display Resolution**	**Website Design**	**Accessibility**

And use following table for evaluation format:



Sr. No.	Website URL	Purpose of Website	Things liked in the website	Things disliked in the website	Overall evaluation of the website (Good/Bad)

Conclusion:

Hence, we did a case study on different websites by Evaluation criteria and evaluated the same domain different websites and compared them.

Assignment No. 2

Title: Web Page using HTML and CSS.

Problem Statement: Implement a web page index.html for any client website (e.g., a restaurant website project) using following:

a. HTML syntax: heading tags, basic tags and attributes, frames, tables, images, lists, links for text and images, forms etc.

b. Use of Internal CSS, Inline CSS, External CSS

Objectives: Students will be able to:

1. Understand Basic Tags of HTML
2. Understand CSS

Software and Hardware requirements:

i. **Software: html editor-**Notepad/Visual Studio code and Any Browser.

Theory:

1. HTML

Basic HTML

HTML stands for **Hyper Text Markup Language**. An HTML file is a text file containing markup tags. The markup tags tell the Web browser how to display the page. An HTML file must have an 'htm' or 'html' file extension. An HTML file can be created using a simple text editor.

Example: Creating a simple web page
1. Start Notepad.
2. Type in the following text :

```
<html>
        <head>
                <title>Title of page</title>
        </head>
        <body>
                This is a very basic webpage. <b>This text will be displayed in bold</b>
        </body>
</html>
```

3. Save the file as "firstpage.html".
4. Double click the saved file the browser will display the page.

HTML TAGS

- HTML tags are used to mark-up HTML elements
- HTML tags are surrounded by the two characters < and >
- The surrounding characters are called angle brackets
- HTML tags normally come in pairs like <b> and </b>
- The first tag in a pair is the start tag, the second tag is the end tag
- The text between the start and end tags is the element content
- HTML tags are not case sensitive, <b> means the same as <B>

HEADINGS

Headings are defined with the <h1> to <h6> tags. <h1> defines the largest heading. <h6> defines the smallest heading.

```
<h1>This is a heading</h1>
<h2>This is a heading</h2>
<h3>This is a heading</h3>
<h4>This is a heading</h4>
<h5>This is a heading</h5>
<h6>This is a heading</h6>
```

HTML automatically adds an extra blank line before and after a heading.

THE ANCHOR TAG AND THE HREF ATTRIBUTE

HTML uses the <a> (anchor) tag to create a link to another document. An anchor can point to any resource on the Web: an HTML page, an image, a sound file, a movie, etc.

The syntax of creating an anchor:

```
<a href="url">Text to be displayed</a>
```

The <a> tag is used to create an anchor to link, the href attribute is used to address the document to link to, and the words between the open and close of the anchor tag will be displayed as a hyperlink.

e.g.This anchor defines a link to **EEE 111** webpage:

```
<a href="http://faraday.ee.emu.edu.tr/eee111">Visit EEE 111</a>
```

The target attribute

With the target attribute, you can define **where** the linked document will be opened. The line below will open the document in a new browser window:

```
<a href="http://faraday.ee.emu.edu.tr/eee111" target="_blank"> Visit EEE 111</a>
```

The anchor tag and the name attribute

The name attribute is used to create a named anchor. When using named anchors we can create links that can jump directly into a specific section on a page, instead of letting the user scroll around to find what he/she is looking for. Below is the syntax of a named anchor:

```
<a name="label">Text to be displayed</a>
```

The name attribute is used to create a named anchor. The name of the anchor can be any text you care to use.

TABLES

Tables are defined with the <table> tag. A table is divided into rows (with the <tr> Tag), and each row is divided into data cells (with the <td> tag). The letters td stands for "table data," which is the content of a data cell. A data cell can contain text, images, lists, paragraphs, forms, horizontal rules, tables, etc.

```
<table border="1">
<tr>
        <td>row 1, cell 1</td>
        <td>row 1, cell 2</td>
</tr>
<tr>
        <td>row 2, cell 1</td>
        <td>row 2, cell 2</td>
</tr>
</table>
```

How it looks in a browser:

row 1, cell 1	row 1, cell 2
row 2, cell 1	row 2, cell 2

HTML supports ordered, unordered lists.

UNORDERED LISTS

An unordered list is a list of items. The list items are marked with bullets (typically small black circles).

An unordered list starts with the <ul> tag. Each list item starts with the <li> tag.

```
<ul>
<li>Coffee</li>
<li>Milk</li>
</ul>
```

Here is how it looks in a browser:

- Coffee
- Milk

Inside a list item you can put paragraphs, line breaks, images, links, other lists, etc.

ORDERED LISTS

An ordered list is also a list of items. The list items are marked with numbers. An ordered list starts with the <ol> tag. Each list item starts with the <li> tag.

```
<ol>
<li>Coffee</li>
<li>Milk</li>
</ol>
```

Here is how it looks in a browser:

1. Coffee
2. Milk

Inside a list item you can put paragraphs, line breaks, images, links, other lists, etc.

HTML Links - Syntax

The HTML <a> tag defines a hyperlink. It has the following syntax:

```
<a href="url">link text</a>
```

The most important attribute of the <a> element is the href attribute, which indicates the link's destination.
The *link text* is the part that will be visible to the reader.
Clicking on the link text, will send the reader to the specified URL address.

HTML Links - Use an Image as a Link

To use an image as a link, just put the <img> tag inside the <a> tag:

```
<a href="default.asp">
<img src="smiley.gif" alt="HTML tutorial" style="width:42px;height:42px;">
</a>
```

WHAT IS CSS?

