

Unit-I

History of Internet

- * Internet, then known as ARPANET (Advanced Research Project Agency Network), was a project of US Defence Department. It was brought online in 1969 & initially connected four major computers at universities in southwestern US.
- * In 1970s, they created emails which became so popular.
- * By collaborations of computer science network, which was funded by NSF (National Science Foundations), they standardized TCP/IP which permitted world wide Interconnection of Networks.
- * In end of 1980s, researchers speed up the internet upto 45 Mbps. Then in 1990s, commercial internet service providers (ISPs) emerged and Internet became fully commercialized.

Basic Internet Protocols

- * A protocol is a set of rules, that is used ~~for~~ ^{to} communicate applications to each other.
- * Different types of protocols used in web are:-
 - (i) HTTP (HyperText Transfer Protocol)
 - (ii) HTTPS (HyperText Transfer Protocol Secure)
 - (iii) ICMP (Internet Control Message Protocol)
 - Used by networked computer OS to send error message
 - (iv) OSPF (Open shortest Path First)
 - Dynamic routing protocol for use in IP (Internet Protocol) Networks
 - Builds routing tables bases solely on destination IP address found in IP packet.
 - Support VLSM (variable Length Subnet Masking)
 - (v) RIP (Routing Information Protocol)
 - Dynamic routing protocol used in LANs & WANs using Bellman Ford algorithm.
 - (vi) IGRP (Internet Gateway Routing Protocol)
 - Overcome limitations of RIP, used by routers to exchange routing data within an autonomous system.
 - (vii) TCP/IP (Transmission Control Protocol / Internet Protocol)

(viii) UDP (User Datagram Protocol).

- Data transmission but less reliable

(ix) FTP (File Transfer Protocol)

(x) MTP (Multimedia Transfer Protocol)

(xi) TelNet → Remote Connections

(xii) SMTP (Simple Mail Transfer Protocol)

(xiii) POP3 (Post Office Protocol version 3)

(xiv) MIME (Multipurpose Internet Mail Extensions)

- Extended format of email to support text other than ASCII, non-text attachment etc.

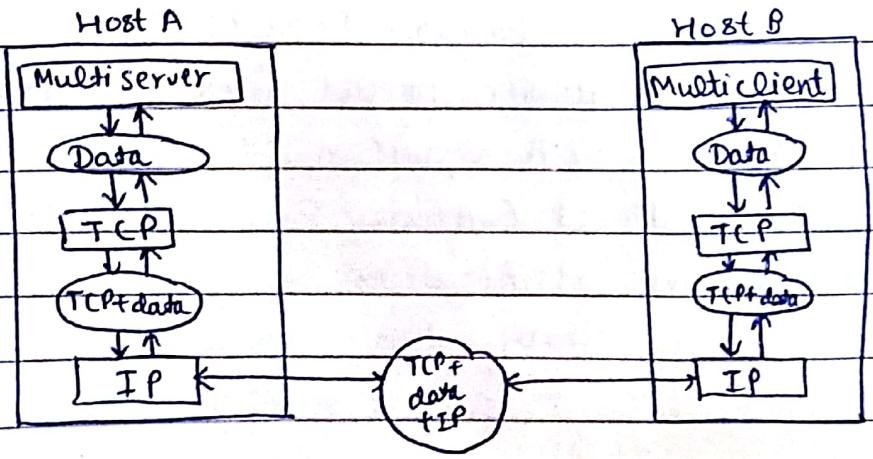
World Wide Web (www)

- Tim Berners-Lee is known as the father of www.
- W3C (World Wide Web Consortium) is the international standards organisation for the World Wide Web.
- www is an information space where the documents and other web resources are identified by URL (Uniform Resource Locator) & interlinked by HyperText links & can be accessed via Internet.
- The webpages are initially in text documents & ~~contains~~ associated with HTML which has text, documents, images, videos, audio & software components.

HyperText Transfer Protocol (HTTP)

- HTTP is ~~a~~ set of rules for transferring files like text, images, audio, video & other multimedia files on www.
- HTTP is an application protocol that runs on TCP/IP protocols.

Basically HTTP communication is to follow request-respond model.



IP working

- HTTP connections standard port is port 80

- HTTPS connections standard port is port 443

Markup Languages - HTML

- HTML purpose is to define and specify the specific elements that make up a page such as body of the text, heading, paragraphs, line breaks, text elements & so on.

Basics of XHTML:

→ ^{syntax} Similar to HTML syntax but with some restrictions ^{as case-sensitive}

→ XHTML is case sensitive markup language.

Ex:- Abc ⇒ Invalid

 Abc ⇒ valid

→ Each & every XHTML tag should have an equivalent closing tag, even empty elements should also have closing tags.

Ex:- < img src = "abc.jpg" > ⇒ Invalid

 ⇒ valid

→ All XHTML documents must have a ~~doctype~~ "DOCTYPE" declaration at the start.

→ All attributes of XHTML must be quoted.

Ex:- < img src = "abc.jpg" width=250 height=50 > ⇒ Invalid

 ⇒ valid

→ XHTML doesn't allow attribute minimization.

Ex:- <option selected> ⇒ Invalid

<option selected = "selected"> ⇒ valid

→ The id attribute replaces the name attribute in XHTML.

Ex:- ⇒ Invalid

 ⇒ valid

→ The language attribute of the script tag is deprecated in XHTML.

Ex:- <script language = "javascript" type = "text/javascript">
document.write ("Hello");
</script>

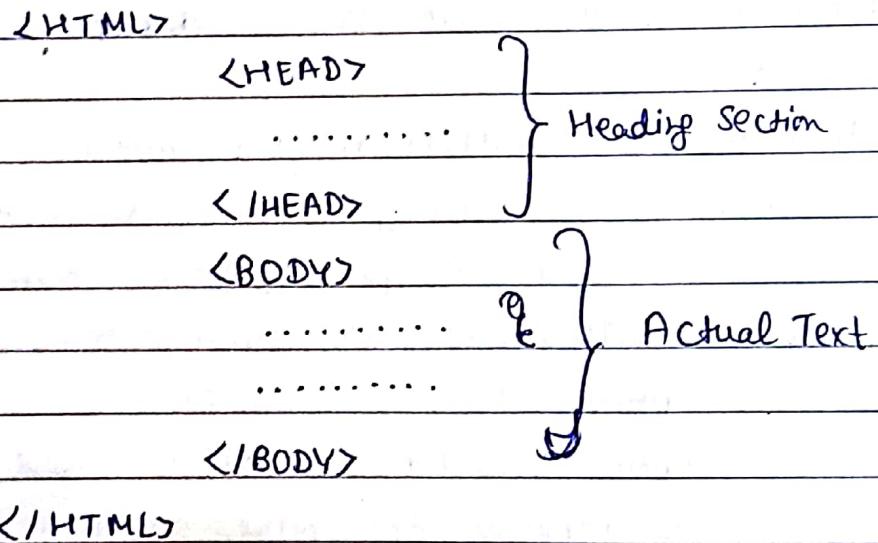
<script type = "text/javascript">
document.write ("Hello");
</script>

→ All the XHTML tags should be nested properly
Ex:- `<i> Hello </i>` ⇒ Invalid
`<i> Hello </i>` ⇒ Valid

- HTML elements

- (i) Load any Editor ex- Notepad
 - (ii) Create the HTML file & save it, with extension .htm or .html
 - (iii) view if any browser ex- Internet Explorer etc.
- Tag or Tag element refers to HTML codes that define the element in a HTML file such as heading, image etc.
- Two kinds of tags:
- (i) Container tag : These tags ~~are~~ in which bracket contain text or other tag elements are called container tags.
 - (ii) Empty tag : These are standalone & do not bracket or contain text or any other tag elements.

→ Skeleton of HTML document is:



- HTML tags

- (i) `<HEAD>` → To identify heading or title of a document
 - (ii) `<TITLE>` → To identify text that appears in title of browser window. Not exceed 60 characters.
 - (iii) `<BODY>` → ~~Contains~~ text that will show up on webpage
 - (iv) Heading Tags → To display headings on webpage in various sizes
- `<H1>` - - - `<H6>`
- Largest → Smallest

(iv) <P> - To give paragraph
<P ALIGN="value"> value = center, left, right

(vi)
 - To give line break

(vii) **** - To change text style specifications

Attributes: - To change
For style For size For colour of text

value1 = Arial, Courier, Impact, Garamond etc.

$$\text{value 2} = 1, 2, 3,$$

1 = 8 pt 2 = 10 pt 3 = 12 pt

$$4 = 14 \text{ pt}$$

value 3 = Red, green, blue etc or $\#FF0000$, $\#00ff00$, $\#0000ff$ etc.

(viii) **L87** - To bold the text

(ix) **L1** - To italicize the text

(X) `<DIV>` - For applying alignment & style characteristics to only a section of a document.

(xj) <!-- ... --> - For comments

(xii) <PRE> - (Preformatted Text) To display a block of "Preformatted" text in a monospace, fixed-pitch font.

(xiii) Lists → - For ordered lists

└ - For unordered lists

Attributes: <OL type="value1" - start = "value 2">

<LI value="x"> ...

value 1 = 1, i, I, a, A

value 2 = (2,3,...) (ii,iii,...) etc

so $x = (8, 9_{\text{etc}}) \quad (\text{vii etc})$

Attribut: <ul type="value">

 ...

value = circle, square, disc

(x iv) Tables in HTML

a) ~~<table>~~ <TABLE> </TABLE>

Attributes: <TABLE Align="left/right/center" border="1/2/3... -"

cellpadding = "1/2/-" cellspacing = "1/2/... width = "1/2")

(space b/w content & inside of cell)

(space b/w cells)
in table

⑥ `<TR> . . . </TR>` → Table row

Attributes: `<TR bgcolor="red/green/blue/...">`

⑦ `<TD> . . . </TD>` → Table data

Attributes: `<TD align="...">`

`colspan="..." (no. of columns)`

for a table `rowspan="..." (no. of rows)`

`<TABLE>`

`<TR>`

`<TD> . . . </TD>`

: :

`</TR>`

`</TABLE>`

⑧ `<CAPTION> . . . </CAPTION>` → Used for title/caption of table

Attributes: `<CAPTION ALIGN="...">`

⑨ `<TH> . . . </TH>` → Used to designate cell to be header cell of
used with `<TR>`

Ex:- `<TABLE>`

`<CAPTION> . . . </CAPTION>`

`<TR>`

`<TH> . . . </TH>`

: :

`<TR>`

`<TD>`

`... </TD>`

:

`<TR>`

`<TD>`

`... </TD>`

:

`<TR>`

`<TD>`

`... </TD>`

:

`</TABLE>`

(XV) Frames in HTML:

→ Allow coder to present documents in multiple view which may
be independent to window or sub window

`<frameset rows/cols = " , " >`

`<frame src = "value" />`

`name = "value"`

} source of frame content
src

`scrolling = "value"`

} name to frame window,

`marginwidth = "value"`

} scrolling bars or not
value = "yes/no/auto"

`marginheight = "value"`

} set left & right margins

`marginheight = "value" />`

} set top & bottom margins

`</frame>`

`</frameset>`

→ `<frameset>` can be used instead of `<body>`. If use ~~with~~ `<body>` with `<frameset>`, it will be effective to frames & also to `<head>`

→ advantages: ① users have multiple pages in same web browser

② keeps one part of page static while other part of page can be used.

