# CHAPTER 11

# WEB TECHNOLOGIES

**Syllabus:** Web technologies: HTML, XML, basic concepts of client–server computing.

## 11.1 INTRODUCTION

To communicate over the Internet, Web servers and Web browsers act as client−server programs. Web data is distributed over Internet using these programs. HTML is a markup language that is used for presentation and interaction with people in Web browsers. Web servers use IP address and port numbers for identification. Using Web browser, data request is sent using HTTP protocol. Apache, Tomcat and IIS are popular Web-server programs, whereas Internet Explorer (IE), Chrome, Safari, Opera and Firefox are popular Web browsers.

The World Wide Web (WWW), also termed as 'web', is an information medium which is accessed through the Internet. The following table describes the major milestones in the development of the WWW (Table 11.1).

**Table 11.1** | Major milestones in development of Web

| | |
|---|---|
| 1980 | Tim Berners-Lee invents the WWW, at CERN (the world famous nuclear research lab at Switzerland). |
| 1990 | Concepts like − HTTP, Web browser, Uniform Resource Identifier (URI) and HTML |
| 1993 | Launch of the first graphical web browser, named − MOSAIC at USA. |
| 1994 | Hosting of the first International WWW Conference, formation of W3C |
| 1996 | Commercialization of the Web |
| 1998 | Google was founded by Larry Page and Sergey Brin |
| 1999 | Concept of Dot-com, its boom and bust. |

(*Continued*)

**Table 11.1** | Continued

| | |
|---|---|
| 2002 | Launch of Web 2.0 |
| 2004 | Launch of FaceBook and use of Internet for social networking |
| Future | Semantic Web (Web 3.0) |

## 11.2 HTML

HTML stands for hypertext markup language, widely used for the development of Web pages. It is used to display text, images and other objects in a specified way on the browser. Hypertext signifies the technique of linking HTML documents together. The link available on a Web page is called hypertext. Markup language signifies the marking of a text document with tags, which are used to display on Web browsers. It was designed in 1991 by Berners-Lee, and later, it was published in 1995 with standard specifications. The latest version of HTML-5 was published in 2012. World Wide Web Consortium (W3C) defines the specifications of HTML. HTML 4.0 was published in 1997 and then revised in 1998. HTML 4.01 was published in 1999.

HTML files are written in Notepad, a text editor on Windows, with a file extension. html. The output can be shown by opening it using a Web browser.

### 11.2.1 HTML Tags

HTML tags are used to mark up HTML elements and are surrounded by two characters '<' and '>' known as angle brackets (see Table 11.2). HTML tags generally comes in pair such as <b> and </b>. The first tag in a pair is the opening tag and the second tag is the ending tag. The text between the start and end tags is the element content. These are not case-sensitive, which means <b> and <B> are the same.

A typical HTML document has the following structure:

```
<html>

<head>

                Document header related tags

</head>

<body>

                 Document body related tags

</body>

</html>
```

**Table 11.2** | HTML tags

| Tag | Description |
|---|---|
| <html> | Defines an HTML document |
| <head> | Defines the document's header, which can keep other HTML tags such as <title>, <link>, etc. |
| <title> | Defines the document title |
| <body> | Defines the document's body |
| <h1> – <h6> | Defines headings 1 to 6 |
| <p> | Defines a paragraph |
| <br> | Inserts a single line break |
| <center> | Defines a centering content |
| <hr> | Defines a horizontal rule |
| <pre> | Preserves formatting |
| <!--> | Defines a comment |
| <samp> | Defines a sample computer code |
| <var> | Defines a variable |

### 11.2.2 HTML Attributes

Attributes are extra bits of information in tags. These are used to define the characteristics of an HTML element. These show inside the opening tag and their values are always inside quotation marks and consist of two parts: name is the property and value is the value of the property to be set. For example, `<body bgcolor="blue">`, bgcolor is the name and blue is the value.

#### *11.2.2.1 Core Attributes*

There are four core attributes used in HTML: id, title, class and style.

1. The id attribute can be used to uniquely identify any element within an HTML page. For example, `<p id="html">` This paragraph explains the definition of HTML.
2. The title attribute is used for the title of the element. The syntax of title attribute is same as that of the id attribute.
3. The class attribute is used to associate an element with a style sheet, and specifies the class of the element.
4. The style attribute is used to specify Cascading Style Sheet (CSS) rules within the element.

#### *11.2.2.2 Generic Attributes*

| Attribute | Value | Description |
|---|---|---|
| align | Right, left, centre | Horizontally aligns tags |
| bgcolor | Numeric, hexadecimal, RGB values | Places a background colour behind an element |
| background | URL | Places a background image behind an element |
| class | User defined | Classifies an element for use with Cascading Style Sheets |
| height | Numeric value | Specifies the height of tables, images or table cells |
| id | User defined | Names an element for use with Cascading Style Sheets |
| title | User defined | Pop-up title of the elements |
| valign | Top, middle, bottom | Vertically aligns tags within an HTML element |
| width | Numeric value | Specifies the width of tables, images or table cells |

### 11.2.3 HTML Elements

Elements are bits of information that make up Web pages. These reside in the tag pairs, for example, "My first Web page" and "This is my first Web page" are HTML elements.

```
<html>

   <head>

      <title>My first Web page</title>

   </head>

      <body>This is my first Web page</body>

</html>
```

### 11.2.4 HTML Character Entities

In HTML, some characters have special meaning; if we want to display such characters, character entities must be used in place of actual characters. A character entity has three parts:

ampersand (&), entity name or an entity number and semicolon (;).

