

UNIT-III XML AND WEB SERVERS

3.1 Create XML file

XML Introduction

- XML is Extensible Mark-up Language and it is text based mark-up language, derived from Standard generalized Markup Language (SGML).
- It is used to store and process the data on web. It is easy to implement.
- XML allows us to develop interoperable (platform independent, language independent), distributed applications.
- XML provides customized (user defined) tags to format and display textual information.
- XML can be used to exchange the information between organizations and different systems.

Features of XML

- **Standard language:** It specifies structural rules of tags, and was developed by an organization called the World Wide Consortium(W3C) it is an open standard.
- **Platform independent:** It can support any operating system.
- **Extensible language:** It specifies that user can create their own tags.
- **XML files are text files** ,they can be managed by all text editors.
- **XML** is extremely suitable for document storage and processing both online and offline

XML Syntax Rules:XML document must contain one root element hat is the parent of all other elements.

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

```
<student>
  <id>1234</id>
  <name>
    <firstname>abc</firstname>
    <lastname>xyz</lastname>
  </name>
  <branch> DCME</branch>
</student>
```

3.1.1 Describe the organization of data in the form of XML file

- ✓ XML defines certain rules for its syntax to specify how to create or structure an XML document. Syntax that is used to create an XML document is called markup syntax.
- ✓ XML section contains three sections **1)XML Declaration**
2)Document Type Definition
3)Document Elements

1)XML Declaration: It contains details that an XML processor to parse the xml document.It must appear in the first line of XML Document.It is also called as **Document Prolog**.

- <? Xml version ="version number"?>
- In processing instructions we have three parameters: **Version number,encoding, stand alone**

2) Document Type Definition(DTD):It is used to describe the XML language completely.

- DTD provides a framework to validate XML documents.
- DTD defines the structure of an XML document with a list of legal elements.

➤ **Syntax:**

```
<! Doctype element DTD identifier  
[  
  Declaration 1  
  Declaration 2  
]>
```

- **DTD** starts with <!doctype delimiter.
- **Element** tells the parser to parse the document from the specified root element
- **DTD Identifier** it may be the path to a file on the system or url to a file on the internet.
- **[] brackets** enclose an optional list.
- DTDs are of two types 1. Internal/inline
2. External

1.Internal DTD: **Syntax:-**<!DOCTYPE root element name[local subset definitions]>

Example

```
<?xml version="1.0"?>  
<!DOCTYPE email [  
<!ELEMENT email (to,from,heading,body)>  
<!ELEMENT to (#PCDATA)>  
<!ELEMENT from (#PCDATA)>  
<!ELEMENT heading (#PCDATA)>  
<!ELEMENT body (#PCDATA)>  
>  
<email>  
<to>abc</to>  
<from>xyz</form>  
<heading>Welcome</heading>  
<body>Welcome to our school</body>  
</email>
```

2.External DTD: **Syntax:**<!DOCTYPE root element name SYSTEM "external dtd file">

```
<!ELEMENT email (to,from,heading,body)>  
<!ELEMENT to (#PCDATA)>  
<!ELEMENT from (#PCDATA)>  
<!ELEMENT heading (#PCDATA)>  
<!ELEMENT body (#PCDATA)>
```

Save this file as **email.dtd**

XML file:-

```
<?xml version="1.0"?>
<!DOCTYPE email SYSTEM "email.dtd">
<email>
<to>abc</to>
<from>xyz</form>
<heading>Welcome</heading>
<body>Welcome to our school</body>
</email>
```

- We use **! exclamation** mark at the start of element name
- **#PCDATA** (parsable character data) defines the element name of the type.

3) Document Elements

- XML document must have a starting tag and closing tag. Tags are case-sensitive.
- Elements must be nested properly it must have only one root element.
- XML attribute values must be enclosed in double quotes.
- Comment begins with **<!--** and ends with **--!>**
Eg: <!--This is a comment --!>
- XML elements are divided into two categories.
 - (a) **Empty element:** An empty element does not contain any content or any other element within it.