Cascading **S**tyle **S**heets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.
CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, and variations in display for different devices and screen sizes as well as a variety of other effects.

Three Ways to Insert CSS

There are three ways of inserting a style sheet:

- External CSS
- Internal CSS
- Inline CSS

External CSS

With an external style sheet, you can change the look of an entire website by changing just one file!
Each HTML page must include a reference to the external style sheet file inside the <link> element, inside the head section.

Example:

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="mystyle.css">
```

```
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

An external style sheet can be written in any text editor, and must be saved with a .css extension.
The external .css file should not contain any HTML tags.
Here is how the "mystyle.css" file looks:

```
body {
  background-color: lightblue;
}

h1 {
  color: navy;
  margin-left: 20px;
}
```

Internal CSS

An internal style sheet may be used if one single HTML page has a unique style.
The internal style is defined inside the <style> element, inside the head section.

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: linen;
}
h1 {
  color: maroon;
  margin-left: 40px;
}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Inline CSS

An inline style may be used to apply a unique style for a single element. To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

```
<!DOCTYPE html>
<html>
<body>

<h1 style="color:blue;text-align:center;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>

</body>
</html>
```

DESIGN/EXECUTION STEPS:

Following steps are used to Create and Execute Web Page:

1. Start Notepad.
2. Write the HTML code in notepad and save the file as .html extension.
3. Write the CSS code in the notepad and save the file as .css extension.
4. Insert the CSS file in the HTML file using any type of inserting method as mentioned above.
5. Open the HTML file in the respective browser.

CONCLUSION: Hence we have implemented a web page using HTML and CSS.

Assignment No. 3

Title: XML and CSS.

Problem Statement: Design the XML document to store the information of the employees of any business organization and demonstrate the use of:

a) DTD
b) XML Schema

And display the content in (e.g. Tabular format) by using CSS/XSL.

Objectives: Students will be able to,

1. Design static webpage using XML.
2. Apply CSS to XML pages.
3. Write DTD and XML schema.

Software and Hardware requirements:

Software: Notepad/Visual Studio cod and Any Browser.

Theory:

1. **XML:**

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 the 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. XML is not going to replace HTML in the near future, but it introduces new possibilities by adopting many successful features of HTML.

There are three important characteristics of XML that make it useful in a variety of systems and solutions –

- 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 the World Wide Web Consortium (W3C) and is available as an open standard.

2. **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 a web page. By using CSS, one can control the color of text, style of fonts, spacing between paragraphs, layout

designs. CSS is easy to learn, easy to understand and it provides powerful control on presentation of an HTML document.

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 Independence.

CSS3 Modules: CSS3 Modules are having old CSS specifications as well as extension features.

- Box Model
- Selectors
- Background
- Border
- Image Values and Replaced Content
- Text Effects

3. XML Schema :

An XML Schema describes the structure of an XML document. The XML Schema language is also referred to as XML Schema Definition (XSD). The purpose of an XML Schema is to define the legal building blocks of an XML document:

- the elements and attributes that can appear in a document
- the number of (and order of) child elements
- data types for elements and attributes
- default and fixed values for elements and attribute

eg:

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="note">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="to" type="xs:string"/>
      <xs:element name="from" type="xs:string"/>
      <xs:element name="heading" type="xs:string"/>
      <xs:element name="body" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

</xs:schema>
```

4. DTD :

A DTD is a Document Type Definition. A DTD defines the structure and the legal elements and attributes of an XML document. With a DTD, independent groups of people can agree on a standard DTD for interchanging data.

An application can use a DTD to verify that XML data is valid.

Eg:

```
 <?xml version="1.0"?>
<!DOCTYPE note [
<!ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
]>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend</body>
</note>
```

DESIGN/EXECUTION STEPS:

Following steps are used to Create and Execute web applications:

1. Write the XML document to store the information of the employees of any business organization in notepad and save with .xml extension.

2. Write the external CSS code in notepad and save with .css extension.

3. Import CSS file in XML page.

4. Add DTD and XML schema to .xml file separately.

4. Open XML page in the browser

CONCLUSION/ANALYSIS:

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

Assignment No. 4

Title: Application in Java Script.

Problem Statement: Implement any web application in Java Script (e.g. Calculator) using following:

a) Design UI of application using HTML, CSS etc.

b) Include Java script validation

c) Use of prompt and alert window using Java Script

Objectives: Students will be able to:

1. Design an application using HTML, CSS, and Java Script.
2. Understand the concept of prompt and alert window in java script.

Software and Hardware requirements:

Software: Notepad/Visual Studio code and Any Browser.

Theory:

JavaScript

JavaScript is a very powerful **client-side scripting language**. JavaScript is used mainly for enhancing the interaction of a user with the webpage. In other words, you can make your webpage more lively and interactive, with the help of JavaScript. JavaScript is also being used widely in game development and Mobile application development.

How to Run JavaScript?

Being a scripting language, **JavaScript cannot run on its own. In fact, the browser is responsible for running JavaScript code**. When a user requests an HTML page with JavaScript in it, the script is sent to the browser and it is up to the browser to execute it. The main advantage of JavaScript is that **all modern web browsers support** JavaScript. So, you do not have to worry whether your site visitor uses Internet Explorer, Google Chrome, Firefox or any other browser. JavaScript will be supported. Also, JavaScript **runs on any operating system** including Windows, Linux or Mac.

Data validation is the process of ensuring that user input is clean, correct, and useful.

Typical validation tasks include:

- Has the user filled in all required fields?
- Has the user entered a valid date?
- Has the user entered text in a numeric field?

Most often, the purpose of data validation is to ensure correct user input. Validation can be defined by many different methods, and deployed in many different ways.

Server side validation is performed by a web server, after input has been sent to the server.

Client side validation is performed by a web browser, before input is sent to a web server.

The getElementById Method: The most common way to access an HTML element is to use the id of the element.

In the next example the getElementById method used id="demo" to find the element.

The innerHTML Property

The easiest way to get the content of an element is by using the innerHTML property.

The innerHTML property is useful for getting or replacing the content of HTML elements.