&and; specifies the beginning and ending of a special character, whereas Entity name is an abbreviation of that character.

For example, '<' is a beginning tag in HTML. To display this character we have to write '&lt;'. There are some more examples listed in Table 11.3.

**Table 11.3** | HTML Character Entities and their Description

| Character | Character Entity | Description |
|---|---|---|
|  | &nbsp; | Non-breaking space |
| < | &lt; | Less than |
| > | &gt; | Greater than |
| & | &amp; | Ampersand |
| " | &quot; | Quotation mark |
| ' | &apos; | Apostrophe |

### 11.2.5 HTML Formatting

HTML formatting tags are basically designed to display special type of text. The tags will provide some additional feature than the normal text. Some of the formatting tags used in HTML are shown in Table 11.4.

**Table 11.4** | HTML Formatting tags and their description

| Tag | Description |
|---|---|
| <b> | Bold text |
| <big> | Larger text |
| <del> | Deleted text |
| <div> | Grouping content |
| <i> | Italic text |
| <ins> | Inserted text |
| <pre> | Preformatted text |
| <small> | Smaller text |
| <span> | Grouping content |

*(Continued)*

**Table 11.4** Continued

| Tag | Description |
|---|---|
| <strike> | Strike text |
| <sub> | Subscript text |
| <sup> | Superscript text |
| <tt> | Teletype (monospaced) text |
| <u> | Underlined text |

**Note**: The <font> tag in HTML is deprecated

## 11.2.6 HTML Phrase Tags

The purpose of designing phrase tags is to add structural information to a text fragment. They are displayed in a similar way as other basic tags like <b>, <i>, <pre>, and <tt>, etc. The phrase tags used in HTML and their description is given in Table 11.5.

**Table 11.5** HTML phrase tags and their description

| Tag | Description |
|---|---|
| <abbr> | Text abbreviation |
| <acronym> | Acronym element |
| <address> | Address text |
| <bdo> | Text direction |
| <blockquote> | A long quotation |
| <cite> | Text citation |
| <code> | Computer code text |
| <dfn> | HTML definition element, introducing a special term |
| <em> | Emphasize text |
| <kbd> | Keyboard text |
| <mark> | Marked text |
| <q> | A short quotation |
| <strong> | Strong text |
| <small> | Smaller text |

Programming variables are used when the content of the element is a variable; in most cases, the <pre> and <code> tags are also used alongwith it.

Program output is used while documenting line(s) of code, also used to check the output of the code or script.

## 11.2.7 HTML Background

The background can be a colour or an image and specified in the <body> tag.

### *11.2.7.1 Bgcolor*

Bgcolor is an attribute and specifies a background colour for an HTML page. The value of this attribute can be a hexadecimal number, an RGB value or a colour name.

### *11.2.7.2 Background*

It specifies background image for an HTML page. The value of this attribute is the uniform resource locator (URL) of the image.

## 11.2.8 HTML Lists

HTML lists are of three types: unordered lists, ordered lists and definition lists.

1. Unordered lists are used for non-sequential lists (bullets). <ul> … </ul> tag is used to define unordered lists and <li> tag is used to define each list item.
2. Ordered lists are used for sequential lists (incremental numbers). <ol> … </ol> tag is used to define ordered lists.
3. Definition list consists of two parts: a termand a description. <dl> tag is used to define a definition list; <dt> is used for definition term and <dd> is used for definition description.

## 11.2.9 HTML Links

An anchor tag <a> … </a> is used to define a link, the destination is also included in the anchor tag <a>, for example, `<a href="http://www.html.com"> HTML </a>`, href is an attribute.

## 11.2.10 HTML Images

<img> tag is used to display image. It does not have a closing tag.

1. **src attribute:** It has an src attribute, which contains URL of the image. For example, `<imgsrc="html.gif">`.
2. **alt attribute:** It also uses alt attribute to define an alternate text for an image.
3. **Image dimensions:** Width and height of an image can be used with <width> and <height> attributes.
4. **border attribute:** This attribute is used to specify border thickness in terms of pixels.

## 11.2.11 HTML Tables

The HTML tables are used to arrange data into rows and columns of cells. These are created using the <**table**> tag.

