Module 3) Web Technologies in Java

- ➢ HTML Tags: Anchor, Form, Table, Image, List Tags, Paragraph, Break, Label

  ➢ Theory:

  1.Introduction to HTML and its structure

  Ans:

  HTML stands for HyperText Markup Language.

  It is the standard language used to create and design web pages.

- It structures content on the web like text, images, videos, links, etc.
- HTML uses tags (elements) to define different parts of a web page.
- It works with CSS (for styling) and JavaScript (for interactivity).

2. Explanation of key tags:

Ans:

<a>: Anchor tag for hyperlinks:-

The <a> tag in HTML is called the anchor tag, and it's used to create hyperlinks — which let users click and navigate to another page, website, file, or section.

<form>: Form tag for user input:-

The <form> tag in HTML is used to collect user input and send it to a server for processing (like login forms, registration, feedback, etc.).

<table>: Table tag for data representation:-

The <table> tag in HTML is used to display tabular data — i.e., information arranged in rows and columns (like in Excel).

<img>: Image tag for embedding images:-

The <img> tag in HTML is used to embed images into a web page.

List tags:

i) <ul>:-Unordered List

- displays list items with bullets.
- used when the order doesn't matter.

ii) <ol> :- Ordered List

- displays list items with numbers (1, 2, 3...) or letters (A, B, C...).
- used when the order does matter.

iii) <li>:- List Item

- stands for List Item
- used inside both `<ul>` and `<ol>`

<p>: Paragraph tag:-

The <p> tag in HTML is used to define paragraphs of text.

<br>: Line break:-

The <br> tag in HTML is used to insert a line break it moves the text to the next line without starting a new paragraph.

<a>: Label for form inputs:-

The <label> tag in HTML is used to associate text with form input elements — such as text boxes, checkboxes, radio buttons, etc. It improves accessibility and usability.

- ➢ Lab Exercise:

3. Create a webpage that includes:

o A navigation menu with anchor tags.

o A form with input fields, labels, and a submit button.

o A table that displays user data.

o Images with appropriate alt text

o Both ordered and unordered lists.

Ans:

```
            <!DOCTYPE html>
<html>
<head>
    <title>Simple HTML Page</title>
</head>
<body>

    <!--  Navigation Menu -->
    <h2>Navigation Menu</h2>
    <a href="#form">Go to Form</a> |
    <a href="#table">Go to Table</a> |
     <a href="#images">Go to Images</a> |
    <a href="#lists">Go to Lists</a>

    <!--  Form Section -->
    <h2 id="form">User Form</h2>

    <!--  Lists Section -->
     <h2 id="lists">My Lists</h2>

<h3>Ordered List (Steps)</h3>
<ol>
  <li>Wake up</li>
  <li>Eat breakfast</li>
  <li>Study HTML</li>
</ol>

<h3>Unordered List (Fruits)</h3>
<ul>
  <li>Apple</li>
  <li>Banana</li>
  <li>Mango</li>
```

```
          </ul>


      </body>

      </html>
```

- ➢ CSS: Inline CSS, Internal CSS, External CSS
    - ➢ Theory:

3.Overview of CSS and its importance in web design.

Ans:

CSS (Cascading Style Sheets) is a stylesheet language used to describe the presentation (look and feel) of a document written in HTML or XML.
It controls the layout, colors, fonts, spacing, and overall visual styling of web pages

4.Types of CSS:

Ans:

Inline CSS: Directly in HTML elements:-

CSS is written directly inside the HTML tag using the style attribute.

Internal CSS: Inside a <style> tag in the head section:-

CSS is written within a <style> tag inside the <head> section of the HTML document.

External CSS: Linked to an external file:-

CSS is written in a separate .css file and linked to the HTML document.

➢ Lab Exercise:

5. Create a webpage where:

o You apply inline CSS to an element.

o Use internal CSS for another element.

o Link an external CSS file to

ans:

```
   <!DOCTYPE html>
<html>
<head>
 <title>TOPS Technologies 2024</title>

 <!-- Link to external CSS -->
 <link rel="stylesheet" href="style.css">

 <!-- Internal CSS -->
 <style>
  h2 {
   color: darkblue;
   font-size: 28px;
   text-align: center;
  }
 </style>
```

```
</head>

<body>


  <!-- Inline CSS -->

  <h1 style="color: red; text-align: center;">Welcome to TOPS Technologies 2024</h1>


  <!-- Internal CSS applied -->

  <h2>Learn & Grow With Us</h2>


  <!-- External CSS applied -->

  <p>This paragraph is styled using external CSS.</p>


</body>

</html>
```

❖ CSS: Margin and Padding

➢ Theory:

6. Definition and difference between margin and padding

Ans:

Margin:

The margin is the space outside the border of an element. It creates space between the element and other elements around it.