```
<html>
<body>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = "Hello World!";
</script>

</body>
</html>
```

In the example above, getElementById is a **method**, while innerHTML is a **property**.

Form Validation for Empty Inputs

Step 1) Add HTML:

Example:

```
<form name="myForm" action="/action_page.php" onsubmit="return
validateForm()" method="post" required>
  Name: <input type="text" name="fname">
  <input type="submit" value="Submit">
</form>
```

Step 2) Add JavaScript:
If an input field (fname) is empty, this function alerts a message, and returns false, to prevent the form from being submitted:

Example:

```
function validateForm() {
  var x = document.forms["myForm"]["fname"].value;
  if (x == "") {
    alert("Name must be filled out");
    return false;
  }
}
```

JavaScript has three kind of popup boxes: Alert box, Confirm box, and Prompt box.

Alert Box:

An alert box is often used if you want to make sure information comes through to the user.

When an alert box pops up, the user will have to click "OK" to proceed.

Syntax

window.alert("*sometext*");

The window.alert() method can be written without the window prefix. For **Example:**

```
alert("I am an alert box!");
```

Confirm Box

A confirm box is often used if you want the user to verify or accept something.
When a confirm box pops up, the user will have to click either "OK" or "Cancel" to proceed.
If the user clicks "OK", the box returns **true**. If the user clicks "Cancel", the box returns **false**.

Syntax

window.confirm("*sometext*");

The window.confirm() method can be written without the window prefix.

Example

```
if (confirm("Press a button!")) {
  txt = "You pressed OK!";
} else {
  txt = "You pressed Cancel!";
}
```

Prompt Box

A prompt box is often used if you want the user to input a value before entering a page. When a prompt box pops up, the user will have to click either "OK" or "Cancel" to proceed after entering an input value. If the user clicks "OK" the box returns the input value. If the user clicks "Cancel" the box returns null.

Syntax

window.prompt("*sometext*","*defaultText*");

The window.prompt() method can be written without the window prefix.

Example:

```
person = prompt("Please enter your name", "Harry Potter");
text=null;
if (person == null || person == "") {
  text = "User cancelled the prompt.";
} else {
  text = "Hello " + person + "! How are you today?";
}
```

DESIGN/EXECUTION STEPS:

Following steps are used to Create and Execute Application in JavaScript.

1. Start Notepad or any Text editor.
2. Create a **calculator.html** file with HTML code for calculator in it.
3. Create a .css extension file with CSS styling code in it.
4. Create a .js extension file with JavaScript validation code in it.
5. Link the CSS file in the HTML file using any inserting css file methods.
6. Link the JavaScript file in the HTML file under the <script> tag.
7. Save the HTML file and then open it in the browser.

CONCLUSION: Hence, we have implemented an application using Java Script for validation.

Assignment No. 5

Title: Implementation of Servlet.

Problem Statement: Implement the sample program demonstrating the use of Servlet.

e.g., create a database table ebookshop (book_id, book_title, book_author, book_price, quantity) using database like Oracle/MySQL etc. and display (use SQL select query) the table content using servlet.

Objectives: Students will be able to learn -

1. How to create a database and link it with the web application.
2. How to create a servlet.
3. Also how to display the database content using servlet.

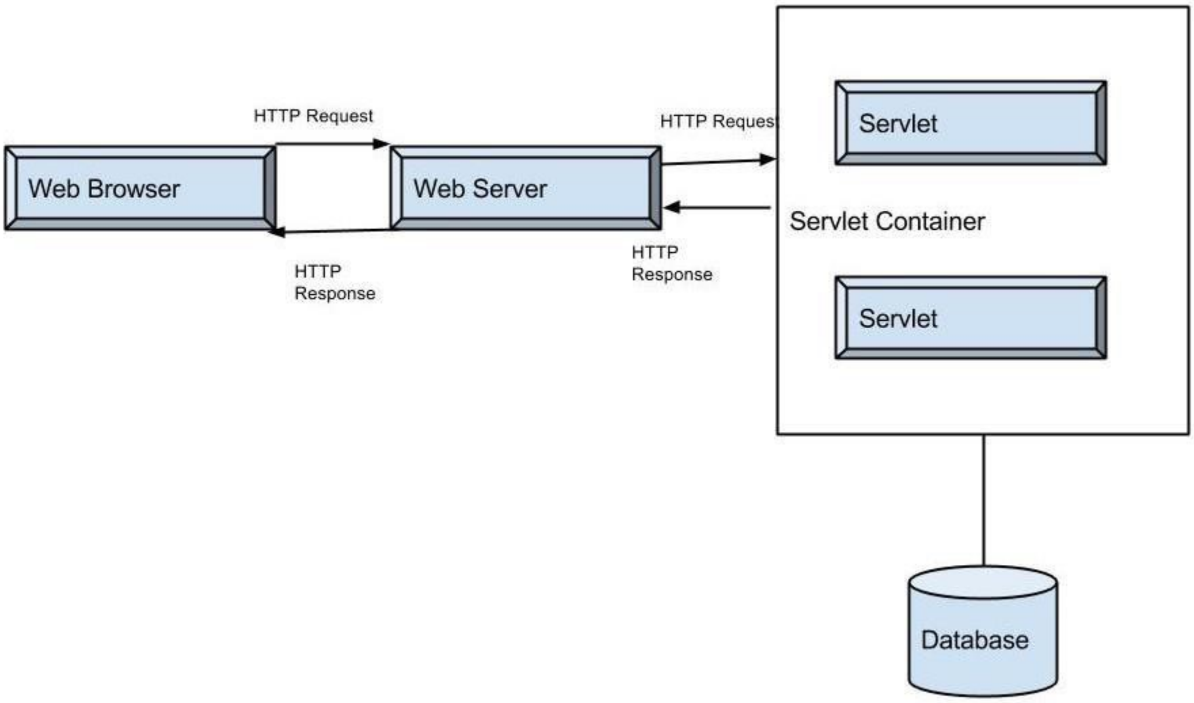
Software and Hardware requirements:

Software: MySQL server, Eclipse with J2EE, Tomcat web server and any browser.

Theory:

What is Servlet?

- Servlets are an important component of a J2EE application. Servlets along with JavaServer Pages (JSP) and EJB modules can be termed as server-side J2EE component types.
- Servlet is a Java Programming Language.
- Servlets are used to create web applications.
- Servlets are used to extend the applications hosted by web servers. Servlet runs in a J2EE application server



The process can be summarized as follows:

- A client sends a request to a Web Server, through a Web Browser.
- The Web Server searches for the required Servlet and initiates it.
- The Servlet then processes the client request and sends the response back to the server, which is then forwarded to the client.

Servlet API

The Servlet API is supported by all Servlet containers, such as Tomcat and Weblogic, etc. The Application Programming Interface (API) contains interface and classes to write a servlet program. The servlet API contains two packages as listed below:

- javax.servlet
- javax.servlet.http

Servlet LifeCycle

Servlets are small programs that run at server side and creates dynamic web pages. Servlets respond to any type of requests sent by user. In MVC architecture servlet act as controller. The controller is the logic that processes and responds to the user requests. Life Cycle of Servlets contain following steps:

- Load servlet class.
- Create servlet instance.
- Call the init() method.
- Call the service() method.
- Call the destroy() method.

Servlet Class Loading

The first step in creation of a servlet component is to load the servlet class file into web container's JVM (Java Virtual Machine). This step is invoked when either first time servlet is invoked or configured in the web.xml with load-on-startup element.

Creating Servlet Instance

After the servlet class has been loaded into the web container's JVM, the next step is to create an instance of that class. Servlet specification declares one and only one servlet instance will be created for a single definition in the deployment descriptor.

The init () method

After servlet instance is created, the web container initializes the parameters that were specified in the deployment descriptor. This method is invoked only when servlet is first loaded into memory. The syntax of init () method look like this:

```
public void init () throws Servlet Exception {
//code
}
```

The service () method

After the servlet component has been initialized, the web container can begin sending requests to that component using the service method .This method is used to process the request. For each request the web container will issue unique request and response to the service method. The syntax of service () method as follows:

```
public void service (Servletrequest request, Servletresponse response) throws
ServletException, IOException {
}
```

When service () method is called by web container it invokes doGet (), doPost (), doPut (), doDelete (), doTrace (), doOptions (), getLastModified () methods.

The destroy () method
When a web application is being shut down web container will call destroy method. It is used to clean up any resources that servlet might have initialized. The syntax of destroy () method as follows:

```
public void destroy(){
//code
}
```

MySQL
MySQL is a Relational DataBase Management System (RDBMS).
- DB stands for Database, a repository for the information store.

- The data in a database is organized into tables, and each table is organized into rows and columns.
- Each row in a table is called a record. A record may contains several pieces (called fields) of information, and each column in a table is known as a field.

-MS stands for Management System, the software that allows you to insert, retrieve, modify, or delete records.
-R stands for Relational, indicates a particular kind of DBMS that is good at relating information stored in one table to information stored in another table by looking for elements common to each of them.
MySQL operates using client/server architecture in which the server runs on the machine containing the databases and clients connect to the server over a network. MySQL is a multi-user database system, meaning several users can access the database simultaneously.

The server (MySQL server) listens for client requests coming in over the network and accesses database contents according to those requests and provides that to the clients.
- Clients are programs that connect to the database server and issue queries in a pre-specified format. The client program may contact the server programmatically (meaning a program call the server during execution) or manually.

For example, when you are issuing commands over a telnet session to a MySQL server, you are issuing the requests to the server by typing commands at your command prompt manually. On the other hand, if you have input some data (say your credit card information on the Internet towards purchase of some goods) in a form, and the form is processed by using a server side program, then the MySQL server is contacted programmatically. This is often the case in credit card approvals, member subscriptions etc.

DESIGN/EXECUTION STEPS:

1. Start MySQL server
2. Create database ‘mydatabase’.
3. Create table ‘Ebookshop’ in the ‘mydatabase’ with columns specified.
4. Add some sample row entries in the table.

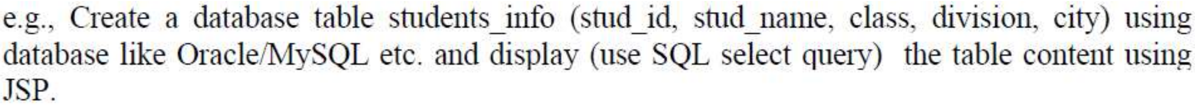
5. Create Servlet.
6. In Eclipse, go to **File -> New -> Dynamic Web Project** and click on it.
7. Enter the **Project Name** and save it.
8. Add servlet to project and Create Web Page
9. Connect servlet to MySQL server DB ‘mydatabase’ with proper dbusername and password (e.g. jdbc Connect)
10. Depending on table name in query get records from database
11. Display records trough servlet response in browser.

CONCLUSION: Hence, we have implemented and displayed the database content on web using Servlet.

Assignment No. 6

Title: JSP.

Problem Statement: Implement a program demonstrating use of JSP.



e.g., Create a database table students_info (stud_id, stud_name, class, division, city) using database like Oracle/MySQL etc. and display (use SQL select query) the table content using JSP.

Objectives: Students will be able to:

1. Understand JSP
2. Understand JSP connectivity with DBserver.

Software and/or Hardware requirements:

Software: eclipse IDE with J2EE, MySQL server, TOMCAT server and Any Browser.

Theory:

JSP technology is used to create web application just like Servlet technology. It can be thought of as an extension to Servlet because it provides more functionality than servlet such as expression language, JSTL, etc. A JSP page consists of HTML tags and JSP tags. The JSP pages are easier to maintain than Servlet because we can separate designing and development.

The Lifecycle of a JSP Page: - The JSP pages follow these phases:

- Translation of JSP Page
- Compilation of JSP Page
- Classloading (the classloader loads class file)
- Instantiation (Object of the Generated Servlet is created).
- Initialization (the container invokes jspInit() method).
- Request processing (the container invokes _jspService() method).
- Destroy (the container invokes jspDestroy() method).

Creating a simple JSP Page

To create the first JSP page, write some HTML code as given below, and save it by .jsp extension. E.g. save this file as index.jsp. Put it in a folder and paste the folder in the web-apps directory in apache tomcat to run the JSP page.

E.g. the simple example of JSP where we are using the scriptlet tag to put Java code in the JSP page as below **index.jsp, It will print 10 on the browser.**