<**tr**> tag is used to create table rows and <**td**> tag is used to create data cells. <**th**> tag is used for table heading. Cellpadding and cellspacing attributes are used to adjust the white space in the table cells. Cellpadding defines the distance between cell borders and the content within a cell and cellspacing defines the width of the border. Colspan and rowspan attributes are used to merge two or more columns into a single column and row, respectively. For table backgrounds, bgcolor and background attributes are used. The height and width of the table can be changed with the height and width attributes, respectively. Caption tag is used for the caption of the table. <thead>, <tbody> and <tfoot> are used for creating portions of HTML table into header, body and footer, respectively.

### 11.2.12 HTML Frames

To divide the browser window into multiple sections, HTML frames are used. A separate HTML document can be loaded in each section. A frameset is simply a collection of such kind of frames in a browser window.

<frameset> tag is used to create frames in the browser window. The rows attribute is used for horizontal frames and cols attribute is used for vertical frames. The value of rows and cols is represented in percentage, for example, `<frameset cols="25%,50%,25%">`.

Each frame is indicated by the <frame> tag. The <frame> tag has mainly two attributes: name and src. For example, `<frame name="top" src="/html/top_frame.htm"/>`.

Other attributes of a frame are border, frameborder and framespacing. Border defines the width of the border of each frame in pixels. Frameborder takes value either 1 (yes) or 0 (no) for 3D border display. The framespacing defines the amount of space between frames in a frameset.

The <frame> tag has the following attributes: src, name, frameborder, marginwidth, marginheight, noresize, scrolling and longdesc.

The <iframe> tag is used to define a rectangular region, that can display another document having scrollbars and borders.

### 11.2.13 HTML Forms

These are used to collect data from the site visitor. There are various form elements: text fields, drop-down menus, combo-box, radio buttons, checkboxes, etc.

The <form> tag has the following attributes: action, method, target and enctype.

1. Action attribute is used for backend script, which is ready to process passed data.
2. Method attribute is used to upload data.
3. Target defines the target window, where the result of the script is displayed, and
4. Enctype defines the encoding of the data by the browser.

### 11.2.14 HTML Marquees

HTML marquee is a scrolling piece of text displayed on the Web page. The <marquee> tag is used for this.

## 11.3 CASCADING STYLE SHEETS

Cascading Style Sheets (CSS) is another method of styling content. It is a style sheet language which describes the appearance and format of the document written in any markup language. Generally, it is used to modify the style of web pages and user interfaces written in HTML and XHTML, plain XML, XUL, etc. Along with HTML and JavaScript, the CSS is a technology used by most websites to create attractive webpages, visually appealing user interfaces for web and mobile applications.

CSS is designed primarily to distinguish document content from document presentation, using elements such as the layout, colours, and fonts.

## 11.4 XML

XML stands for eXtensible Markup Language. It is an open standard to share data and information between computers and computer programs. It is a meta-markup language, which defines the syntax to define other semantic, domain-specific and structured markup languages. It also describes the structure and meaning of a document. It is a set of rules for defining semantic tags. XML documents always have a single root element. The element names are case-sensitive, always closed and correctly nested. Attributes of element are always quoted. The default entities defined are <, >, &, “, and ’. An XML document has a tree-like structure.

Syntax for an XML:

```
<Element Name>

<Element Name>Content</Element Name>

<Element Name>Content</Element Name>

<Empty Element Name/>
```

### Example 11.1

<html> is a root element, <head> and <body> are two branches.

```
<html>

<head>

<title>XML</title></head>

<body>

This is XML</body>

</html>
```

### Example 11.2

<team person1=“Tom” person2=“Dick”>

```
<team>

<person1>Tom</person1>

<person2>Dick</person2>

</team>
```

### Example 11.3

Example of an XML document:

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

<greeting>Hello, world!</greeting>
```

1. **Namespaces:** An XML namespace is a collection of names identified by a universal resource identifier (URI) reference.
2. **XML semantics:** It is an articulation of XML elements that exist in a file, their relationship(s) to each other and their meaning.

## 11.4.1 Advantages of XML

1. Defining your own elements
2. Better organized documents
3. Sorting of database

## 11.4.2 Document-Type Definition

Document-type definition (DTD) defines the tags in the document; the tags contain other tags, the number and sequence of the tags, the attributes of tags and the values of those attributes.

#### *11.4.2.1 Defining XML Vocabularies with DTDs*

The semantic relationships are created by DTD and/or XML schemas. DTDs reside inside an XML document or outside an XML document. If it resides inside, it begins with a DOCTYPE declaration followed by the name of the XML document's root element and a list of all the elements and their relation to each other. If it resides outside an XML document, then the DOCTYPE declaration includes a pointer to a file where the XML elements are described.

#### *11.4.2.2 Names and Numbers of Elements*

Each element is qualified by the number of times it can occur in the XML document. This is done with the asterisk (*), question mark (?) and plus sign (+) symbols. Each symbol has a specific meaning:

1. **Asterisk (*):** The element may appear zero or more times.
2. **Question mark (?):** The element may appear zero or one time only.
3. **Plus sign (+):** The element appears at least once if not more times.

#### *11.4.2.3 PCDATA*

PCDATA stands for parsed character data. It is used to denote that content contains only text, text without markup.

#### *11.4.2.4 Sequences*

An element contains multiple sub-elements when bound together; the list of multiple elements is called a sequence. They can be grouped together as follows:

1. **Comma (,):** Denotes the expected order of the elements in the XML file.
2. **Parentheses (()):** Groups elements together.
3. **Vertical bar (|):** Denotes a Boolean union relationship between the elements.

## 11.4.3 Tag Rules for XML Documents

An XML document is a collection of XML elements. An XML element starts from (including) the element's start tag and ends at (including) the element's end tag. The tags in XML should follow the following restrictions:

1. XML tags cannot overlap.
2. Tags should be closed.
3. A tag that does not contain any text can contain the end marker at the end of the start tag.

### 11.4.4 Types of XML Documents

There are two ways to create an XML document:

1. **Well formed:** These documents follow the XML tag rules and do not follow the DTD. Document has a top-level element, all elements must have a starting and an ending tag, element names should be case-sensitive and elements must be nested properly.
2. **Valid:** A valid document must have followed the rules by DTD and should be well formed.

The comparison of HTML and XML is given in Table 11.6.

**Table 11.6** Comparison of HTML and XML

| Attributes | HTML | XML |
|---|---|---|
| Internal Definition | Yes | Yes |
| External Definitions | No | Yes |
| Defined Tags | Yes | No |
| Add New Tags | No | Yes |
| Easily Add Objects | Yes | Yes |
| Easily Add Entities | Yes | Yes |

**Note:** XML and HTML were designed with different goals:

1. XML was designed to describe data and to focus on what data is.
2. HTML was designed to display data and to focus on how data looks.

### 11.4.5 XHTML

It stands for eXtensible HyperText Markup Language. It was developed by W3C for the transition from HTML to XML. It is similar to HTML4.01, but it gives a consistent and well-organised format. It combines the features of both HTML and XML.

The XHTML document use lower case for all HTML elements and attributes. Certain tags in HTML are non-pair tags but XHTML requires end tags. All values must be quoted. XHTML define three DTD: Strict, Transitional and Frameset. XHTML 1.1 is a module based XHTML, which supports Ruby Annotation elements for the better understanding of East-Asian languages. It is compatible with HTML 4.0.

### 11.4.6 Document Object Model (DOM)

It is a World Wide Web Consortium (W3C) standard, used to access documents such as HTML and XML. It is a programming API. It defines the objects, properties and methods for accessing documents. It has three parts: Core DOM, XML DOM and HTML DOM.

### 11.4.7 XUL

Developed by Netscape and Mozilla, It stands for eXtensible User interface Language and pronounced as “Zool”. It is a series of XML tags, which allows different operating platforms to exchange data or program. It supports Cascading Style Sheets, JavaScript, RDF, DOM and HTML.

### 11.4.8 Flash and Silverlight

Developed by Adobe Systems, Flash is a Rich Internet Application (RIA), used for playing audio and video in webpages. It is basically used for providing animation, interactivity to games, advertisements, etc. It is run on Microsoft Windows, Mac OS, QNX, Google TV and RIM.