(b) Nested elements: Elements which contain other elements are known as nested elements

```
Eg: <?xml version="1.0"?>
<students>
<firstname>Ambrish</firstname>
<lastname title="singh"></lastname>
<brach>DCME<branch>
</students>
```

- An XML tree represents the elements in a tree structure.
 1. **Root element**
 2. **Parent element**
 3. **Child element**
 4. **Siblings**

Differences between XML and HTML

<u>XML</u>	<u>HTML</u>
1.Describes data and focus on what data is.	1.Displays data and focus on how data looks.
2.Supports case-sensitivity.	2.Doesnot support case-sensitivity.
3.Users can define their own tags.	3.Predefined tags only
4.It is mandatory to close each and every tag.	4.It is not compulsory to close the tags.
5.Attribute values must be in double quotes.	5.Not necessary to enclose attribute values in double quotes.

3.1.2. State The Significance Of Namespace

- Namespaces are used to avoid the element name conflicts while combining the data of different xml documents or HTML documents.

Consider a HTML file

```
<table>
<tr>
<td>name1</td>
<td>name1</td>
</tr>
</table>
```

Sample xml program

```
<table>
<bookname>web design</bookname>
<author>Xavier</author>
<version>4.0</version>
</table>
```

Now,if we combine these two files a name conflict may occur ,this can be resolved by using name prefix.

Syntax : <prefix="element" xmlns:element="url"> (or)
<prefix:tagname>tag content</prefix:tag name>

By using prefix we get:

```
<? Xml version ="1.0"?>
<n:table xmlns:n="http://www.example1.com">
<n:tr>
<n:td>name1</n:td>
<n:td>name1</n:td>
</n:tr>
</n:table> // HTML is defined with prefix 'n'
<b:table xmlns:b="http://www.example2.com">
<b:bookname>web design</b:bookname>
<b:author>Xavier</b:author>
<b:version>4.0</b:version>
</b:table> //XML is defined with prefix 'b'
```

3.1.3 List the applications of XML

- Some of the applications of XML are, XHTML,WML,SOAP,CML,
 1. **XHTML**:It is an enhanced version of HTML4 and belongs to a family of XML. The main purpose of developing XHTML is to make HTML more extensible and also increase the interoperability with different data formats.It achieving portability across different platforms and browsers.

2. **WML:** Wireless Markup Language (WML) is a markup language intended for use on mobile applications.
3. **SOAP:** It is defined as a Simple XML –based protocol which permits applications to exchange XML based messages over computer networks using HTTP.
4. **CML:** Chemical Markup Language, it is developed as one of the SGML application, which is used for representing molecular and chemical information without data corruption.

3.2.Explain about Web Servers

Introduction to webserver

- A webserver is a hardware/software that response to the client request by providing resources, when users enter URL into web browsers they request specific documents from web server, it maps URL to file on web server and returns requested documents.
- It uses HTTP protocol that works on any platform, communicates with client and server.

Functions of Web server

- Manage the web
- Perform multitasking
- Tracking of files and directories

Different types of web servers

1. **Microsoft Internet Information Web server(IIS):** IIS is an enterprise level webserver created by Microsoft it works only on windows platform. It provides web capabilities that ranges from publishing information to the data stored in different client/server databases. It supports HTTP,HTTPS and FTP.
2. **Personal Webserver(PWS):** PWS is a scale down version of Internet information server. Both IIS and PWS are introduced by Microsoft company.It is preferred in educational institutions and individual users and small businesses as it do not need the system to be used exclusively.
3. **Apache Web Server:** It is mostly used webserver maintained by Apache software foundation. Its popularity is due to features like stability, efficiency and portability. It is freely available software and can be customized.

3.2.1 Distinguish Client side scripting and Server side scripting

Scripting:-A Script refers to a small non-compiled program that is interpreted by some other program and executed by end user..Scripts are written using different types of languages called scripting languages. These are of two types.

1. Client side scripting: Client side scripting language is used to run scripts usually in web browser. The source code is transferred from the web server to the user's computer over the internet.

Eg:-javascript, jquery, VB script

2. Server side scripting: It runs a scripting language on web server. A users request is fulfilled by running a script directly on the web server to generate dynamic HTML pages. This HTML is then sent to the client browsers.

Eg:-PHP,ASP,Java,Python,perl CGI

<u>Client side scripting</u>	<u>Server side scripting</u>
1.In client side scripting program/script is executed on web browser by client. Eg:Javascript,VB script.	1.In server side scripting program/script is executed on webserver by server. Eg:JSP,PHP.
2.It depends on the browsers	2.It does not depend on the browsers
3.It provides less flexibility to programmers.	3.It provides more flexibility to programmers.
4.It does not provide security as code is visible to client.	4.It provides security as code is not visible to client.
5.Posses less number of built in objects.	5.Posses more number of built in objects.
6.It cannot access data available in database.	6.It can access data available in database.
7.It cannot extend its functionality.	7.It can extends its functionality with the help of softwares available in server side.
8.It cannot access its file directory structure.	8.It can access its file directory structure.