Padding:

The padding is the space inside the border of an element. It creates space between the content and the element's border.

7. How margins create space outside the element and padding creates space inside.

Ans:

i) Padding – Inside the element

- Adds space between the content and the border.
- Makes content more readable and spaced out inside the box.

ii) Margin – Outside the element

- Adds space outside the border, pushing the entire box away from surrounding elements.
- Separates elements from each other.

➢ Lab Exercise:

8. Create a webpage and use CSS to demonstrate:
o Margin applied to an element.
o Padding applied to a div.
o The effect of different margin and padding values on the layout.

Ans:

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <title>Margin and Padding Demo</title>

  <style>

    body {
```

```
  font-family: Arial, sans-serif;
}

.box1 {
  background-color: lightblue;
  padding: 20px;
  border: 2px solid blue;
}

.box2 {
  background-color: lightgreen;
  margin: 30px;
  border: 2px solid green;
}

.box3 {
  background-color: lightcoral;
  padding: 10px 30px;
  margin: 40px 10px;
  border: 2px solid red;
}

.container {
  background-color: #f0f0f0;
  padding: 20px;
```

```
    }

    h2 {
      margin-bottom: 10px;
    }
  </style>
</head>
<body>

  <div class="container">
    <h2>Padding Example (Box 1)</h2>
    <div class="box1">
      This div has <strong>padding: 20px</strong>. The content
is pushed inside.
    </div>

    <h2>Margin Example (Box 2)</h2>
    <div class="box2">
      This div has <strong>margin: 30px</strong>. It is pushed
away from others.
    </div>

    <h2>Mixed Margin and Padding (Box 3)</h2>
    <div class="box3">
      This div has <strong>padding: 10px 30px</strong> and
        <strong>margin: 40px 10px</strong>.
    </div>
  </div>


</body>
</html>
```

❖ CSS: Pseudo-Class

➢ Theory:

9. Introduction to CSS pseudo-classes like :hover, :focus, :active, etc.

Ans:

A pseudo-class is used to define a special state of an element — like when a user hovers over a button, focuses on an input field, or clicks a link.

i)hover

- Triggered when the user moves the mouse over an element.
- Commonly used for buttons, links, etc

ii)focus

- Triggered when an element (like a form input) gets keyboard or mouse focus.
- Useful for highlighting form fields.

iii) active

- Triggered while an element is being clicked.
- Often used on buttons or links

10.Use of pseudo-classes to style elements based on their state.

Ans:

CSS pseudo-classes are used to define a special state of an element without adding extra classes or IDs in HTML. They allow you to style elements based on their interaction or position in the document.

- Lab Exercise:
  11. Create a navigation menu and use pseudo-classes to:
  o Change the color of links on hover.
  o Style form inputs when they are focused.
  Ans:

```
        <!DOCTYPE html>
<html>
<head>
<style>
/* Basic page styling */
body {
  font-family: Arial, sans-serif;
  margin: 20px;
}

/* Navigation menu styling */
nav {
  background-color: #333;
  padding: 10px;
}
nav ul {
  list-style: none;
  margin: 0;
  padding: 0;
  display: flex;
}
nav ul li {
  margin-right: 20px;
}
```

```
nav ul li a {
  color: white;
  text-decoration: none;
  padding: 8px 12px;
  display: inline-block;
}
/* Pseudo-class for hover on navigation links */
nav ul li a:hover {
  background-color: #575757;
  color: yellow;
  border-radius: 4px;
}

/* Form styling */
form {
  margin-top: 20px;
}
/* Style input fields when focused */
input[type="text"], input[type="email"] {
  padding: 8px;
  width: 250px;
  margin-bottom: 10px;
  border: 1px solid #ccc;
  border-radius: 4px;
}
/* Pseudo-class for focus */
input[type="text"]:focus, input[type="email"]:focus {
  border-color: blue;
  outline: none;
  background-color: #f0f8ff;
}
</style>
</head>
<body>
```