Supported by a subset of. NET Framework, Silverlight is also a Rich Internet Application (RIA). In this, user interfaces are declared in eXtensible Application Markup Language (XAML). It supports Windows Media Video (WMV), Windows Media Audio (WMA), MPEG layer III (MP3).

### 11.4.9 User-Interface Language

These are used for graphical user interfaces and control. The objective of these interfaces is to use programs and script codes in the form of markup. Some of user interface markup languages are:

1. XUL (eXtensible User-interface Language) developed by Mozilla Foundation.
2. QML (Qt Meta Language) developed by Nokia, used for mobile applications such as touch input, fluid animation, etc. It is based on JavaScript declarative language.
3. UMIL (User Interface Markup Language) used to define user interfaces for computers. It is an XML based language.
4. UsiXML (User Interface XML) is a complaint markup language. It describes the user interface for multiple contexts of use.
5. XAL (eXtensible Application Language) developed by Nexaweb’s Enterprise Web 2.0 suite, used for applications like Java client and Ajax Client.

Other user-interface languages are as follows: MXML, OpenLaszlo, ZUML, JavaFX, jInterface, Thinlet, Vexi, XHTML, XFDL, Xforms, XAML, XRC, EMML, GladeXML, SVG, etc.

## 11.5 BASIC CONCEPTS OF CLIENT–SERVER COMPUTING

Client–server consists of two parts: server, which provides services and client, which request services from the server (Fig. 11.1). The server in the network provides services to more than one client. There are different servers for different applications, such as file server, print server, mail server, etc. So, a client can request services from several servers on the network. Client and server can be on the same computer or on different computers linked by a network.

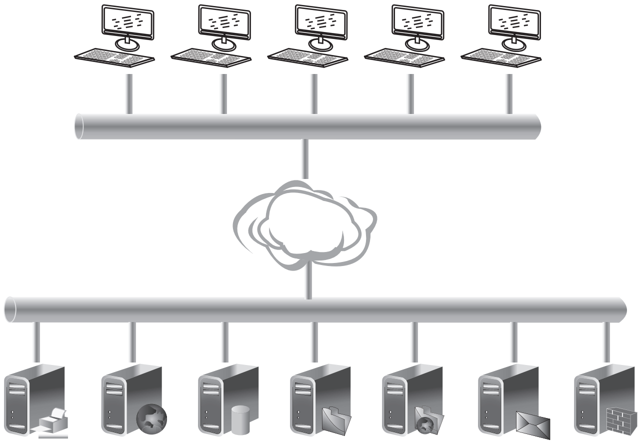

**Figure 11.1** | Client–server model.

### 11.5.1 Server Types

For large and complex systems, the responsibility of the servers is distributed to several servers.

1. **File server:** All files reside in the file server. It is responsible for sharing files on the network. Example: Map Drive by Microsoft Windows, Google Drive, Network File Services (NFS) by Sun Micro Systems.
2. **Print server:** All the output devices, such as printer, are controlled by the print server. It is responsible for sharing printers on the network among clients.
3. **Application server:** It manages access to centralized application software. Example: shared database.
4. **Web server:** It manages Internet and/or Intranet. It shares documents. Thin clients use Web browser to request data and documents from servers. Example: Internet Information Server (IIS) of Microsoft Corporation, WebLogic and Oracle Application Server, etc.
5. **Mail server:** It manages the e-mails, messages, etc. Example: Gmail.
6. **Fax server:** It manages sending and receiving of fax.
7. **Database server:** It manages databases and responds to clients through SQL queries. It shares data in a database. Example: Oracle9i database server.
8. **Transaction servers:** It manages data and remote procedures. It shares data and high-level processing, such as OnLine Transaction Processing (OLTP), across a network. Example: Microsoft SQL Server, BEA Tuxedo, etc.

### 11.5.2 Stateless and Stateful Servers

1. **Stateless server:** It treats each request as an independent transaction. The request is not related to the previous request. It increases the overhead in each request, that is, extra information has to be included. Though, it simplifies the server design and does not affect the system if clients crash down. Example: Gopher protocol and Gopher+ protocol.
2. **Stateful server:** It does not treat each request as an independent transaction. Each request is related to the previous request. In stateful server, if clients crash frequently, state information may exhaust server's memory. Example: A remote file server.

### 11.5.3 Thin Client and Fat Servers

1. **Thin client:** It handles minimum processing on the client side.
2. **Fat client:** It handles large processing on the client side. It is used in traditional client–server models.
3. **Fat server:** It handles more functions on the server side. It is easy to manage than fat clients.

### 11.5.4 Functions of a Client

1. It manages user interface.
2. It accepts and checks the syntax of user inputs.
3. It generates database request and transmits them to the server.
4. It passes response back to the server.

### 11.5.5 Functions of a Server

1. It checks authorization for a client.
2. It ensures that a client does not violate integrity constraints.
3. It performs query/update processing and transmits responses to the client.
4. It maintains system catalogue.
5. It accepts and processes database requests from a client.

6. It provides concurrent database access.
7. It provides recovery control.