③ Reduce server load

→ disadvantages: ① search engine may not deal with it properly
② hard to navigate through pages
③ difficult to print content of all frames.

(xvi) Forms in HTML:

→ Form is a group of various elements just like exam form. It is a collection of form fields on a page.

→ Classified into three sections:

① Form Header ② Input fields ③ Action Buttons

`<FORM>`

Form elements

`</FORM>`

→ Input elements:

`value1 = Text` → for textbox

`<FORM ACTION = "onClick" method = "get/post" >`

Button → custom push button

`<INPUT Type = "value1" >`

checkbox → on/off checkbox

`Name = " "`

Image → graphical image

`Value = " "`

Password → masked text box

`Align = " "`

Radio → Radio Button

`Size = " "`

Reset → Reset button

`maxLength = " "`

Submit → Submit button

`</FORM>`

→ <Text area rows=" " cols=" " wrap=" " > → for
text area
WRAP = "off / virtual / physical"
no word wrapping word wrapping, text sent at wrap points
(Multiple lines text input)

• <Textarea Read Only> abc </Textarea>
↳ To display abc ~~as~~ by default

→ submit button to send data from data fields to server
→ reset button to clear data from data fields

- Defining XHTML's Abstract Syntax

→ All XHTML documents must have ~~doctype~~ declaration at start.

→ Case sensitivity

→ Closing tags

→ Attributes Quotes

→ Attribute Minimization

→ id attribute

→ language attribute

→ nested attribute

→ Element prohibitions

Element

prohibition

(a) <a> must not contain other <a> elements

(b) <pre> must not contain , <object> <big> <small> <sub> <sup>

(c) <button> " " " <input> <select> <textarea> <label> etc

(d) <label> " " " other label elements

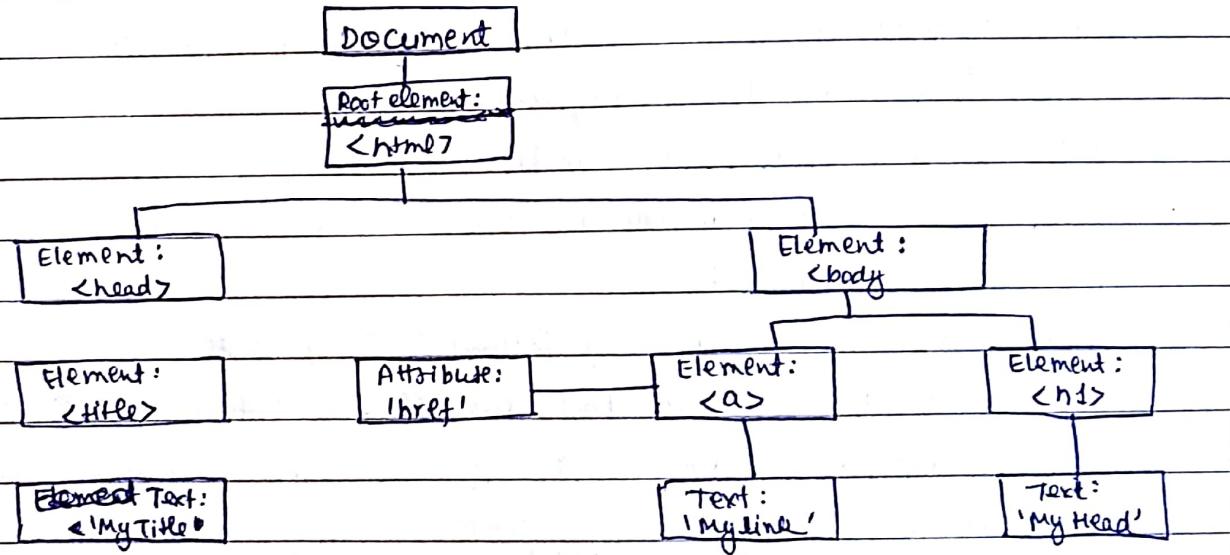
(e) <form> " " " form "

→ A minimal XHTML document example:

```
<?xml version="4.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//IETF//DTD XHTML 1.0 Transitional//EN"
  "abc.dtd">
<html xmlns="abc" xml:lang="en" lang="en">
  <head>
    <title> ABC </title>
  </head>
  <body>
    ...
  </body>
</html>
```

- Defining HTML Documents

- When a webpage is loaded, the browser creates a Document Object Model (DOM) of the page.
- The HTML DOM is a W3C standard & defines a standard set of objects for HTML & a standard way to access & manipulate HTML documents.
- All elements, along their text & attributes, can be accessed through the DOM. The contents can be modified or deleted, & new elements can be created.
- The HTML DOM is platform & language independent. It can be used by any programming language like Java, JavaScript & VBScript.
- HTML DOM model is constructed as tree of objects



- Within the object model, JavaScript gets all the power it needs to create dynamic HTML.

CSS Stylesheets

- CSS Syntax Introduction

- Style sheet or CSS (Cascading Style Sheet) is term used to describe a mechanism that applies a style across one or more webpages. The style of CSS refers to colour schema, layout & other strategies for organising the visual layout of the components of a document.
- The work on CSS began at CERN Lab in 1994.

→ HTML is used for structure content of CSS is used for formatting structured content.

→ Advantages of CSS:

(i) Improve / Increase search engine results by crawling of search engine directly to important contents.

(ii) Site loading speed is fast, i.e., site takes less memory & loads fast.

(iii) Designing is separate from content so helps in more control over a website for designer.

(iv) Great flexibility in presenting contents because different style can be used.

(v) A common style code & use the same class help to use less code that means less bandwidth consumption & easy to maintain.

(vi) Maintenance is easy & keep more than one stylesheet & use them based on requirements.

(vii) The name cascading indicates that we can use more than one style & the priority is given to local style first.

- CSS Core Syntax:

→ A StyleSheet consists of following Components:

(i) Style Rule (ii) Including ~~set~~ Style Information InLine - Inline Styles

(iii) Embedding style sheet (iv) Grouping style rules (v) Selectors

→ Style Rule

↳ It is a ~~collection of~~ set of HTML tags specifying the ~~formatting styles~~ which can format elements

↳ Syntax:

selector { property: value }
 |
 declaration

ⓐ selector determines which element you are working with

ⓑ property = attributes like font type, size etc

ⓒ value = setting for the attribute

→ We can insert a stylesheet in three ways:

(i) InLine style

(ii) Internal stylesheet

(iii) External stylesheet

→ Inline Styles - Including Style Information Inline

Syntax: <HTML tag style="Property : value">

Ex:- <body style="Background-Color : #FF0000">

- Advantages: (a) useful for small quantities of style definitions
(b) can override other style methods at local level.

Disadvantages: (a) can't control multiple documents

- (b) selector grouping method can't be used to create complete elements.

(Embedding style sheet)

→ Internal Stylesheet

Syntax: <style type="text/css">

selector {

 property : value;

}

</style>

Ex:- <head> <style type="text/css">

 body { background-color : #000000;
 }

</style>

</head>

→ External Style Sheet

Ex:- <head> ~~style~~

~~<link href="styleSheet" type="text/css"~~

 href = "style.css" >

</head>

Advantages: (a) Can control style for multiple documents at once

- (b) classes can be created for use on multiple HTML element types.

- (c) Selector & grouping methods can be used to apply styles under complex contents.

Disadvantages: (a) an extra download required to import style info.

- (b) Rendering of document may be delayed until style sheet is loaded.

→ Grouping Style Rules

Ex:- `<style>` H1, H2 { color: Red; font-family: Arial }
`</style>`

→ Selectors:

(a) Simple selectors

Ex:- H1 { color: blue }

(b) HTML selector.

use names of HTML elements without brackets.

Ex:- `<style>` P { font-style: italic }
`</style>`

(c) Class selector

Gives authors the ability to apply styles to specific parts of a document & don't necessarily to whole document

Syntax: `<style>`
 (class selector)
 class name class selector { property: value }
`</style>`
`<body>`
`<p class = "class name">`

class attribute
`<body>`

(d) id selector

Like class selector, id is used in place of class

Syntax: `<style>`
 (ID selector name)
 # ID selector name { Property: value }
`</style>`
`<body>`
`<p ID = "ID selector name">`
`</body>`

Ex:- `<style>` #control { color: Red }
`</style>`

`<body>` `<p H2 (Id = "Control")>` Abc `</H2>`
`</body>`

(e) Contextual Selector

Ex:- H1 { color: red }
 I { color: red } } } Team id → H1, I { color: red }

- CSS Text properties

- (a) Text-Letter-spacing → Space b/w letters
 - (b) Word-spacing → Space b/w words
 - (c) vertical-align → vertical positioning of text & image w.r.t baseline
 - (d) Text-align → Text alignment
 - (e) Text-indent → Margin text lowercase
 - (f) Text-transform → Transformation of text like capitalize, ~~bold~~ etc
 - (g) Text-decorate → Text decoration like bold, blink, underline etc
 - (h) Text
- Color & Background properties
- (a) Color
 - (b) Background-color
 - (c) Background-image
 - (d) Background-position

→ Font properties

- (a) family
- (b) color
- (c) style
- (d) align
- (e) size

→ Border properties

- (a) width
- (b) color
- (c) style

- CSS Box Model

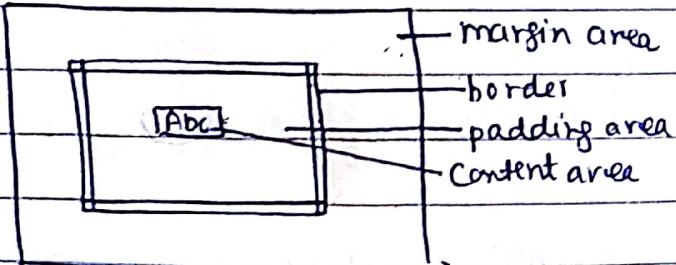
→ Block style elements such as `<p>` element can be considered as rectangular boxes on the screen. These box properties include:

(a) Margin-properties

(b) Border properties:

- (i) * Border-style
- (ii) * Border-width
- (iii) * Border-color

(CSS Box model
+
Normal Flow
Box Layout)



Other properties of box model:
(i) content area
(ii) padding area

- Normal Flow Block Box Layout

→ The space b/w an element's border & its content can be

Specified in four padding regions, can be set using padding-top, padding-right, padding-left, ~~padding~~ padding-bottom properties.