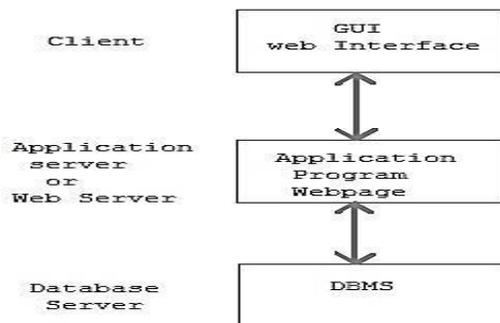
3.2.2 Architecture of webserver

The architecture of webserver consists of three-tier, they are

1. Information tier(data tier (or)bottom tier)
2. Middle tier.
3. Client tier (or) top tier

1. Information tier: This is present at the bottom of the architecture and it consists of database. So, it is called as bottom tier (or) data tier. However, it is responsible for maintaining application data.

2.Middle tier: This tier incorporate between client tier and data tier. It is responsible for implementing the business logic, presentation logic and control logic in order to maintain interaction between application clients and application data. It process the client request which come from top-tier and acquires the desired data from the database (i.e information tier).After processing the request, it responds to the client by presenting the content in the form of XHTML,HTML etc.



3.Client tier: This is present at top of the architecture. So, it is called top tier. Client can interact directly. It displays data to user.

3.2.3 Identify various HTTP request types and their differences

- The Hypertext Transfer Protocol (HTTP) is designed to enable communications between clients and servers.

- HTTP works as a request-response protocol between a client and server.
- A web browser may be the client, and an application on a computer that hosts a web site may be the server.

The various **HTTP request types** are

a)GET: The get request type sends the data captured by the form elements to the webserver whose address is written in the URL.URL refers to the address of the web server. It is used when there is a small amount of data to be sent to the web server. Minimum amount of data that can be sent to the web server is 1024 characters.

Eg: /test/demo_form.php?name1=value1&name2=value2

b)POST: The post request type also sends the data captured by form elements to the web server as a separate bit stream of data. When there is large amount of data to transfer we can use post request.

Eg: POST /test/demo_form.php HTTP/1.1

Host: w3schools.com

name1=value1&name2=value2

c)PUT: Uploads a representation of the specified URI.

d)HEAD: Same as GET but returns only HTTP headers and no document body.

e)CONNECT: Converts the request connection to a transparent TCP/IP tunnel.

f)DELETE: Deletes the specified resource.

Differences between get and post request:

<u>Get request</u>	<u>Post request</u>
1.Data is appended to the URL	1.Data is not appended to the URL
2.It can send limited amount of data	2.It can send large amount of data
3.It does not offer security since it is visible to every person.	3.It offer security since it is displayed in the URL.
4.It can be bookmarked	4.It cannot be bookmarked
5.It can be cached.	5.It cannot be cached.
6.It is more efficient.	6.It is less efficient.

3.2.4 Installation steps for IIS,PWS and Apache web server

Installation of IIS

1. To open the **Windows Features** dialog box, click **Start**, and then click **Control Panel**
2. In the **Control Panel**, click **Programs**.