### 11.5.6 Topologies for Client−Server

Topology provides a physical connectivity of all clients and servers to each other. There are three types of topologies designed for client−server as follows:

1. **Single client single server:** In this topology, one client is directly connected to one server (Fig. 11.2). Example: One dedicated server connected to a single remote-sensing satellite.

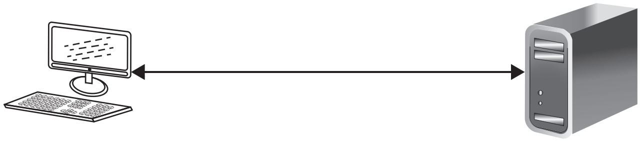

**Figure 11.2** Single client single server.

2. **Multiple clients single server:** In this topology, multiple clients are directly connected to one server (Fig. 11.3). Example: Different nodes connected to a server in a lab.

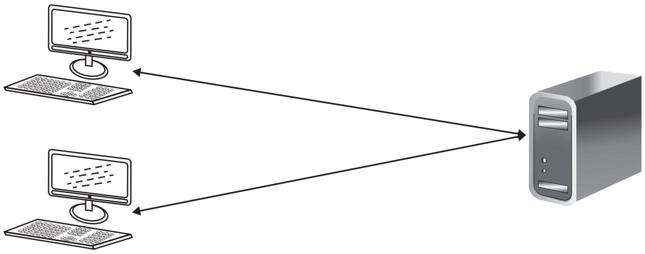

**Figure 11.3** Multiple clients single server.

3. **Multiple clients multiple servers:** In this topology, multiple clients are connected to multiple servers (Fig. 11.4). Example: People accessing facebook, Google and other applications.

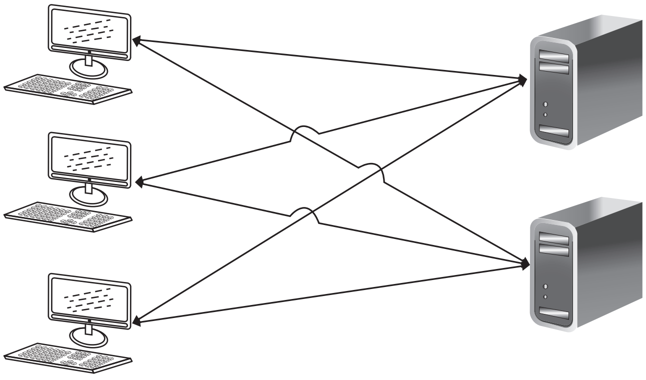

**Figure 11.4** Multiple clients multiple servers.

### 11.5.7 Types of Client−Server Model

There are three types of client−server models:

1. Two tier
2. Three tier
3. *N* tier

#### *11.5.7.1 Two-Tier Architecture*

In this, the business rules exist either at the client or at the server. The client is the first tier and the server is the second tier. Two-tier architecture at the client is called "fat client". The client uses a driver to translate the client's request into a database native library call.

**Advantages**

1. Client requests services directly from the server.
2. It is less complicated for implementation.
3. It provides attractive graphical user interface applications.
4. It has persistent connection between the client and the server.
5. It can be easily managed.

**Disadvantages**

1. Maintenance cost of application at client is high.
2. It increases load in the network.
3. Several PCs are required for individual applications.
4. Software distribution procedure is complicated in two-tier architecture.

#### *11.5.7.2 Three-Tier Architecture*

In this, business logic is located in the middle tier. Client's request is also handled by the middle tier. Only middle tier has to change if business rules are changed. The driver translates the request into a network protocol and makes a request via the proxy server. Application responsibilities in the three-tier architecture are of the following three types:

1. **Presentation (GUI) or user services:** It maintains the graphical user interface and generates output for the monitor. Presentation logic deals with screen formatting, windows management, input editing and what-if analysis.
2. **Application services or business rules:** It executes applications and controls the flow of the program. Business logic dealing with domain and range validation, data dependency validation and request/response architecture of inter-process communication level.
3. **Database services or data server:** It manages the databases. Server logic deals with data access, data management, data security and SQL parsing.

**Advantages**

1. Centralized application maintenance
2. Separate application logic from user interface control and data presentation
3. Easy and dynamic load balancing
4. Less expensive hardware (thin client)
5. Clients need not have native libraries loaded locally
6. Centralized drivers

**Disadvantages**

1. Client does not maintain a persistent database connection.
2. Separate proxy server is required. It increases network traffic.
3. Network protocol used by the driver may be proprietary.

#### *11.5.7.3 N-Tier Architecture*

In this, the developer designs components according to business rules and distributes the functionalities accordingly. It has better utilization and sharing of resources. The *N*-tier architecture provides finer-grained layers.

**Advantages**

1. Improved overall performance
2. Centralized business logic
3. More secure

### 11.5.8 Merits and Demerits of Client−Server

The client−server architecture facilitates with various features but it has some disadvantages also. The merits and demerits are discussed below:

1. **Merits:**
   - It increases the performance and reduces the workload on each system.
   - It gives workstation independence, which does not bind the users to a single operating system.
   - It allows different types of component systems such as client, network or server to work together.
   - It provides scalability in the network.
   - It preserves data integrity.
   - It provides data sharing, so data accessibility is good to users.
   - Due to centralized management of data, it becomes easy to secure and backup recovery of data by system administration.
   - It provides sharing of resources among diverse platforms.
   - It provides location independence of data processing.
   - The operating cost is less.
   - It has reduced hardware cost in terms of storage, printers and other output devices.
   - It has less communication cost.
2. **Demerits:**