```
<html>
<body>
<% out.print(2*5); %>
```

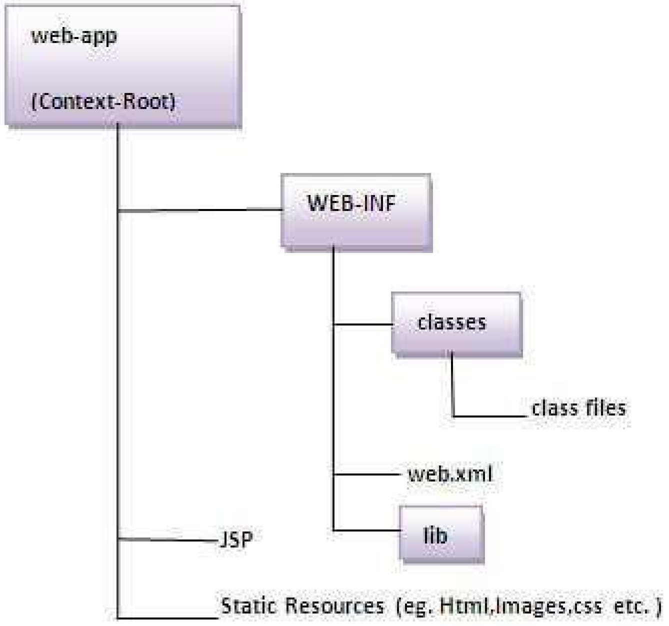
```
</body>
</html>
```

How to run a simple JSP Page?

Follow the following steps to execute this JSP page:

- Start the server
- Put the JSP file in a folder and deploy on the server
- Visit the browser by the URL http://localhost:portno/contextRoot/jspfile, for example, http://localhost:8888/myapplication/index.jsp

The Directory structure of JSP



JSP Scripting elements

The scripting elements provides the ability to insert java code inside the jsp. There are three types of scripting elements:

- scriptlet tag
- expression tag
- declaration tag

JSP scriptlet tag

A scriptlet tag is used to execute java source code in JSP. Syntax is as follows:

```
<%  java source code %>
```

Example code:

```
<html>
<body>
```

```
<% out.print("welcome to jsp"); %>
</body>
</html>
```

Expression tag

The code placed within JSP expression tag is written to the output stream of the response. So you need not write out.print() to write data. It is mainly used to print the values of variable or method.

Syntax of JSP expression tag

```
<%=  statement %>
```

Example of JSP expression tag

In this example of jsp expression tag, we are simply displaying a welcome message.

```
<html>
<body>
<%= "welcome to jsp" %>
</body>
</html>
```

JSP Declaration Tag

The JSP declaration tag is used to declare fields and methods. The code written inside the jsp declaration tag is placed outside the service() method of auto generated servlet. So it doesn't get memory at each request.

Syntax of JSP declaration tag

```
<%!  field or method declaration %>
```

Example of JSP declaration tag that declares field

```
<html>
<body>
<%! int data=50; %>
<%= "Value of the variable is:"+data %>
</body>
</html>
```

Example of JSP declaration tag that declares method