→ Microsoft Specific Style Sheet Properties

filter: filter-name (filter-value1, filter-value2, ...)

↓
alpha, blur, chroma, dropshadow, flipH, flipV, glow, grayscale,
invert, Light, mask, shadow, wave, X-ray

— CSS Other properties (Lists, tables)

(i) Lists

↳ In HTML, two types of list (ordered & unordered)

↳ CSS ~~ordered~~ list properties allow us to:

- ① set different list item markers for ordered & unordered lists
- ② set image as list item marker
- ③ add background colour to list & list items

Ex:- `ul { list-style-type: circle; }`

`ol { list-style-type: upper-roman; }`

(ii) Tables

↳ To specify table borders in CSS, use border property

Ex:- `table, td, th {`

`border: 1px solid black;`

`}`

↳ To specify width & height in CSS, use width & height

Ex:- `table { width: 100%; }`

`th { height: 50px; }`

DHTML

- DHTML is the art of combining HTML, javascript, DOM & CSS.

- DHTML is not a language or web standard.

- DHTML stands for dynamic HTML.

- It is a new tool for designers to create webpages

with special effects & animations.

- DHTML is about using Javascript of HTML DOM to control, access & manipulate HTML elements.

- It helps to:

- (a) make web experience faster & more interactive for end user.
- (b) CSS provides absolute ^{positioning} of every object on a webpage
- (c) requires four independent components to work: HTML, CSS, DOM & Scripting Language

XML (Extensible Markup Language)

- Introduction

→ XML is the markup language derived from SGML. An XML document is written using the markup language to contain structured information.

→ It is called extensible because it is not in fixed format like HTML.

→ XML is a metalinguage that means a language that describes other language which allows you to design your own customized markup language.

→ XML is not a markup language but a language for defining markup language

HTML

* Pre-defined tags

* Missing end tags are permitted

* Attributes don't require quotes " "

* Tolerates non nested tags

* Not case-sensitive tags

XML

* User-defined tags

* Requires matching end tags.

* Attributes require all quotes " "

* ~~Tolerates~~ Tolerates nested tags.

* Case-sensitive tags

- XML document and vocabulary

```
Ex:- <?xml version = "1.0"?>  
<home>  
  <head> <title> My HomePage </title>  
  <banner source = "tobanner.gif"/>  
  </head>  
  <body>  
    <main.title> Welcome to my HomePage </main.title>  
    <rule/>  
    <para> Page under construction </para>  
  </body>  
  <footer source = "foot.gif"/>  
</home>
```

(i) XML declaration

Identifies what follows it is XML code

```
<?xml version = "1.0"?>
```

(ii) Root Element

Each XML document must have only one root element & all other elements must be enclosed in ~~the~~ that element.

In above ex:- home is the root element

(iii) Empty Element

Special case in XML if ~~this~~ is empty & has no content

In above ex:- rule is empty element

rule(), empty tag close delimiter

(iv) Attributes

Element tags can include one or more optional or mandatory attributes that give further information about elements they delimit.

Syntax for specifying an attribute:-

```
<element.type.name attribute.name = "attribute.value">
```

In above ex:- banner & footer have attributes

- Types of information in XML files

- (i) Processing : Used to pass an instruction to application that is processing the files.
- (ii) Comment : To give comment in XML files
- (iii) XML element : To present a unit of information with name, optional body & optional attributes
- (iv) XML entities : Special sequence to represent reserved HTML characters

& &
> >
< <
' '
" "

- XML syntax rules

- (i) Element must have closing tags
- (ii) XML tags are case-sensitive
- (iii) Opening & closing tags must be written:
- (iv) XML elements must be properly nested
- (v) XML attribute value must be in quotes

- XML processing instructions

These instructions are at the beginning of the file :

(i) version

Required attribute that specifies version of XML standards

(ii) encoding

Optional attribute that specifies the character encoding schema used in XML file

(iii) standalone

Optional attribute that specifies that the XML file is self contained or not.

- XML documents

dtd (document type definition)

- ↳ It is a language that can be used to define document type of SGML (Standard General Markup Language) document
- ↳ XML is subset of SGML & dtd is used to define a document type of XML.
- ↳ All elements & attributes & entities in XML document must be meet in dtd document
- ↳ XML documents are of three types:
 - (i) invalid ⇒ Document that do not follow tag rules
 - (ii) well formed ⇒ Document that follow tag rules
 - (iii) valid ⇒ Document that follow XML tag rules as well as rules defined in dtd.

- XML versions & declarations

(i) XML schema → XML based alternative to DTD. It describes structure of XML document

→ Use ~~XML~~ XML syntax

→ Are Extensible as written in XML

→ ".xsd" extension of XML schema file

XML document with ~~referenced~~ DTD

<?xml version="1.0"?>

<!DOCTYPE note SYSTEM

"URL/note.dtd">

<note>

<to> Abc </to>

<from> xyz </from>

<heading> Reminder </heading>

<body> Don't forget me. </body>

</note>

XML document with ^{reference to} XML schema

<?xml version="1.0"?>

<note xmlns="URL"

xmlns:xsi="URL"

xsi:schemaLocation="URL/note.xsd"

<to> Abc </to>

<from> xyz </from>

<heading> Reminder </heading>

<body> Don't forget me </body>

</note>

→ (b) The `<schema>` element is root element of every XML schema

`<?xml version="1.0"?>`

`<xss: schema>`

:

:

`</xss: schema>`

→ `<schema>` element may contain some attributes.

`<xss: schema xmlns:xss="URL/XMLSchema"`

namespace location { target Namespace = "URL" }

`xmlns = "URL"` } default namespace

Elements in this schema must be namespace qualified { element form Default = "qualified" }

(ii) Naming Rules of objects in XML

① Name consists of atleast one letter a to z or A to Z

② If name consists more than one character, it may start with underscore (-) or colon (:)

③ Initial letter can be followed by one or more character & special symbols & special characters

(iii) Entity Declaration

`<!ENTITY name "replacement text">`

every time string frame ; appear in your XML code, XML processor will automatically replace it with replacement text.

(iv) Working with elements & attributes

All markup declarations for XML document are entered b/w square brackets []

Syntax: `<!DOCTYPE document.type.name [internal.subset]>`

(v) Element Declaration

form: `<!ELEMENT name content>`

Ex:- `<!ELEMENT message (header, body)>`

`<!ELEMENT header (to, from, subject)>`

(vi) Empty Elements

form: `<empty.element />`

declaration: `<!ELEMENT empty.element EMPTY>`

- Introduction To WML (Wireless markup language)

- WML is in topmost layer in wap(wireless application protocol) architecture & is made up of wae(wireless application environment)
- wml is an application of xml (as defined in dtd).
- wml takes care of small screen & low bandwidth of transmission.
- It is the markup language defined in wap specifications.
- Wap sites are written in wml like webpages are written in HTML.
- extension of wml file is .wml & also support client side scripting
- wml use decks & cards.
- wml script
 - (i) client side scripting language based on ECMA script (which is based on Netscape's Javascript language), whose purpose is to provide client-side procedural logic
 - (ii) wmlscript is javascript of wireless world
 - (iii) wml script is function-oriented scripting language, which enhances wml applications
 - (iv) wml script has weak type definition (variables ^{types} not defined during declaration but determined during execution)
 - (v) wml script code maintained on server & execution takes place on local device.

Client side Programming

Unit - II

* Scripting language

- It is the simple base programming language designed to enable programmers to write useful programs quickly. It is similar to macro programs that tell a program how to perform a specific procedure.

- Advantages :

- (a) easy to learn & use
- (b) require minimum programming knowledge & experience
- (c) code running & editing is fast
- (d) allow complex task to be performed relatively in few steps
- (e) allow simple creation & editing in a text editor
- (f) allow ~~add~~ addition of dynamic & interactive activities in web pages
- (g) anyone can change & enhance in text editor.

- Disadvantages :

- (a) can be slower to run since they are interpreted & not compiled into machine code
- (b) can be harder to debug since no development environment is available
- (c) it is easy to modify & break since it is in text editor.

* Javascript

- It is a scripting language that is developed jointly by SUN & Netscape

- What Javascripts can do ?

- (a) gives HTML designers a programming tool
- (b) can put dynamic text into HTML page
- (c) can react to events
- (d) can read & write HTML elements
- (e) can be used to validate data
- (f) can be used to detect visitor's browser
- (g) can be used to create cookies

- Advantages: (i) speed (ii) simplicity

(iii) versatility (iv) server load (reduce demand on server)
(play nicely
with other languages)

- Disadvantages:
 - (a) Security
 - (b) Trust on end user

javascript

(i) interpreted, not compiled to byte code

(ii) object oriented, no mismatch b/w types of objects

(iii) codes integrated with HTML

(iv) variable datatypes, not declared

java

(i) compiles bytecode

(ii) class-based, objects are divided into classes

(iii) applets distinct with HTML

(iv) variable datatypes must be declared

* Javascript Basic Syntax

- To use script, <script> is used

Syntax: <script type="text/javascript">

<html>

<body>

<script type="text/javascript">

.....