   - Maintenance cost is high as it needs administrator. So, it also increases the training cost.
   - Hardware cost is increased in terms of high-powered platform with large amount of RAM and storage space.
   - It is more complex than PC.

## 11.6 J2EE PLATFORM

J2EE (Java2 Platform Enterprise Edition) is used to reduce cost involved in application design and development. It provides distributed multi-tiered application model, reusable components, web-service support and transaction control applications.

J2EE applications are made up of components. There are basically three types of components:

1. **Application clients and applets:** These components run on the client.
2. **Java Servlet and JavaServer Pages (JSP) technology components:** These are the web components that run on the server.
3. **Enterprise JavaBeans (EJB) components (enterprise beans):** These are the business components that run on the server.

### 11.6.1 J2EE Services

1. **HTTP API:** It is defined by the servlet and JSP interfaces.
2. **HTTPS:** It supports HTTP protocol over the SSL protocol.
3. **JDBC:** It is used for connectivity with relational database systems.
4. **JMS (Java Message Service):** It supports reliable point-to-point messaging as well as the publish-subscribe model.
5. **JTA(Java Transaction API):** It consists of two parts. (i) Application-level demarcation interface, which is used by the container and application components to demarcate transaction boundaries, and (ii) interface between the transaction manager and a resource manager used at the J2EE SPI level.
6. **JAF (Javabeans Activation Framework):** It is used for handling data in different MIME types, originating in different formats and locations.
7. **JNDI (Java Naming and Directory Interface):** It is used for naming and directory access.

8. **JAXP (Java API for XML Parsing):** It provides support for the industry standard SAX and DOM APIs for parsing XML documents. It also supports XSLT transform engines.
9. **RMI-IIOP:** It is a sub-system composed of APIs to use RMI-style programming. It supports both the J2SE native RMI protocol (JRMP) and the CORBA IIOP protocol.

### 11.6.2 J2EE Architecture

The Java2 SDK is an Enterprise Edition known as J2EE SDK. It is the reference implementation provided by Sun Microsystems, Inc. Figure 11.5 shows the major elements of the J2EE architecture.

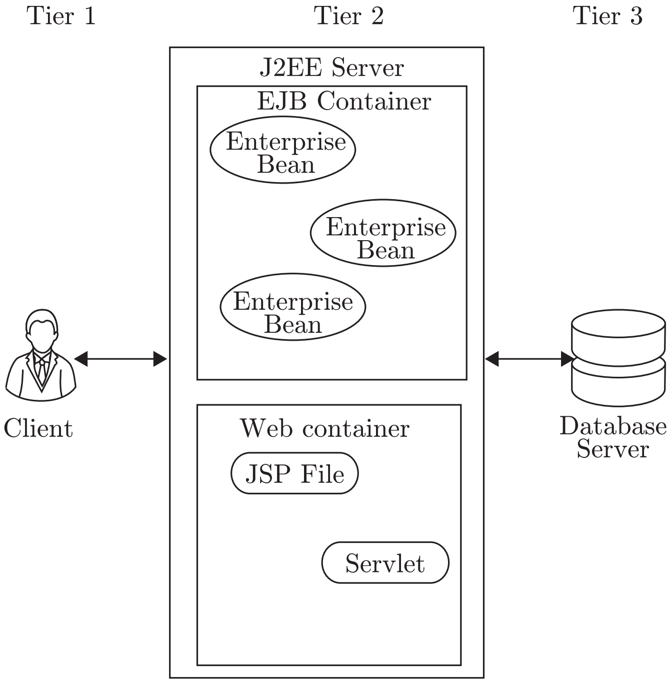

**Figure 11.5** J2EE architecture.

### 11.6.3 EJB Container

The container is a runtime environment, which controls the enterprise beans and provides them system-level services. Enterprise bean instances runs in an EJB container.

The container provides the following services to enterprise beans:

1. **Transaction Management:** The programmer need not to code for the transactions, rather, he has to declare the enterprise bean's transactional properties in the deployment descriptor file. Then, the container reads the file and handles the enterprise bean's transaction.
2. **Security:** Only authorized clients can invoke an enterprise bean's methods. Client has a particular role and each role is permitted to invoke certain methods.
3. **Remote Client Connectivity:** It manages the low level communication between clients and enterprise beans.
4. **Life Cycle Management:** It maintains all the states from the creation to the removal of enterprise beans.
5. **Database Connection Pooling:** The container manages the pool of database connections. The enterprise bean obtains a connection from the pool. These database connections can be re-used.
6. **Web Container:** It is a runtime environment for JSP files and servlets.

### 11.6.4 J2EE Application Server

It provides the following services:

1. **Naming and Directory:** It allows programs to locate services and components through the Java Naming and Directory Interface (JNDI) API.
2. **Authentication:** It enforces security by requiring users to log in.
3. **HTTP:** It enables web browsers to access servlets and JavaServer Page (JSP) files.
4. **EJB:** It allows clients to invoke methods on Enterprise Java Beans.

## IMPORTANT FORMULAS

1. XML is case-sensitive.
2. XML is strict, so all the tags should be properly closed.
3. A tag that doesn't contain any text can contain the end marker at the end of the start tag.
4. Important tags in HTML:

| | |
|---|---|
| <html> | Defines an HTML document |
| <head> | Defines the document's header, which can keep other HTML tags such as <title>, <link>, etc. |
| <title> | Defines the document title |

| | |
|---|---|
| <body> | Defines the document’s body |
| <h1> – <h6> | Defines headings 1 to 6 |
| <p> | Defines a paragraph |
| <br> | Inserts a single line break |
| <center> | Defines a centering content |
| <hr> | Defines a horizontal rule |
| <pre> | Preserves formatting |
| <!--> | Defines a comment |
| <table> | Defines table in html |
| <tr> | Define rows in table |
| <td> | Defines columns in table |

## Important Abbreviations

| | |
|---|---|