```
<html>
<body>
<%!   int cube(int n) {
```

```
return n*n*n*;
} %>
<%= "Cube of 3 is:"+cube(3) %>
</body>
</html>
```

Creating JSP in Eclipse IDE with Tomcat server as follows:

- Create a Dynamic web project in eclipse
- Create and add a .jsp file in project, and write your code.
- start tomcat server and deploy the project

DESIGN/EXECUTION STEPS:

1. Start MySQL server.
2. Create database ‘mydatabase’.
3. Create table ‘students_info’ in the ‘mydatabase’ with columns specified in problem statement.
4. Add some sample row entries in the table.
5. Create **Dynamic Web Project**.
6. In Eclipse, go to **File -> New -> Dynamic Web Project** and click on it.
7. Enter the **Project Name** and save it.
8. Add .jsp file to project and Create Web Page.
9. Connect servlet to MySQL server DB ‘mydatabase’ with proper dbusername and password (e.g. jdbc Connect)
10. Depending on table name in query get records from database.
11. Display records trough servlet response in browser.

CONCLUSION: Hence, we have implemented a program demonstrating use of JSP and displayed the database table content on web browser using JSP.

Assignment No.7

Title: Web application using PHP and MySQL.

Aim: Build a dynamic web application using PHP and MySQL.

a. Create database tables in MySQL and create connection with PHP.
b. Create the add, update, delete and retrieve functions in the PHP web app interacting with MySQL database.

Objective: Students will be able to:

1. Create database tables in MySQL and make connection with PHP

2. Understand the use of different functions like add, update, delete and retrieve in the PHP.

Software and/or Hardware Requirements:

Software: MySQL Server, Apache server, text/code editor/Notepad, any web browser.

Theory:

PHP:

PHP was developed in 1994 by Apache Group. PHP stands for PHP: Hypertext Preprocessor. It is a Server-Side Scripting language embedded in XHTML, mainly used for form handling and database access. It is free to use and download. It is an alternative to CGI, ASP, ASP.NET and JSP. The extension to the PHP files are .php, .php3, or .phtml.

Overview of PHP:

- The PHP processor works in two modes. If the PHP processor finds XHTML tags in the PHP script then the code is simply copied to the output file. But when it finds the PHP code in the script then that code is simply interpreted and the output is copied to the output file.
- The PHP script is not visible if we click on view source on the web browser as the output of PHP script is send directly to the browser but XHTML tags are visible.
- There is no need to declare variables in PHP as it uses dynamic typing.
- PHP has large number of library functions which makes it flexible to develop the code in PHP.

General Syntactic Characteristics:

- PHP code can be embedded in the XHTML document. The PHP code opens with "<?php" and closes with "?>" and statements are terminated by semicolon.
- If the PHP script is stored in other file and if it needs to be referred then "include" construct is used.
  Example: include ("myfile.inc")
- The variable names begin with the $ symbol.

- Though PHP is case sensitive, it has some keywords which are not case sensitive like if, else, echo, etc.
- The comments in PHP can be with #, //, /* **…** */.

Basic Syntax of PHP:

```
<?php

        …. ;

        … ;

?>
```

Example:

```
<html>
    <head>
            <title>Demo of PHP</title>
    </head>
    <body>
            <h1>Welcome…!!!</h1>
            <?php
                    $ a=10;
                    Echo“<h3> My First PHP page</h3>”;
                    echo“The value of the variable a is: $ a”;
            ?>
    </body>
</html>
```

MySQL:

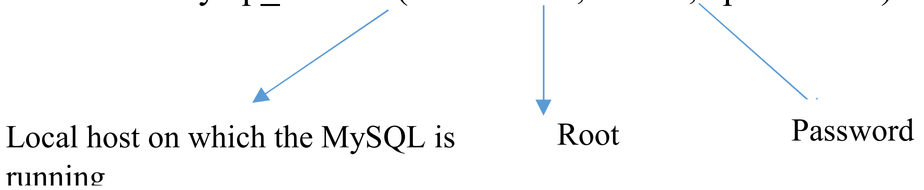
MySQL is an open-source relational database management system used for developing web-based software applications. It is most popular database system used with PHP. MySQL is developed, distributed and supported by Oracle Corporation.

- The data is stored in tables which consists of columns and rows.
- It is a database system which runs on a server.
- It compiles on a number of platforms.
- It is very fast, reliable and easy to use database system and uses standard SQL.

Connecting MySQL with PHP:

The PHP function mysql_connect connects to the MySQL server. We can pass three parameters that can be passed to this function.

```
mysql_connect(“localhost”, “root”, “password”) or die(mysql_error());
```



```
<?php
$servername = “localhost”;
$username = “username”;
$password = “password”;
//Create Connection
$conn = new mysqli ($servername, $username, $password);
// Check Connection
if ($conn->connect_error){
    die(“Connection Failed: ”.$conn->connect_error);
}
echo “Connected Successfully”;
?>
```

Example:

```
//Make a MySQL Connection
<?php
$conn=mysql_connect(“localhost:3306/mydb”, “root”, “password”);
if(!$conn){
die(‘error in connection’.mysql_error());
}
else{
print“connected”;
}
mysql_close($conn);  //closing the database
?>
```

Execution Steps:

1. Start Eclipse.
2. Create a database named as my-db, a table result_table for the database to store the values in the table.
3. Create a html form to accept the input from the user.
4. Create a PHP script to perform different functions like add, update, delete and retrieve.
5. Connect MySQL (i.e. database file) with the PHP files.
6. Load the HTML form which we have created in step no-3 in the web browser. Fill the fields which you have created and perform the different functions.

Conclusion: Thus, created dynamic web application successfully using MySQL and PHP.

Assignment No.8

Title: Design a login page using struts and perform validations.

Aim: Design a login page with entries for name, mobile number, email id and login button. Use struts and perform the following validations

a. Validation for correct names.
b. Validation for mobile numbers.
c. Validation for email id.
d. Validation if no entered any value.
e. Re-display for wrongly entered values with message.
f. Congratulations and welcome page upon successful entries.

Objective: Students will be able to:

1. To design a login page using struts and validate it.

Software and/or Hardware Requirements:

Software: Eclipse IDE, struts package

Theory:

Struts:

Struts is a framework software used for developing the Java Web Applications. The struts framework initially was created by Craig McClanahan and donated to Apache Software Foundation. It is an open source product. The latest commonly used release is struts 2.

Features:

- It is based on Model View Controller (MVC) architecture.
- The action class is POJO i.e. it is a simple Java class.
- Struts2 provides the support to AJAX technology, this means that only required field can be sent to the server and no need to send entire page. Due to this feature the execution is fast.
- Its framework can be integrated with other applications like Hibernate, Spring and so on.
- It provides tag support such as UI tags, Data tags, Control tags, etc. It also provides support for views and templates; and multiple views for the same applications.

Advantages:

- The complex web applications can be built using MVC components as it follows MVC framework.
- The development time is less and this creates an advantage for the application developers.
- The designs made using struts are easy to maintain, simpler and consistent.

Execution Steps:

1. Start Eclipse. Create a Dynamic Web Project. Copy the Required jar files to WEB-INF/lib folder.
2. Create one more JSP document under Web Content or webapp and name the folder.
3. Write a java file for validating the username, mobile number and email-id.
4. Write the configuration files. The struts.xml file under the WEB_INF/classes folder.
5. Modify the web.xml file which is present inside the WEB_INF folder.
6. Execute the application, by right clicking on the project name in the Project Explorer window, click on Run As->Run Configuration.2

Conclusion: Thus understood and implemented login page using struts and validation successfully.

Assignment No. 9

Title: Application using Angular JS.

Problem Statement: Design an application using Angular JS.

e.g., Design registration (first name, last name, username, password) and login page using Angular JS.

Objectives: Students will be able to learn

To understand implementation of AngularJS

Software and Hardware requirements: MYSQL server, text editor and any browser.

Theory:

ANGULARJS INTRODUCTION

AngularJS is a **JavaScript framework**. It can be added to an HTML page with a <script> tag. AngularJS extends HTML attributes with **Directives**, and binds data to HTML with **Expressions**.

ANGULARJS IS A JAVASCRIPT FRAMEWORK

AngularJS is a JavaScript framework. It is a library written in JavaScript.
AngularJS is distributed as a JavaScript file, and can be added to a web page with a script tag:

```
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
```

ANGULARJS EXTENDS HTML

AngularJS extends HTML with **ng-directives**.
The **ng-app** directive defines an AngularJS application.
The **ng-model** directive binds the value of HTML controls (input, select, textarea) to application data.
The **ng-bind** directive binds application data to the HTML view.

ANGULARJS EXAMPLE

```
<!DOCTYPE html>
<html>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
<body>
<div ng-app="">
<p>Name: <input type="text" ng-model="name"></p>
<p ng-bind="name"></p>
</div>
</body>
</html>
```

Example explained:

- AngularJS starts automatically when the web page has loaded.
- The **ng-app** directive tells AngularJS that the <div> element is the "owner" of an AngularJS **application**.

- The **ng-model** directive binds the value of the input field to the application variable **name**.
- The **ng-bind** directive binds the **innerHTML** of the <p> element to the application variable **name**.

ANGULARJS DIRECTIVES

As you have already seen, AngularJS directives are HTML attributes with an **ng** prefix. The **ng-init** directive initializes AngularJS application variables.

ANGULARJS EXAMPLE :

```
<div ng-app="" ng-init="firstName='John'">
<p>The name is <span ng-bind="firstName"></span></p>
</div>
```

Alternatively with valid HTML:

ANGULARJS EXAMPLE

```
<div data-ng-app="" data-ng-init="firstName='John'">
<p>The name is <span data-ng-bind="firstName"></span></p>
</div>
```

You can use **data-ng-**, instead of **ng-**, if you want to make your page HTML valid.

ANGULARJS EXPRESSIONS

AngularJS expressions are written inside double braces: **{{ expression }}**.
AngularJS will "output" data exactly where the expression is written:

AngularJS expressions are much like **JavaScript expressions:** They can contain literals, operators, and variables.
Example {{ 5 + 5 }} or {{ firstName + " " + lastName }}

ANGULARJS EXAMPLE

```
<!DOCTYPE html>
<html>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
<body>
<div ng-app="">
<p>Name: <input type="text" ng-model="name"></p>
<p>{{name}}</p>
</div>
</body>
</html>
```

ANGULARJS APPLICATIONS

AngularJS **modules** define AngularJS applications. AngularJS **controllers** control AngularJS applications.
The **ng-app** directive defines the application, the **ng-controller** directive defines the controller.

ANGULARJS EXAMPLE

```
<div ng-app="myApp" ng-controller="myCtrl">
```

```
First Name: <input type="text" ng-model="firstName"><br> Last Name: <input type="text"
ng-model="lastName"><br>
<br>
Full Name: {{firstName + " " + lastName}}
</div>
<script>
var app = angular.module('myApp', []); app.controller('myCtrl', function($scope) {
$scope.firstName= "John";
$scope.lastName= "Doe";
});
</script>
```

AngularJS modules define applications:

ANGULARJS MODULE

var app = angular.module('myApp', []); AngularJS controllers control applications:

ANGULARJS CONTROLLER