</script>

</body>

</html>

- <Script> can be used anywhere (in head, in body or both)

- For external scripting, use src attribute

<script src="abc.js"></script>

* Variables, Literals & Data Types in Javascript

- Declare variable

var strname = some value

or

strname = some value

- assign value to variable

var strname = "abc" or strname = "abc"

- Lifetime of variable

When within funcⁿ, only accessed within funcⁿ. When exit funcⁿ, variable destroyed. \Rightarrow Local variables

* Javascript functions

Syntax:

```
function functionname(var1, var2, ..., varX)  
{  
    some code  
}
```

Ex:- `function display()`

```
{  
    alert("Hello World!");  
}
```

- Return statement is funcn to return value from func

* Javascript objects & properties

- Javascript is oop language so we can define own objects & make own variable types

- Properties are values associated with an object

Ex:- `<script type="text/javascript">`

`var txt = "Hello World!"`

`document.write(txt.length)`

`</script>`

Output: 12

~~txt~~ → string
txt → object
length → property of
txt object

NOTE: Semicolons are optional

- Methods are actions performed on objects

Ex:- `txt.toUpperCase()` in above example

Output: HELLO WORLD!

- String object used to manipulate stored piece of text

- Date object used to work with dates & times

`var myDate = new Date()`

`myDate.getFullYear(2018, 1, 1)`

`myDate.setDate(myDate.getDate() + 5)`

* Javascript Arrays

`var myArray = new Array()`

`myArray[0] = "Abc"`

`myArray[1] = "xyz"`

`myArray = new Array("Abc", "xyz")`

* Javascript Built-In Objects

(i) Boolean object
var myBoolean = new Boolean()

* (ii) Math object

Math.E, Math.PI, Math.SQRT2, Math.SQRT1_2, Math.LN2,
Math.LN10, Math.LOG2E, Math.LOG10E

* Javascript Form Programming

- Javascript can be used to validate input data in HTML forms before sending off the content to a server.

- Required fields

```
function validate_required (field, alerttxt)
{
    with (field)
    {
        if (value == null || value == " ")
        {
            alert (alerttxt);
            return false;
        }
        else
        {
            return true;
        }
    }
}
```

- E-mail validation

```
function validate_email (field, alerttxt)
{
    with (field)
    {
        apos = value.indexOf ("@");
        dotpos = value.indexOf (".");
        if (apos < 1 || dotpos - apos < 2)
        {
            alert (alerttxt);
            return false;
        }
        else
        {
            return true;
        }
    }
}
```

- Ex:- <html>

```
<head>
```

```
<script type="text/javascript">
```

```
function validate_email (field, alerttxt)
```

```
{
    with (field)
```

```
    apos = value.indexOf ("@");
    dotpos = value.indexOf (".");
    if (apos < 1 || dotpos - apos < 2)
```

```

    {
        alert (alerttxt);
        return false;
    }
}
```

```

else
{
    return true;
}
}
```

```

function validate_form (this.form)
{
    with (thisform) {
        if (validate_email (email, "Empty email") == false) {
            email.focus();
            return false;
        }
    }
}

</script>

<body>
<form action = "submit.htm" onClick = "validate_form (this); " method="post">
    Email: <input type = "text" name = "email" size = "30" >
    <input type = "submit" value = "Submit" >
</form>
</body>
</html>

```

* Javascript Intrinsic Event Handling

- Javascript interaction with HTML is handled through Event that occur when user or browser manipulates when the page loads
 events :- onClick, onSubmit, onFocus, onBlur, onChange, onLoad, onUnload etc

Timing events :- setTimOut(), clearTimOut()

* Javascript conditional statements

(i) if statement

```

if (condition)
{
    code to be executed
}

```

(ii) if...else statement

```

if (condition)
{
    code to be executed if condition true
}
else
{
    code to be executed if condition false
}

```

(iii) if...else if...else statement

```

if (condition)
{
    code to be executed if condition 1 true
}

```

else if (condition2)

{ code to be executed if condition 2 true
}

else

{
} Code to be executed if condition 1 & condition 2 false

(iv) switch statement

switch(n)

{ case 1:
execute code block 1
break

case 2:
execute code block 2
break

:
:

default:

code to be executed if different from cases defined above

}

* Operations in Javascript

(i) Arithmetic : +, -, *, /, %, ++, --

(ii) Assignment : =, +=, -=, *=, /=, %=

(iii) Comparison : ==, ===, !=, >, <, >=, <=

↳ if $x=5$ "x==y" true but $x==y$ false

(iv) Logical : &&, ||, !

(v) String : + (concat), or

(vi) Conditional : variable name = (condition) ? value 1 : value 2

* Javascript Popup boxes

- Dialog Boxes used to raise alert or get confirmation on any input from user side :

(i) Alert Box

(ii) Confirm Box

(iii) prompt dialog box

`alert("txt")`

* Javascript loops

(i) for loop

`for(var = start value; var <= end value; var = var + increment)`

Code to be ~~execute~~ executed

}

(ii) while loop

```
while (var <= end value)
{ code to execute
}
```

(iii) do... while loop

```
do
{ code to be executed
}
while (var <= end value)
```

(iv) Javascript break & continue statement

break → break loop

continue → break current loop & continue next value

(v) Javascript for... in statement

Used to loop (iterate) through elements of array or through properties of object

Syntax: `for (variable in object)
{
 code to be executed
}`

Ex:- `var mycar = new Array ("BMW", "Mercedes-Benz", "Audi", "Bugatti")
for (var x in mycar)
{
 document.write (mycar[x] + "
")
}`

* Javascript Document Tree

(Same as HTML DOM)

- Additional to that

- Within the object mode, javascript gets all the power it needs to create dynamic HTML:

- (i) can change all HTML elements in the page
- (ii) can change all HTML attributes in the page
- (iii) can change all CSS styles in the page
- (iv) can remove existing HTML elements & attributes
- (v) can add new HTML elements & attributes
- (vi) can react to all existing HTML events in the page
- (vii) can create new HTML events in the page

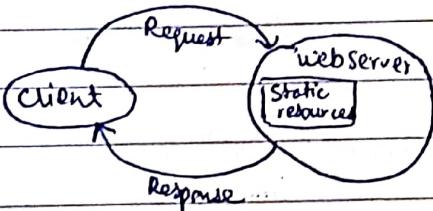
Server Side Programming

Java Servlets

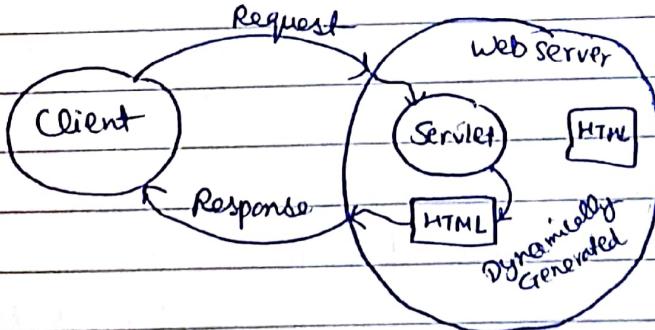
- * Servlet is a Java program that programmatically extends the functionality of a web server in request-response programming model.
- * Servlets provide a way of creating sophisticated server side extensions in a server as they follow the standard framework & use the highly portable java language
- * Servlet uses the classes in java package `javax.servlet`

`javax.servlet.http`

- * Servlet architecture { It includes
 - (i) servlet interface
 - (ii) request handling methods
- It is executed within web server. Hence, it doesn't have visible interface. They are responsible for generating dynamic contents in web applications
- When data or information doesn't differ from user to user we use static pages otherwise dynamic pages will be used.



For static contents



For dynamic content (Servlet)

- Current technologies for dynamic web application development:
 - a) Servlet / JSP — Sun Microsystems
 - b) ASP / ASP.NET — Microsoft
 - c) PHP — Open Source

Before these technologies CGI (Common Gateway Interface) was used which has following disadvantages over Servlets:

(i) Platform Dependency (ii) For each request a process is created

- Advantages of Servlets:

- i) Platform Independence
- ii) Extensibility
- iii) Performance
- iv) Safety
- v) Secure

- Characteristics of Servlet

- ⇒ (a) Servlets can use RMI, multi-threading, database connectivity etc.
- (b) Servlets are robust
- (c) Servlets are portable
- (d) Servlets are persistent

- Applet

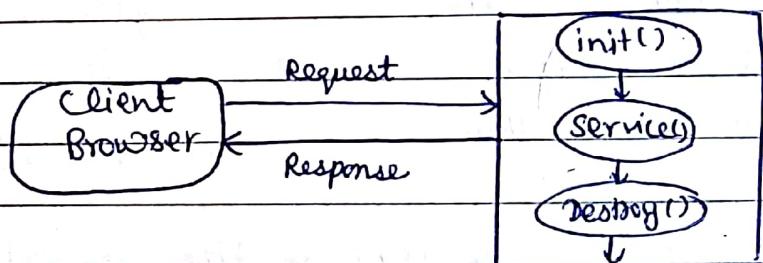
- (a) Java programs embedded in webpage
Need to download to client to open
byte code thus time consuming if
size too large

Servlet

- (a) executes on web server so
no overcome problem of download.
Dont require java enabled browser
unlike applets

* Life Cycle of Servlet

- A server is loaded only once in memory & is initialized in the init() method. After the servlet is initialized it starts accepting requests from the client and processes them through the service() method till it is shutdown by the destroy() method. The service() method is executed for every incoming request.