```
<h2>Navigation Menu with Pseudo-Classes</h2>

<!-- Navigation Menu -->
<nav>
  <ul>
    <li><a href="#">Home</a></li>
    <li><a href="#">About</a></li>
    <li><a href="#">Services</a></li>
    <li><a href="#">Contact</a></li>
  </ul>
</nav>

<!-- Form -->
<form>
  <label for="name">Name:</label><br>
  <input type="text" id="name" name="name"
placeholder="Enter your name"><br>

  <label for="email">Email:</label><br>
  <input type="email" id="email" name="email"
placeholder="Enter your email"><br>

  <button type="submit">Submit</button>
</form>

</body>
</html>
```

- ❖ CSS: ID and Class Selectors
- ➢ Theory:

12. Difference between id and class in CSS.

Ans:

Feature	id	class
Uniqueness	Only once per page	Multiple times
Selector	#idname	.classname
Specificity	Higher	Lower
Usage	Unique elements	Reusable styles

13.Usage scenarios for id (unique) and class (reusable).

Ans:

id :-For unique elements (used once on a page)
Examples: header, footer, main section, unique button, unique form.

class :-For reusable styles (used on multiple elements)
Examples: buttons, navigation links, cards, text highlights.

➢ Lab Exercise:

14.Create a webpage where:

o You apply an id to an element and style it uniquely.
o Use class to apply the same style to multiple elements.

Ans:

```
<!DOCTYPE html>
<html>
  <head>
  <style>
  /* Style for unique element using id */
  #main-heading {
```

```
  color: darkblue;
  text-align: center;
  font-size: 28px;
  margin-bottom: 20px;
}

/* Style for multiple elements using class */
.highlight {
  background-color: yellow;
  color: black;
  padding: 5px;
  border-radius: 4px;
}
</style>
</head>
<body>

<!-- Unique element with id -->
<h1 id="main-heading">Welcome to My Webpage</h1>

<!-- Multiple elements with same class -->
<p class="highlight">This is the first highlighted
paragraph.</p>
<p class="highlight">This is the second highlighted
paragraph.</p>
```