3. Click **Turn Windows features on or off**.

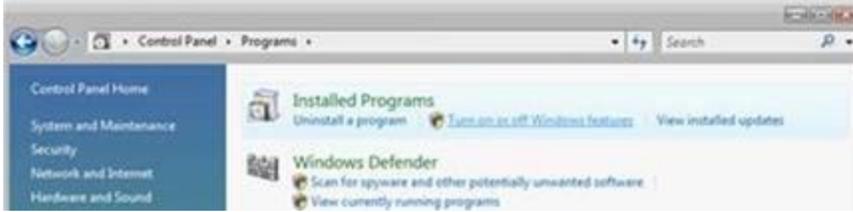


Figure 3: Control Panel Install options

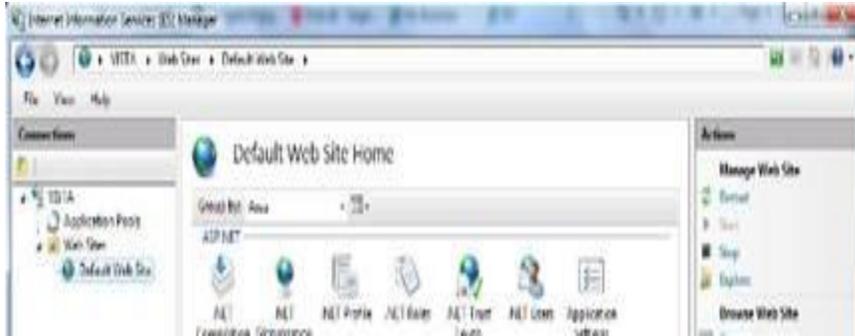
4. Expand **Internet Information Services**. Additional categories of IIS features are displayed. Select **Internet Information Services** to choose the default features for installation. Expand the additional categories displayed, and select any additional features you want to install, such as **Web Management Tools**.



5. If you are installing IIS for evaluation purposes, you may want to select additional features to install. Select the check boxes for all IIS features you want to install, and then click **OK** to start installation.
6. IIS is now installed with a default configuration on Windows Vista or Windows 7. To confirm that the installation succeeded, type the following URL into your browser, <http://localhost>.



7. Next, you can use Internet Information Services Manager to manage and configure IIS. To open IIS Manager, click **Start**, type `inetmgr` in the **Search Programs and Files** box, and then press ENTER.

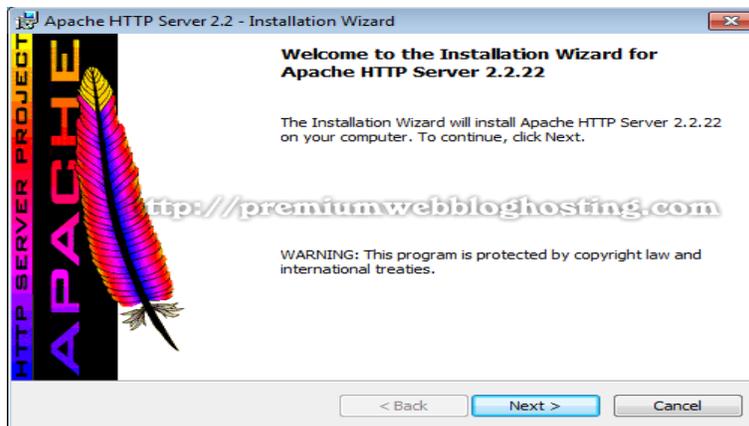


INSTALLATION STEPS FOR PWS

- Insert your Windows 98 CD-ROM in your CD-ROM drive.
- Click Start, and then click Run.
- In the Open box, type the path to the Setup.exe file, where x is the letter of your CD-ROM drive.
- Click OK. Follow the instructions in Personal Web Server Setup.

INSTALLATION OF APACHE WEB SERVER

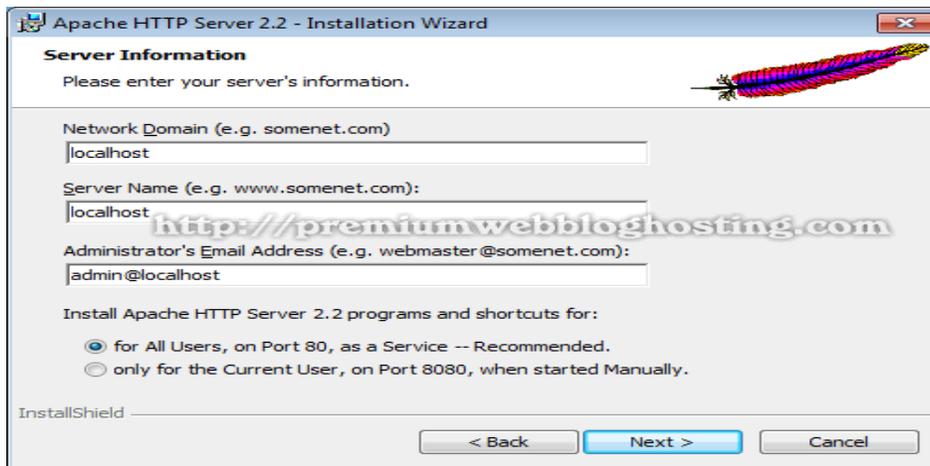
- Go to archive.apache.org/dist/httpd/binaries/win32 webpage
- Once you've finished downloading the Apache MSI Installer, double click the file to start the installation.