* Parameter Data

Few
- Methods: ~~Methods~~

- (i) `Servlet.init(ServletConfig config)` } When servlet is first loaded
throws `ServletException` } all info about code for servlet
- (ii) `Servlet.service(ServletRequest, ServletResponse)` } Receives all requests, identifies
request & dispatch them to `doGet()` or
`doPost()` for processing
- (iii) `Servlet.destroy()` } Clean up code for servlet
- (iv) `ServletResponse.getWriter()` } returns a reference to `PrintWriter` object
- (v) `ServerResponse.setContentType(String type)` } set type of content that
is to be sent

- Type of HTTP Requests

- (i) Get
- (ii) Post
- (iii) Head
- (iv) Put
- (v) Delete
- (vi) Trace etc.

get

post

- | | |
|--|--|
| 1) Conventionally for static resources | (i) for dynamic resources |
| 2) Request parameters applied to URL & sent as part of request header | (ii) Request parameters are sent as part of request body |
| 3) Maximum data that can be sent determined by size of header which has fixed size | (iii) Unlimited data can be sent as request parameters |

Servlet (interface)

↑
Generic servlet (Adapter)

↑
HttpServlet (Additional method to handle HTTP)

↑
Your service

- Application / Web Server

1.) Tomcat

Name of Servlet API JAR

servlet-api.jar

2.) web logic

weblogic.jar

3.) Sun Application Server

j2ee.jar

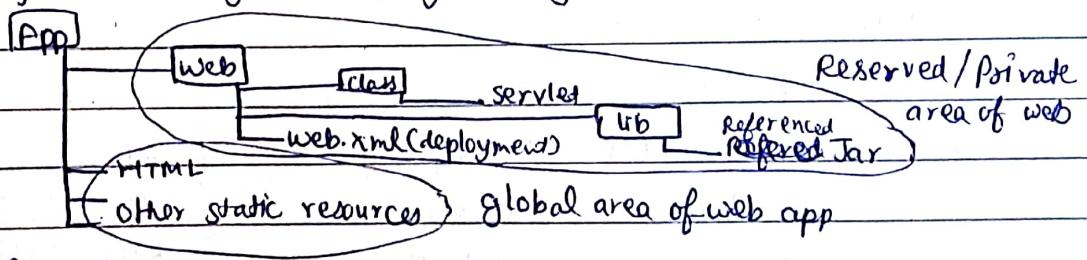
4.) IBM Web Sphere

j2ee.jar

5.) Macromedia JRun

jrun.jar

- Structure for creating & Deploying Servlet



TOP CHAR following

- ↳ In a web application, a servlet may have three names associated with it:
 - 1st Name used by client to send a request to servlet
 - 2nd Name used by deployer to map an incoming request to servlet class
 - 3rd Name of the servlet class

- Types of Deployment

1.) Hot Deployment:

War file containing web application copied to staging folder

(a) Packaged Format

(b) Exploded Format

2.) Cold Deployment:

Interface provided by server used to deploy application

(a) Packaged Format

(b) Exploded Format

→ Packaged Format: War files \Rightarrow If make change create new war file of disadvantage
deploy again advantage

→ Exploded Format: Production format \Rightarrow If make change no need to deploy again

Parameter

(a) Represents data sent as part of request

(b) Are immutable, i.e., value can't be changed or removed

(c) Are of String type

Attributes

(d) Represent data stored by servlet programmer to use in another servlet for processing same request

(e) Are mutable

(f) Are of object type

- Commonly used methods of Servlet Request Interface:

1.) getParameter()

5.) setAttribute()

2.) getParameterNames()

6.) removeAttribute()

3.) getAttribute()

7.) getInputStream()

4.) getAttributeName()

- Commonly Used methods of Servlet Response Interface:

1.) setContentType()

2.) getWriter()

3.) getOutputStream()

* Sessions

- HTTP is a "stateless" protocol which means each time a client retrieves a webpage, client opens a separate connection to web server.
- There are three ways to maintain session b/w web client & web server:
 - (i) Cookies?
 - (ii) Hidden Form FieldsIn HTML form, `<input type="hidden" name="sessionid" value="12345">`
- (iii) URL Rewriting
ex :- `http://www.abc.com/file.htm; sessionid=12345`

- Http session object

`Http session session = request.getSession();`

* Cookies

- Cookies are text files stored on client computer & they are kept for various information tracking purpose
- Three steps involved in identifying returning users:
 - (i) Server script sends a set of cookies to the browser.
 - (ii) Browser stores this information on local machine for future use.
 - (iii) When next time browser sends any request to web server then it sends those cookies to server & server uses that information to identify user.
- Cookies are usually set in a HTTP header
- If browser is configured to store cookies, it will then keep this information until the expiry date

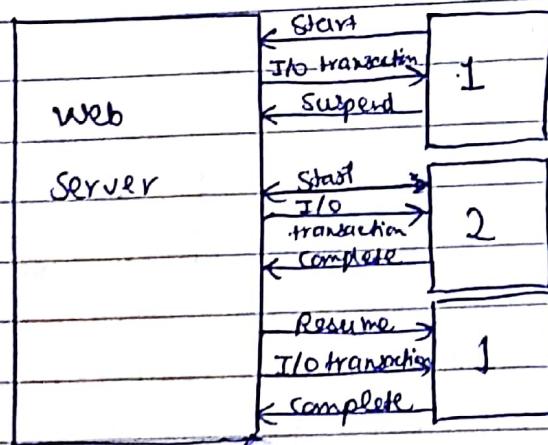
* Servlets Capabilities

- (i) Servlets for generating dynamic web pages
- (ii) Java Servlet engine (Request Respond model)
- (iii) Servlet implementation (`init()`, `service()`, `destroy()`, `ServletConfig()`)

* Servlets & Concurrency

- A java servlet container / web browser is typically multithreaded.
- So to make servlet thread safe:

- (i) servlet service() method should not access any member variables, unless these members ~~are~~ variables are thread safe themselves.
- (ii) servlet service() should not reassign member variables. If really need to reassign then it should be done inside a synchronized block
- (iii) rule (i) & (ii) also counts for static variables
- (iv) Local variables are always thread safe unless pointing to some shared object.

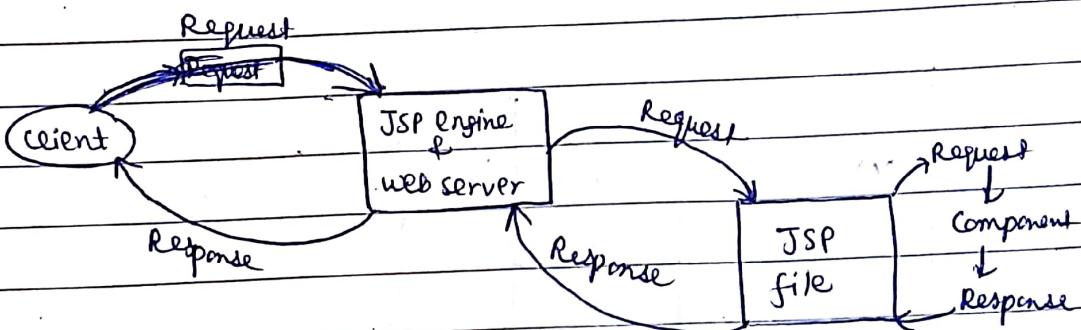


JSP (Java Server Pages)

- * JSP enables the creation of dynamic platform independent method (directly fetch from server)
- * JSP is a method building web based applications. It consists of ~~new~~ HTML tags & JSP tags
- * JSP pages are easy to maintain than servlet because we can separate designing & developing part
- * Features of JSP:
 - (i) Platform & Server Independence
 - (ii) Environment (Pure Java & OOP)
 - (iii) Extensible JSP tags
 - (iv) Reusability across platform
 - (v) Easier maintenance
- * Advantages of JSP:
 - (i) As JSP is an extension of servlet technology so we can use all features of servlet.

- (ii) JSP maintenance is easy because we can easily separate business (processing) & presentation (layout) logic.
 - (iii) * JSP requires less code than servlet.
- * JSP is part of Java platform, enterprise edition (J2EE), which is the java architecture for developing multilayer enterprise application

* Handling JSP page

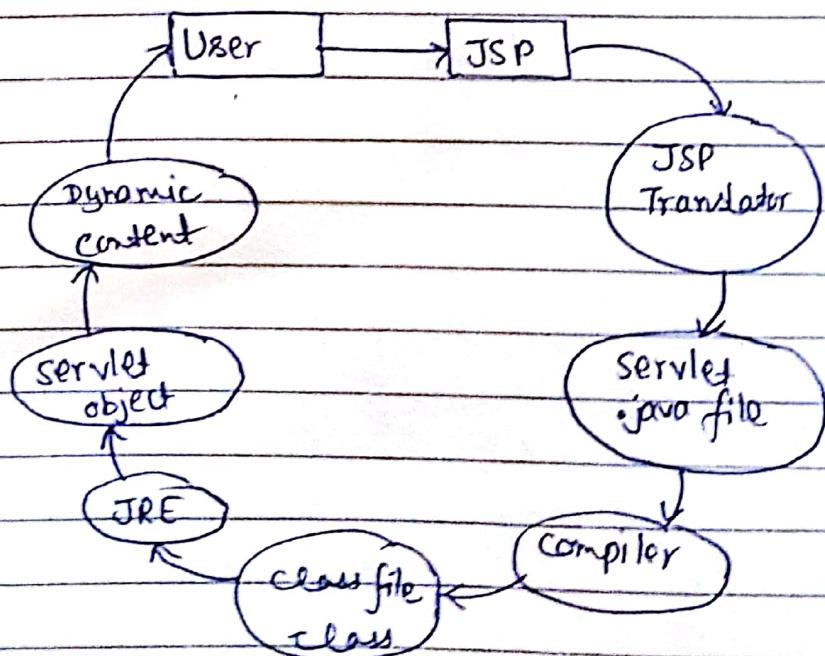
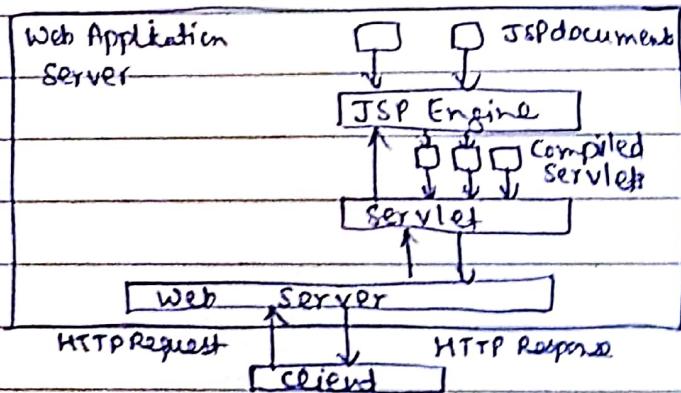


- * JSP syntax is similar to XML syntax. JSP pages are processed by web server while HTML pages are processed by browser.
- * JSP Directives are instructions for the JSP engine that are processed when the JSP page is translated into a servlet. They begin with `<%@` and end with `%>`. There are three types of directives:
 - 1.) page directives → no. of attributes that affect whole page
`<%@ page attributes %>`
 - 2.) include directives → To insert text & code at JSP translation time
`<%@ include file = "relative URL" %>`
 - 3.) taglib directives → Declare that page uses custom tags
`<%@ taglib uri = "taglibrary URL prefix" %>`
- * `<% code fragment %>` → Code fragment / scriptlets
- `<%@ directive %>` → Directives
- `<%! declaration %>` → Declarations
- `<% = expression %>` → Expressions
- `<%-- comment --%>` → Comments
 - ↳ HTML comment `<! -- Comment -->`
 - ↳ Hidden `<%-- Comment --%>`