| API | Application Program Interface |
| ASP | Active Server Pages |
| CORBA | COmmon Resource Broker Achitecture |
| CSS | Cascading Style Sheets |
| DOM | Document Object Model |
| DTD | Document Type Definition |
| EJB | Enterprise Java Beans |
| HTML | HyperText Markup Language |
| HTTP/HTTPS | HyperText Transfer Protocol / Secure |
| J2EE | Java 2 Enterprise Edition |
| JAF | Javabeans Activation Framework |
| JMS | Java Message Service |
| JNDI | Java Naming and Directory Interface |
| JSP | Java Server Pages |
| JTA | Java Transaction API |
| MathML | Mathematical Markup Language |
| MIME | Multi-purpose Internet Mail Extensions |
| OLTP | OnLine Transaction Processing |
| RIA | Rich Internet Application |
| RSS | Rich Site Summary or Really Simple Syndication |
| SGML | Standard Generalized Markup Language |
| SVG | Scalable Vector Graphics |
| UIML | User Interface Markup Language |
| URI | Uniform Resource Identifier |
| URL | Uniform Resource Locator |
| WML | Wireless Markup Language |
| XAL | eXtensible Application Language |
| XAML | eXtensible Application Markup Language |
| XHTML | Extensible HyperText Markup Language |
| XML | Extensible Markup Language |
| XSL | eXtensible Stylesheet Language |
| XUL | XML User Interface Language |

## SOLVED EXAMPLES

**1.** The firewall is used for

(a) Sensing the size of the packet.
(b) Isolating intranet from extranet.
(c) Screening packets to/from the network and filtering of network traffic.
(d) Isolating Internet from virtual LAN.

*Solution:* Firewall is used to prevent unauthorized access of private networks through packet screening from outside user, especially Internet.

Ans. (c)

**2.** LDAP stands for

(a) lightweight digital audio protocol.
(b) lightweight directory access protocol.
(c) large domain access protocol.
(d) large data audio protocol.

*Solution:* The LDAP stands for Lightweight Directory Access Protocol which is a directory service protocol based on client-server model.

Ans. (b)

**3.** A digital signature is used to provide security makes use of

(a) Digitally scanned signatures.
(b) A unique ASCII code number of the sender.
(c) Private key encryption.
(d) Public key encryption.

*Solution:* Digital signature is an asymmetric cryptography technique which uses public key encryption mechanism to provide integerity, authentication and non-repudiation.

Ans. (d)

**4.** The <b> tag makes the enclosed text bold. The other alternative is to use

(a) <strong> (b) <dark>
(c) <big> (d) <highblack>

*Solution:*
<strong> tag is used to define important text by bold the text.
<dark> — there is no tag in html named dark.
<big> tag is used to make text bigger than the normal text.
<highblack> — there is no tag in html named highblack.

Ans. (a)

**5.** What does HTML stand for?

(a) High tool and text markup language
(b) Heavy tool markup language
(c) Hypertext markup language
(d) Hypertext my language

*Solution:* HTML is a markup language which is used to design web pages. It stands for Hypertext Markup Language.

Ans. (a)

**6.** Who describes and controls the making of various Web standards?

(a) The World Wide Web Consortium
(b) IEEE and IETE
(c) Administrator and Mozilla
(d) IEEE

*Solution:* The World Wide Web Consortium (W3C) is the international standard organisation for managing and controlling web standard. IEEE and IETE are professional association societies for advancement of technology.

Ans. (a)

**7.** Which of the following HTML tag will provide the largest heading

(a) <h6> (b) <h1>
(c) <head8> (d) <heading6>

*Solution:* <h1> will provide largest heading, then <h2>, then <h3> and so on. <h6> gives the smallest heading.

Ans. (b)

**8.** The following HTML tag is used for inserting a line break?

(a) <br/> (b) <startbreak/>
(c) <lb/> (d) <LINEBRK/>

*Solution:* <br/> tag is used for inserting a line break. There is no tag in html with names<startbreak>, <lb/> and <LINEBRK>.

Ans. (a)

**9.** The correct syntax for adding yellow as a background colour in HTML is

(a) <bodystyle="background-color:yellow">
(b) <backgroundcolor>yellow</background color>
(c) <color.background="yellow">
(d) <backgrndcolor="yellow">

*Solution:* Syntax for background color is:

**1.** <body bgcolor="color_name|hex_number| rgb_number">

**2.** <body style="background-color:color_ name">

Any of these two can be used.

Ans. (a)

**10.** The HTML tag used to make a text bold is

(a) <b> (b) <heavybold>
(c) <blackbold> (d) <bld>

*Solution:* <b> tag is used in HTML to make text bold.

Ans. (a)

**11.** The HTML tag used to make a text italic is

(a) <italic> (b) <i>
(c) <textitalic> (d) <slantingtext>

*Solution:* <i> tag is used in HTML to make text italic. There is no tag in html with names<italic>, <textitalic> and <slantingtext>.

Ans. (b)

**12.** The HTML tag used to create a hyperlink is

(a) <ahref="http://www.myexams.com"> MYEXAMS</a>
(b) <aurl="https://www.myexams.com"> myexams</a>
(c) <anamelink="http://www.myexams.com"> myexams.com</a>
(d) <a>http://www.MYEXAMS.COM</a>