```
app.controller('myCtrl', function($scope) {
$scope.firstName= "John";
$scope.lastName= "Doe";
});
```

DESIGN/EXECUTION STEPS:

1) Create Web Pages which allow user to register and login:
    a. Registration form with proper buttons and text fields like: firstname, lastname, username and password
    b. Login page with submit button and text fields-username and password
2) Connect to MySQL to save and access user data.
3) Match entered user name and password to allow user to login.
4) In case of mismatch of them and then don’t allow login.

CONCLUSION: Hence we created an application using AngularJS.

Assignment No.10

Title: Study of EJB.

Aim: To study the Enterprise Java Beans (EJB).

Objective: Students will be able to:

1. To learn the Enterprise Java Beans (EJB).

Software and/or Hardware Requirements:

Software: JDK 1.5 or above, NetBeans 7.3, JBoss application server

Theory:

- EJB stands for "Enterprise Java Beans". EJB is an essential part of a J2EE platform. J2EE platform has component based architecture to provide multi-tiered, distributed and highly transactional features to enterprise level applications.
- EJB provides an architecture to develop and deploy component based enterprise applications considering robustness, high scalability, and high performance.
- An EJB application can be deployed on any of the application server compliant with the J2EE 1.3 standard specification.

Types of EJB:

Primarily, the EJB is divided into three categories as follows:

1. **Session Bean:**

   Session bean stores data of a particular user for a single session. It can be **stateful** or **stateless**. It is less resource intensive as compared to entity bean. Session bean gets destroyed as soon as user session terminates.

2. **Entity Bean:**

   Entity beans represent persistent data storage. User data can be saved to database via entity beans and later on can be retrieved from the database in the entity bean.

3. **Message Driven Bean:**

   Message driven beans are used in context of JMS (Java Messaging Service). Message Driven Beans can consume JMS messages from external entities and act accordingly.

Benefits of EJB:

- Simplified development of large-scale enterprise level application.
- Application Server/EJB container provides most of the system level services like transaction handling, logging, load balancing, persistence mechanism, exception handling, and so on. Developer has to focus only on business logic of the application.
- EJB container manages life cycle of EJB instances, thus developer needs not to worry about when to create/delete EJB objects.

Environment Setup:

Step1: Verify Java Installation in your System.

Step2: Set Java Environment.

Step3: Download and Install NetBeans IDE.

Step4: Setup JBoss Application Server.

Step5: Configure JEE plugin to NetBeans.

Step6: Configure JBoss Server to NetBeans.

Step7: Install Database Server(PostGresql)

Difference between Stateless Beans and Stateful Beans:

STATELESS BEANS	STATEFUL BEANS
1) It is used to perform independent operations.	1) It preserve the conversational state with client.
2) It doesn’t have any associated client state but it may preserve its instance state.	2)It keeps associated client state in its instance variables.
3) EJB Container normally creates a pool of few stateless bean's objects and use these objects to process client's request.	3) EJB Container creates a separate stateful session bean to process client's each request.

Annotations

Annotations were introduced in Java 5.0. The purpose of having annotations is to attach additional information in the class or a meta-data of a class within its source code. In EJB 3.0, annotations are used to describe configuration meta-data in EJB classes. By this way, EJB 3.0 eliminates the need to describe configuration data in configuration XML files.

Callbacks

Callback is a mechanism by which the life cycle of an enterprise bean can be intercepted. EJB 3.0 specification has specified callbacks for which callback handler methods are created. EJB Container calls these callbacks. We can define callback methods in the EJB class itself or in a separate class. EJB 3.0 has provided many annotations for callbacks.

Timer Service

Timer Service is a mechanism by which scheduled application can be build. For example, salary slip generation on the 1st of every month. EJB 3.0 specification has specified @Timeout annotation, which helps in programming the EJB service in a stateless or message driven bean. EJB Container calls the method, which is annotated by

@Timeout.EJB Timer Service is a service provided by EJB container, which helps to create timer and to schedule callback when timer expires.

Dependency Injection

EJB 3.0 specification provides annotations, which can be applied on fields or setter methods to inject dependencies. EJB Container uses the global JNDI registry to locate the dependency. Following annotations are used in EJB 3.0 for dependency injection.

- **@EJB** − used to inject other EJB reference.
- **@Resource** − used to inject datasource or singleton services like sessionContext, timerService etc.

Interceptors

EJB 3.0 provides specification to intercept business methods calls using methods annotated with @AroundInvoke annotation. An interceptor method is called by ejbContainer before business method call it is intercepting. Following is the example signature of an interceptor method

```
@AroundInvoke
public Object methodInterceptor(InvocationContext ctx) throws Exception {
   System.out.println("*** Intercepting call to LibraryBean method: "
   + ctx.getMethod().getName());
   return ctx.proceed();
}
```

Embeddable Objects

EJB 3.0 provides option to embed JAVA POJO (Plain Old Java Object) into an entity bean and allows to map column names with the methods of the embedded POJO class. A java POJO to be embedded must be annotated as @Embeddable.

```
@Embeddable
public class Publisher implements Serializable{
   private String name;
   private String address;
   ...
}
```

Conclusion: Thus, studied the Enterprise Java Beans (EJB).

Assignment No.11

Title: WTL Mini-project

Problem statement of Mini-Project: *write here your project's prob stmt.*

Objective: To learn web technologies like HTML, CSS, JS, JSP/SERVELET/ASP.NET/PHP, MYSQL, XML, etc. to implement dynamic web application.

Technologies Used:

S/W:

H/W:

Introduction of Topic/Project: (*scope of the work and its importance*)

Project design/functionalities/features: (*explain module by module***)**

Working/flowchart/algorithms: *(as applicable)*

Other information: (*any other important relevant info as applicable)*

Conclusion:

Code: *(all code files with their names and proper extension*)

Output/screenshots: (*all features should be covered*)

******note**: for this assignment, no write-up, just prepare this report and print it.****