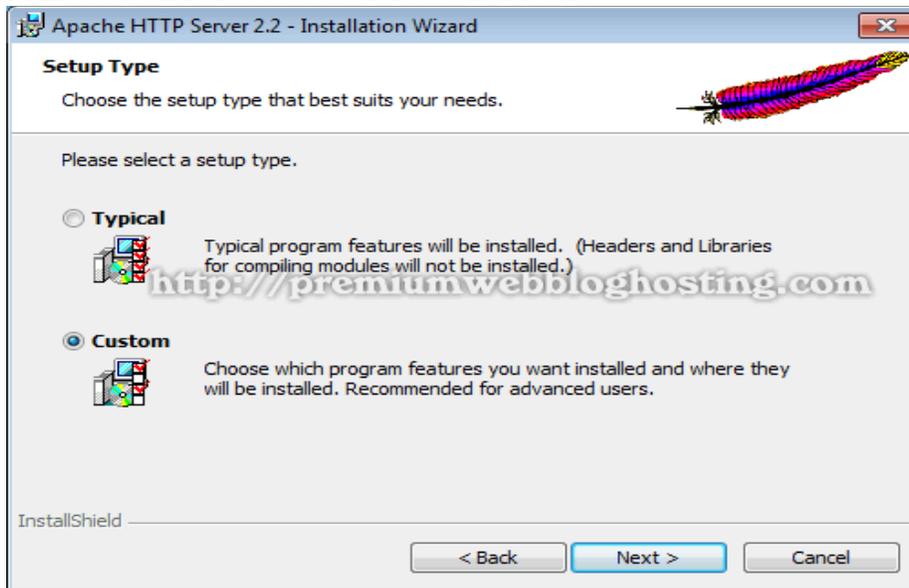
- Click Next.



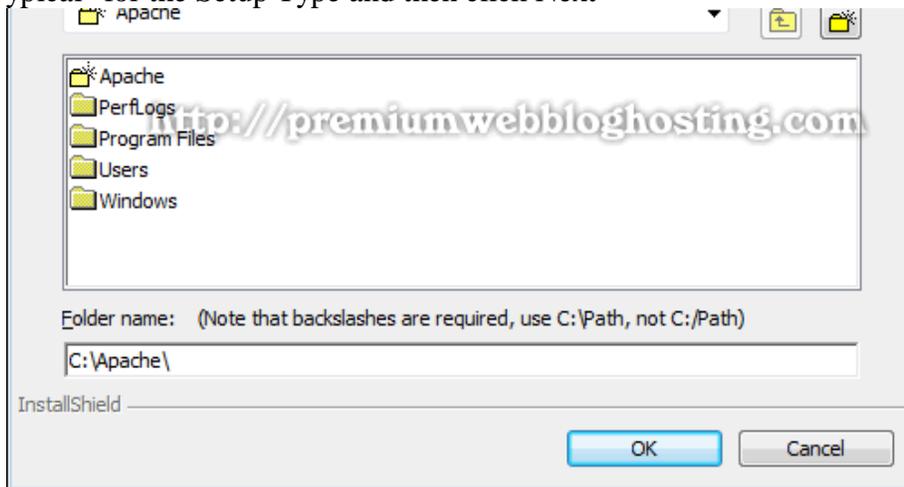
- Read the License Agreement - If you have no problem with it check "I accept the terms in the license agreement," then click Next.



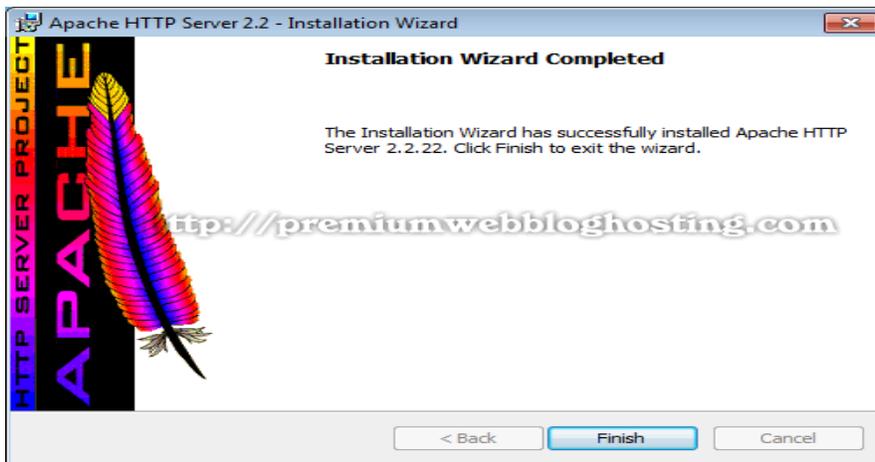
- Check for All User on Port 80 as a service. Click Next.