* JSP scriptlet example:

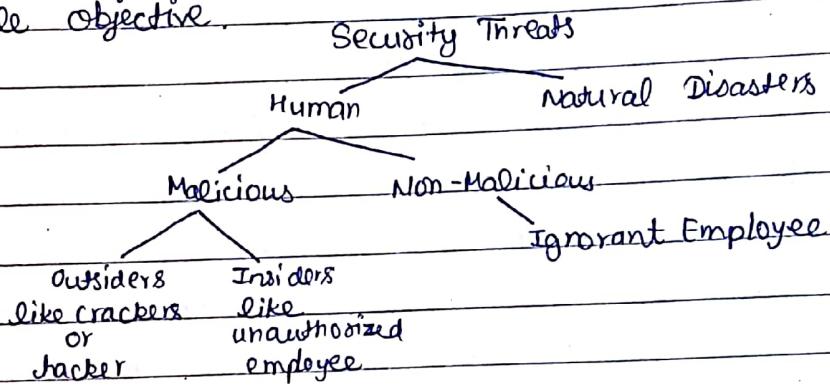
```
<html>
<head> <title>Hello </title> </head>
<body> %
<%
    int n=0;
    for (n=0; n<5; n++)
    {
        out.println("<h1>Hello World, " + </h1>");
    }
%
</body>
</html>
```

* JSP Lifecycle



Unit - III

- * A security threat is a possible danger that might exploit a vulnerability to breach security & then cause possible harm. So, the protection to an automated information system in order to attain the applicable objective.



- * The classification of threats could be :

- Physical Threats
- Accidental Error
- Unauthorized Access
- Malicious Misuse

- * In a computer, security risk is any event or action that could loss or damage to computer hardware, software, data or information.

- * various types of security risks are :

- (i) Injection Flaw
- (ii) XSS (Cross Site Scripting)
- (iii) ~~Insecure~~ Direct Object Reference
- (iv) Cross Site Request Forgery
- (v) Insecure Cryptographic Storage
- (vi) Restrict Url Access
- (vii) Unvalidated Redirect Page

- * Security Risk of a Site

- Two principal components, mail servers and mail clients support the email process
- Attacks on email system have taken different approaches, some attackers ~~are~~ with extensive knowledge of working of these systems exploit weaknesses of user systems to distribute viruses or other malwares throughout an organisation.

- Some sophisticated attacks have used e-mail^{to} compromise user workstation. Flaws in system have enabled unauthorized user to gain admin privileges & execute commands & install softwares on server.

- * Web Attacks & Their Prevention

- Mail servers & clients can be vulnerable to events such as
- ① ~~DoS~~ DoS Attacks (Denial of Services)

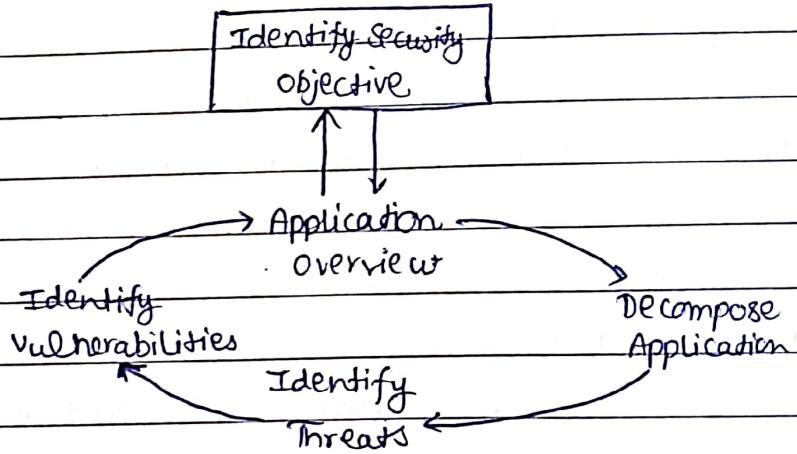
- (ii) Unencrypted connection b/w server & client
- (iii) Spam

(iv) Viruses & malicious code

(v) Sensitive Data Leak

- Prevention:

- (A) Plan Carefully & Address the Security Aspects of the Deployment of Server
- (B) Implement Appropriate Security Management Practices & Control to Assure that the Server is Maintained & Operated Securely
- (C) Ensure that the Server Operating System is deployed, configured & managed to meet the security requirements of the organization
- (D) Be sure that the Server application is deployed, configured & managed to meet the security requirements of the organization
- (E) Consider Implementing & using cryptography to protect user authentication and data.

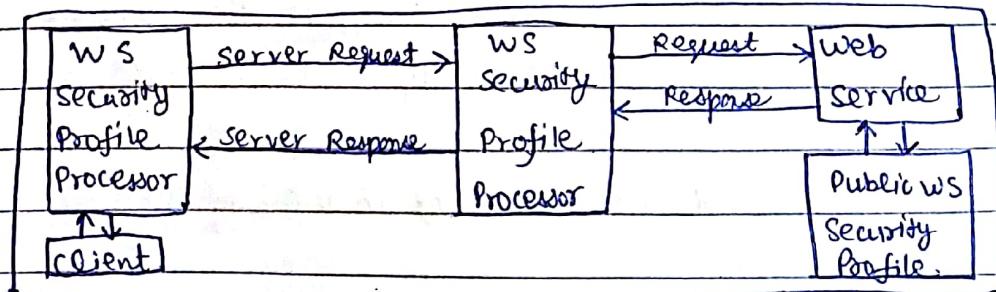


* Types of Threats:

- Worm : A self-replicating virus that does not alter files but duplicated itself & consumes system resources, slowing or halting other tasks.
- Logic Bomb : Programming code, inserted surreptitiously or intentionally, that is designed to execute (or "explode") under circumstances such as lapse of a certain amount of time or failure of program user to respond to program command. When exploded, may be designed to display or print a spurious message, delete or corrupt data, or have other undesirable effects.

- Trapdoor: Method of gaining access to some part of a system other than by the normal procedure.
- Trojan (Trojan Horse): Program in which malicious code is contained inside apparently harmless programming of data.
- RATs (Remote Admin Trojans): Special form of Trojan Horse that allows remote control over a machine.
- Malware: Any program or file that is harmful to a computer user.
- Spyware: Programming that gathers information about a computer user without permission.
- Mobile Malicious Code: Web documents often have server-supplied code associated with them which executes inside web browser & may provide a mechanism to attack systems running a client browser.
- Malicious Font: Webpage text that exploits default method used to de-compress Embedded Open Type fonts & these malicious fonts are designed to trigger a buffer overflow which will disable security on windows-based PCs.
- Rootkits: Set of software tools used by intruder to gain & maintain access to a computer system without user's knowledge.
- ~~Dos (Denial of Service)~~: Malicious attempt to make a server or network resource unavailable to user, usually by interrupting or suspending services of a host connected to Internet.
- DDoS (Distributed Denial of Service): Special Dos which uses many devices of multiple Internet connections, often distributed globally into botnet.

* Web Security Model



- It consists of:
 - ① Secure communication between client & server like HTTPS

① User authentication & session management . ex- cookies

③ Active content from different websites

④ web application security

⑤ website authentication

- Security Features:

① Authentication ② Authorization ③ Encryption ④ Auditing

- Point 8 of Attacker can Target:

① Shopper ② Shopper Computer ③ Network connection b/w Shopper & website server
④ website's server ⑤ Software vendor

- Attacks:

- ① Tricking the Shopper
- ② Snooping ~~Shopper~~ Computer
- ③ Sniffing network (Monitoring Data b/w ~~computer~~ ^{client} of server)
- ④ Guessing Passwords
- ⑤ Using ~~Do~~ DDoS Attacks

* Session Management

- Session refers to series of user application interactions that are tracked by the server. The session itself resides in the server.

- For each request, client transmits session ID ~~in~~ in a cookie or if browser does not allow cookies, server automatically writes session ID into the URL.

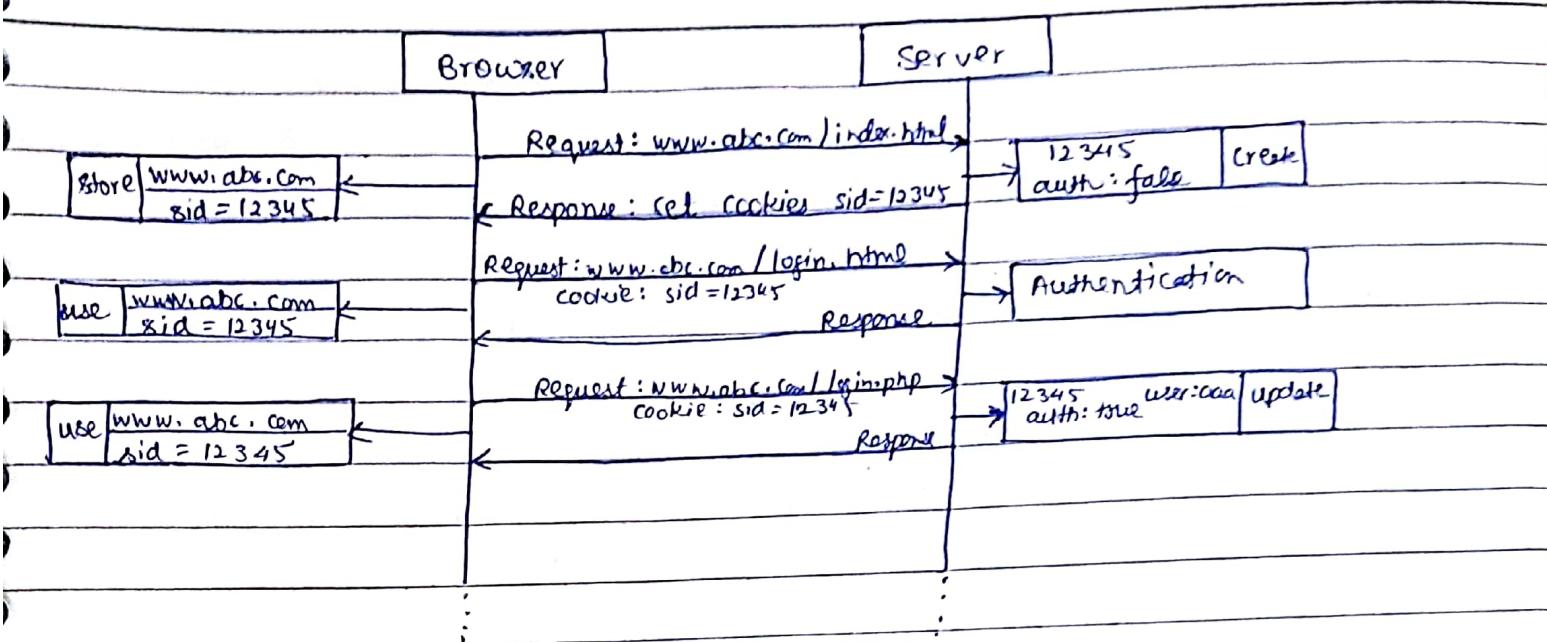
- A cookie is a small collection of information that can be transmitted to a calling browser to recognize session & returned with each call to site that created it, unless it expires.