```
<p>This paragraph is normal and not highlighted.</p>

</body>
</html>
```

❖ Introduction to Client-Server Architecture

➢ Theory:

15.Overview of client-server architecture.

Ans:

Client-Server Architecture is a network design model where tasks or services are divided between two main entities:

- Client: The requester of services or resources.
- Server: The provider of those services or resources.

16. Difference between client-side and server-side processing.

Ans:

Feature	Client-Side	Server-Side
Execution	Browser (user's device)	Server
Speed	Fast for UI changes	Slower due to server communication
Security	Less secure (code is visible)	More secure (code hidden)
Examples	Form validation, animations	Login verification, database queries

17.Roles of a client, server, and communication protocols.
Ans:

i)Role of Client

- Definition: The client is the device or application that requests services or resources from the server.
- Responsibilities:
  - Initiates requests to the server.
  - Provides user interface for interaction (e.g., web browser, mobile app).
  - Handles presentation logic (displaying data, user input validation).

ii)Role of Server

- Definition: The server is the system that processes client requests and provides services or resources.
- Responsibilities:
  - Handles business logic and heavy computations.
  - Stores and manages data (databases, files).
  - Sends responses back to clients.

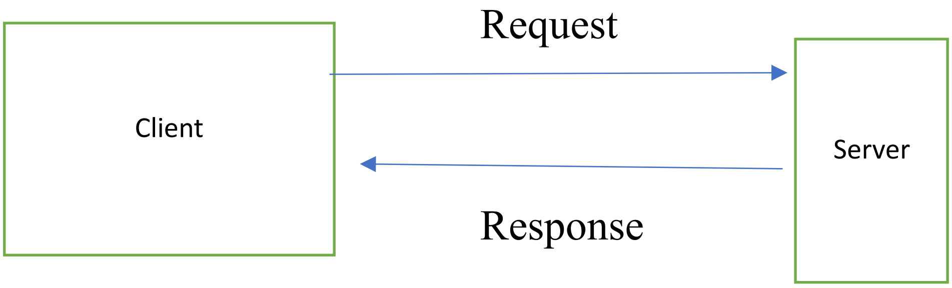
iii)Role of Communication Protocols

- Definition: A set of rules that define how data is transmitted and received between client and server.
- Responsibilities:
  - Ensure proper communication between client and server.
  - Define data format, error handling, and connection rules.

- Lab Exercise:

  18. Create a diagram explaining client-server communication flow and explain howa request isprocessed by the server and sent back to the client.

Ans:



❖ J2EE Architecture Overview

➢ Theory:

19.Introduction to J2EE and its multi-tier architecture.

Ans:

J2EE (Java 2 Platform, Enterprise Edition) is a platform for building enterprise-level applications in Java.It provides APIs and runtime environment for developing large-scale, distributed, and component-based applications.

Supports technologies like Servlets, JSP, EJB, JMS, JNDI, JDBC.

i) Client Tier (Presentation Layer)

- Runs on the client machine.
- Handles UI and user interaction.
- Examples: Web browser, mobile app.

ii) Web Tier

- Runs on web server.
- Handles HTTP requests/responses.
- Components: Servlets, JSP.

iii) Business Tier

- Runs on application server.
- Contains business logic.

- Components: Enterprise JavaBeans (EJB).

iv) Enterprise Information System (EIS) Tier

- Handles data storage and legacy systems.
- Examples: Databases (JDBC), ERP systems.

20.Role of web containers, application servers, and database servers.

Ans:

i) Web Container (Servlet Container)

- Definition:
  A web container is part of a web server or application server that manages the execution of web components like Servlets and JSP.

ii)Application Server

- Definition:
  An application server provides an environment for business logic execution and enterprise-level services.

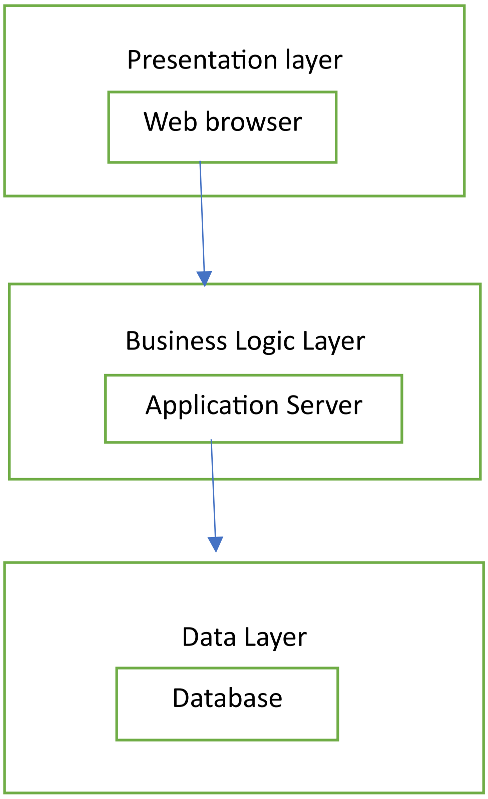
iii)Database Server

- Definition:
  A database server is a system that stores, manages, and retrieves data requested by application servers or web containers.

- ➢ Lab Exercise:

21. Draw and explain the J2EE architecture, labeling the layers like the presentation layer, business logic layer, and data layer.

Ans:



i). Client Tier (Presentation Layer)

- Role: Interacts with the end user.
- Technologies: HTML, JSP (Java Server Pages), Servlets, JSF.
- Function: Displays data to the user and collects input from the user.
- Example: A web page or mobile app screen where the user logs in.

ii). Web Tier (Controller Layer)

- Role: Processes client requests and manages user interactions.
- Technologies: Servlets, JSP, JavaBeans.
- Function: Acts as the controller in MVC, forwarding requests to the business logic layer.
- Example: A Servlet that processes login data and decides whether to forward to the home page or an error page.

iii). Business Tier (Business Logic Layer)

- Role: Implements business rules and application logic.
- Technologies: EJB (Enterprise Java Beans), POJOs, Spring (in modern apps).
- Function: Handles core business processes such as transactions, validations, and computations.
- Example: Processing an order, checking inventory, calculating discounts.

iv). Integration Tier (Middleware Layer)

- Role: Connects business logic with enterprise systems and services.
- Technologies: JMS (Java Messaging Service), JCA (Java Connector Architecture).
- Function: Manages messaging and integration with other systems.
- Example: Sending an order confirmation message to a queue for shipping.

v). Data Tier (Persistence Layer)

- Role: Stores and retrieves data from the database.
- Technologies: JDBC (Java Database Connectivity), JPA (Java Persistence API).
- **Function:** Handles database operations like CRUD (Create, Read, Update, Delete).
- **Example:** Saving a new user record in the database.