- Sun Java System application server supports the servlets standard session interface, called ~~JSP~~ HttpSession, for all session activities.

* Authentication

- It is the mechanism of associating ~~by~~ an incoming request with a set of identifying credentials.

- REST framework provides a number of authentication schemes out of the box & allows us to implement custom schemes.



Session Management & Authentication

* HTTPS and Certificates

- HTTPS pages typically use one of two secure protocols to encrypt communication: SSL (Secure Sockets Layer) or TLS (Transport Layer Security). Both the TLS and SSL protocols use an 'asymmetric' Public Key Infrastructure (PKI) system.
- An asymmetric PKI system uses two keys to encrypt communications, a public & a private key. Anything encrypted with public key can only be decrypted by private key & vice-versa.
- When you request a HTTPS connection to a webpage, the website will initially send its SSL certificate to browser which contains the public key needed to begin secure session. Based on initial exchange, browser & website initiate 'SSL handshake' which involves generation of shared secrets to establish a uniquely secure connection.
- When an Extended Validation Certificate is installed on a website, the address bar padlock icon turns green.

* Applications Vulnerabilities & Defense

- SET is the Secure Electronic Transaction protocol for sending money over Internet. It has been jointly developed jointly by MasterCard, Visa & several computer companies.
- In SET the order information consists of two parts: the part which is private between customer & merchant of information which is private by customer & bank.

- A mechanism called dual signature, which allows either merchant or bank to read & validate signature on its half of purchase request without having to decrypt the other half, is used.

- Paypal

It is an electronic payment system which can transfer money between its accounts.

- Smart Cards

These are cards that look like credit cards but store information on a microprocessor chip instead of magnetic strips. A microchip can hold significantly more information than magnetic strip. Smart cards can run RSA encryption & can be programmed to generate pair of public/private keys.

- Payment Gateway

It is an e-commerce application service provider that authorizes payments for e-business, online retailers or traditional brick & mortar.

* Client Side Security

- Broken Authentication & Session Management
- Cross-Site Scripting (XSS)
- Sensitive Data Exposure
- Eavesdropping Attack
- ~~Man-in-Middle~~ Man-in-Middle Attack
- UI Redressing / Client jacking / Tab jacking
- Session Hijacking
- Session Fixation
- Social Engineering Attack

* Cookies Security Policy

- Cookies →
 - Session Cookies
 - Persistent Cookies

- A popular scheme is to include the following information in cookies

- 1) session ID or authorization information
- 2) time & date the cookie was issued

3) an expiration time

4) IP address of the browser the cookie was issued to

5) a message authenticity check (MAC) code (Ensure that none of the fields of the cookie have been tampered with)

* HTTP Security Extensions

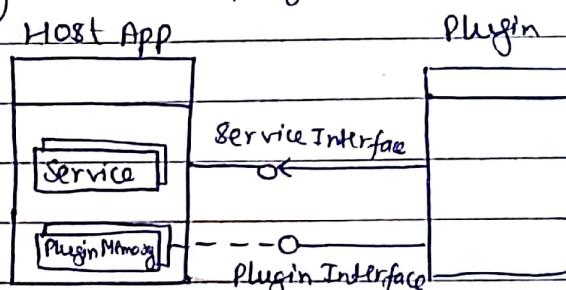
- HTTPS is the secure version of HTTP. It means all communications between your browser & websites are encrypted.

- HTTP security extension is a small software program that can modify & exchange the functionality of HTTP security

* Plugins, Extensions & Web Apps

- Plugins are scripts that enhance the functionality of analytics.js to help solve problems & aid in measuring user interaction.

- Plugin is a software component that adds a specific feature of an existing computer program



- By getting command, the name of the plugin is invoked by plugin's function. When plugin command is run, then it registers the plugin with gac command.

* Web User Tracking

- Campaign Reporting

- Real-time Reporting

- Keep stats indefinitely

- Why visitors leave?

- Human Events

* Server-Side Security Tools

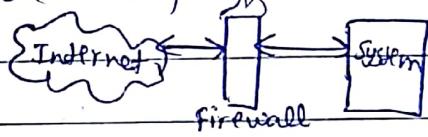
① Password Cypher

② Security Onion → Linux based tool used as IDS (Intrusion Detection System) of network security monitoring.

- ③ NST (Network Security Toolkit) → Bootable Live CD containing an open source networking & security tool
- ④ OpenVAS (vulnerability Access System)
- ⑤ OpenBuff → Stenography tool with feature include multilevel encryption
- ⑥ OpenStego → Stenography tool that takes a secret message file as input & output
- ⑦ ~~Keepass~~ Keepass → Lightweight userfriendly password manager that allow to store username & password in highly encrypted database
- ⑧ RandomOrg → Generates a random password to provide a secure path over SSL connection

* Web Application Firewalls (WAF)

- It is a security module in an application proxy device that protects the web application server in back end from various attacks like XSS & SQL injection.
- Categories of WAF:



① Network Based Firewall:

It is a computer firewall operating at application layer protocol stack & is also known as proxy based application

② Host Based Firewall:

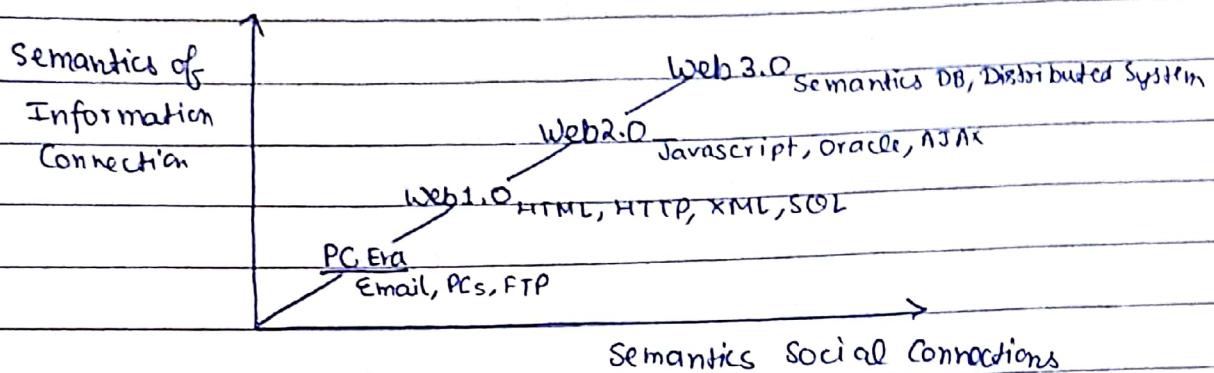
It monitors any application input, output or system service calls made from or by an application.

* Fuzzer

- It is a program which injects automatically semi-random data into a program / stack and detect bugs.
- The data generation part is made of generators, & vulnerability detection identification relies on debugging tools.
- Generators are usually use combinations of static fuzzing vectors, or totally random data. New generation fuzzers use genetic algorithms to link injected data & observed impact.

Unit-IV

* Introduction to Web 2.0 and Web 3.0



- During web1.0 phase the focus was primarily on building web, making it accessible, & commercializing it for the first time.
- Web2.0 refers to a supposed second generation of Internet-based services such as social networking sites, wikis, folksonomies etc.
- Web3.0 refers to the supposed third generation of Internet-based services that collectively comprise 'the intelligent web'-such as semantic web, microformats, natural language search, data-mining etc.

* Concepts and Terms

- Web 2.0

⇒ Main characteristics:

↳ Flexible ↳ Creative ↳ Reusable ↳ Collaborative

⇒ Key features:

① Free classification of Information

② Rich User Experience

③ User as a contributor

④ Long Tail

⑤ User Participation

⑥ Basic Trust

⑦ Dispersion

⇒ Web2.0 can be described in three parts:

(RIA)
① Rich Internet Application : defines the experience brought from desktop to browser whether it is from graphical point of view or usability point of view. Some buzzword related to RIA are AJAX & Flash.

(ii) Web-Oriented Architecture (WOA): key piece in Web 2.0 which defines how Web 2.0 ~~app~~ applications expose their functionality so that other applications can leverage and integrate the functionality providing a set of much richer applications. Examples are feeds, RSS, web services, Mash-ups.

(iii) Social Web: defines how Web 2.0 tends to interact much more with the end user & make end-user an integral part.

=> Interactive & Social Web

A social networking service is an online service, platform or site that focuses on facilitating the building of social networks or social relations among people who, for example, share interests, activities, backgrounds, or real-life connections.

=> Blogs:

A blog is a discussion or informational site published on the WWW & consisting of discrete entries typically displayed in reverse chronological order (the most recent post appears first). Different types of blogs are:

- i) Personal blogs
- ii) Microblogging
- iii) Corporate & Organisational blogs
- iv) By genre (such as political, health, travel etc)
- v) By media (blog comprising videos called vlog, links called linky etc. A rare type of blog hosted on Gopher protocol is called phlog)
- vi) By device (moblog etc)
- vii) Reverse Blog (composed by its user rather than single blogger)

=> wikis:

A wiki is a site which allows its users to add, modify or delete its content via a web browser usually using a simple mark-up language or a rich-text editor.

=> Technologies behind Web 2.0:

① AJAX

↳ AJAX is a technique for creating fast & dynamic web pages.

↳ AJAX allows web pages to be updated asynchronously by exchanging small amounts of data with server behind the scenes. This means it is possible to update parts of a web page without reloading the webpage.

② RSS

②

RSS

- ↳ RSS stands for Rich Site Summary or Real Simple Syndication
- ↳ Weblogs generate a behind-the-scenes code in XML. This code is usually referred to as a "feed" makes it possible for readers to "subscribe" to content that is created on a particular weblog so they no longer have to visit the blog itself to get it.

③ Syndication

- ↳ It refers to the websites providing information of the websites displaying it.

④ Web Syndication

- ↳ It is a form of syndication in which website material is made available to multiple other sites.

⇒ Open API

It is a word used to describe sets of technologies that enable websites to interact with each other using REST, SOAP, Javascript & other web technologies.

⇒ SLATES

- ↳ It is an initialism that describes the business impacting capabilities derived from effective use of web 2.0 technologies in & across enterprises.

↳ Search, Links, Authorship, Tags, Extensions, Signalling

S → ① A web search query is a query that a user enters into web search engine to satisfy their information needs.

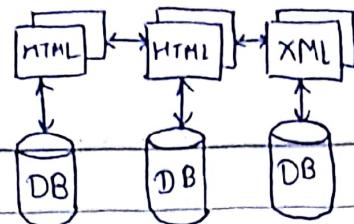
L ② The use of links or URLs to forge deep interconnections b/w information content across collaborating enterprises.

A ③ The ability of all individuals within & across enterprises to easily publish contents accessible across collaborating enterprises

T ④ The use of tags to enable the rapid & humanistic organisation of data across collaborating enterprises

E ⑤ The mining of previously gathered data relating to a user activities or transactions which ~~can~~ allow users to be guided to initiate other valuable activities or transactions

S ⑥ The sending of alerts to users of changing state of an element of interest, i.e., using RSS.



- Web 3.0

⇒ Key features:

- ① Ubiquitous Connectivity
- ② Network Computing
- ③ Open Technologies
- ④ Open Identity
- ⑤ The Intelligent Web

⇒ Semantic Web

↳ A collaborative movement led by W3C. It provides a common framework that allows data to be shared or reused across application, enterprise & community boundaries.

↳ Semantic Web stack is built on W3C's Resource Description Framework (RDF).

↳ Semantic Web sol'n involves publishing in languages specifically designed for data: Resource Description Framework (RDF), Web Ontology Language (OWL), & XML.

⇒ Widgets

↳ A widget is a standalone application that can be embedded into third party sites by any user on a page where they have rights of authorship.

↳ Major two types of widgets are Web Widgets & Desktop Widgets.

↳ One major consideration in use of widgets is how they will impact the SEO (Search Engine Optimization) of your site.

⇒ Drag & Drop Mashups

↳ iGoogle is a customizable homepage: iGoogle lets you create a personalised homepage that contains a Google search box at top, & your choice of any no. of gadgets below.

⇒ Technologies behind Web 3.0 - RDF

↳ RDF data model is based upon idea of making statements about resources in the form of subject-predicate-object expressions.

↳ These expressions are known as triples in RDF terminology.

↳ The subject denotes the resources, & the predicate denotes traits or aspects of the resource & expresses a relationship between the subject & the object.

⇒ Web Based Information System

↳ A Web information system, or web-based information system, is an

information system that uses Internet Web Technologies to deliver information & services, to users or other information systems / applications. It is a software system whose main purpose is to publish & maintain data by using hypertext principles.

↳ ~~Database~~ Approaches to design recommender systems:

① Collaborative Filtering

These methods are based on collecting & analyzing a large amount of information on users' behaviours, activities or preferences & predicting what users will like based on the similarity to other users.

② Content-based Filtering

These methods are based on information about & characteristics of the items that are going to be recommended.

③ Hybrid Recommender Systems

Combining collaborative filtering & content-based filtering

④ Mobile Recommender Systems

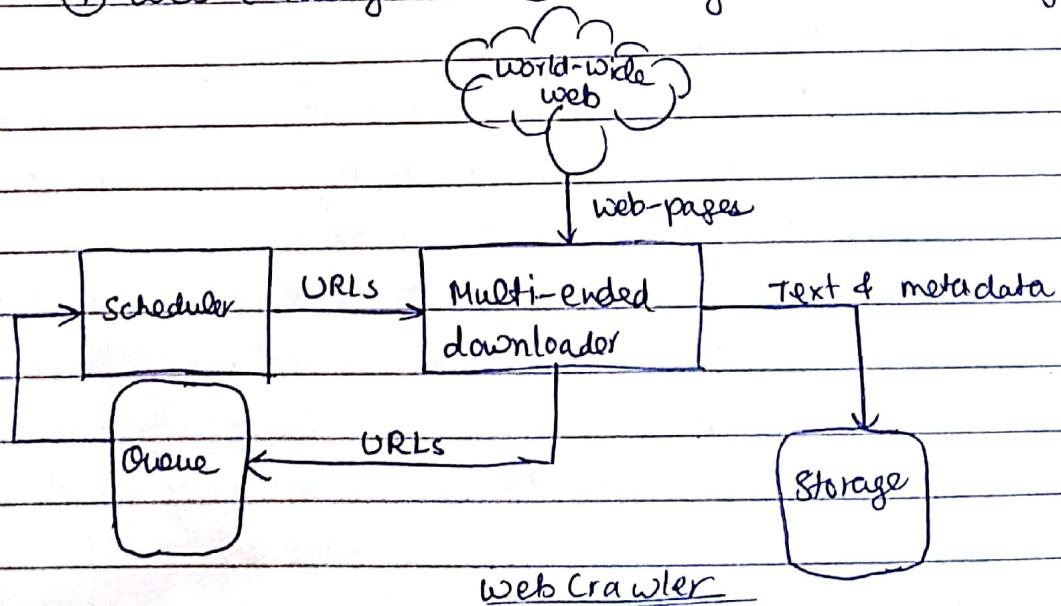
Mobile & Smart phones based internet-accessing

⑤ ⇒ Web Search Engine

↳ It is designed to search for information on World Wide Web. The search results are generally presented in a line of results often referred to as search engine result pages (SERPs).

↳ A search engine operates in the following order:

- ① Web Crawling
- ② Indexing
- ③ Searching



⇒ Recommender Systems : Subclass of information filtering system that seek to predict the 'rating' or 'preference' that a user would give to an item or social element they had not yet considered, using a model built from the characteristics of an item or the user's social environment.

⇒ Web Mining

↳ web mining is the application of data mining techniques to discover patterns from the Web.

↳ Web mining can be divided into three different types:

(1) Web Usage Mining

It is the process of extracting useful information from server logs, i.e., users history.

(2) Web Structure Mining

It is the process of using graph theory to analyze the node connection structure of a web site. Web structure mining can be divided into two kinds:

a) Extracting patterns from hyperlinks in a web

b) Mining the document structure.

(3) Web Content Mining

It is the mining, extraction & integration of useful data, information & knowledge from webpage contents.

* Latest Trends In Web Technologies

- The client-side technologies used in web 2.0 development include AJAX & Javascript framework such as YUI Library, Dojo Toolkit, MooTools, jQuery, Ext JS & Prototype Javascript Framework.

- AJAX programming uses Javascript & the DOM to update selected regions of the ~~web~~ page area without undergoing a full page reload.

1) To allow users to continue interact with page, communications such as data requests going to the server are separated from data coming back to the page (asynchronously).

2) The data fetched by an AJAX request is typically formatted in XML or JSON format, two widely used structured data formats.

- 3.) As a widely available plugin independent of W3C standards, Adobe Flash is capable of doing many things that are ^{not} possible pre-HMIS.
- 4.) In addition to Flash & AJAX, Javascript / Ajax frameworks have recently become a popular means of creating web 2.0 sites.
- 5.) On the server-side, web 2.0 uses many of the same technologies as Web 1.0 technologies such as PHP, Ruby, Perl, Python as well J2EE & Microsoft .NET framework.

* Web Security Concerns

- Web sites are unfortunately prone to security risks. And so are any networks to which web servers are connected. Setting aside risks created by employee use or misuse of network resources, your web server & the site it hosts present your most serious sources of security risk.
- Web servers by design open a window between your network & the world. The care taken with server maintenance, web application updates & your web site coding will define the size of that window, limit the kind of information that can pass through it & thus establish the degree of web security you will have.

* Applications of Web Engineering Technologies in Distributed Systems

- The www can be viewed as a huge distributed system with millions of clients & servers for accessing linked documents.
- Servers maintain collections of documents while clients provide user an easy-to-use interface for presenting & accessing those documents.
- A document is fetched from a server, transferred to ~~the~~ a client, & presented on the screen. To a user conceptually no difference b/w a document stored locally or in another part of the world.
- Now, web has become more than just a simple document based system \Rightarrow with emergence of web services, it is becoming a system of distributed services rather than just documents offered to any user or machine.
- web documents can be built in two ways :
 - ① Static - locates & returns the object identified in the request.

Static objects, include predefined HTML pages & JPEG or GIF files, doesn't require web servers to communicate with any server-side application

② Dynamic - the request is forwarded to an application system where the reply is generated dynamically, i.e., data is generated through a server-side program execution.

- Although web started as a simple two-tiered client-server architecture for static web documents, this architecture has been extended to support advanced type of documents.

- Because of the server side processing, many websites are now organized as three-tiered architectures consisting of a web server, an application server, & a database server.

- User data comes from an HTML form, specifying the program & parameters.

- Server-side scripting technologies are used to generate dynamic content:

⇒ Microsoft : Active Server Pages (ASP.NET)

⇒ Sun : Java Server Pages (JSP)

⇒ Netscape : Javascript

⇒ Free Software Foundation : PHP

* Case Studies Using Different Tools

- Distributed computing is a branch of computer science that studies the distributed systems. It refers to the use of distributed systems to solve computational problems. Here the problem is divided into multiple tasks, & each task is solved by one computer.

- Distributed System consists of multiple autonomous computers that communicate through a computer network. These computers interact with each other in order to achieve a common goal. The computer application or program which runs on a distributed system is called distributed program.

The process of writing the distributed program is called distributed programming.

- Distributed systems are the Internet based applications that where there are two types of computer systems: Server node & Client node.