- Select "Typical" for the Setup Type and then click Next

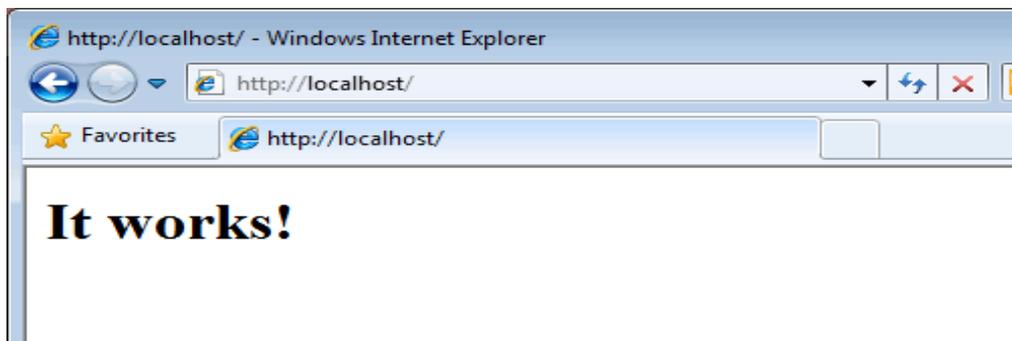
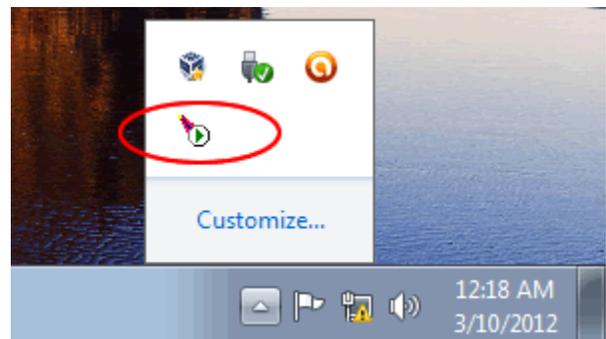


- Select the destination folder and click on next.



- Wait a few moments until Apache Server fully installed. Click "Finish" to exit the installation.
- An icon (Apache Monitor) will appear in the System tray icons as shown below. Left click the icon to stop, to start and to restart the Apache Server.
- Don't forget to always restart Apache Server each time we edit or make any changes in the Apache Configuration file (http d.conf).

Testing Server: Now it's time to test our server, open up browser in the address bar type in "http://localhost/" or "localhost" and hit Enter.



3.2.5. Compare IIS,PWS and Apache web server

IIS	PWS	Apache
-----	-----	--------

1.IIS is abbreviated as Microsoft Internet Information Services which is introduced by Microsoft corporation.	1.PWS is abbreviated as Microsoft Personal Web Server which is introduced by Microsoft corporation.	1.Apache is open source apache web server which is introduced by Apache software.
2.It is released on 17-2-2000	2.It is released on 4-12-1997	2.It is released on 21-5-2001
3.It can run on windows platform.	3.It can run on windows95/98/ME/NT platforms.	3.It can run on UNIX,LINUX and Windows platforms.
4.It is most popular web server available with windows 2000	4.It is basic web server used to publish personal web pages.	4.It is most popular web server which is currently in use on the internet.
5.It is included with windows.	5.Included with windows OS.	5.It is freely available on internet.

3.2.6. Describe the steps to place and request HTML,PHP documents from webservers:

Requesting HTML documents

(a) The following are the steps to request an HTML /XHTML document from the web servers such as IIS and PWS:

1. Initially copy the file named test .html into the directory which refers as virtual directory
2. Launch the internet explorer web browser.
3. Put the location of HTML/ XHTML document in the "address field".
4. Press 'enter' key and view the result of requested document (test.html) from the web server.

(b) The following are the steps to request an HTML /XHTML document from the apache web server:-

1. Initially, the HTML /XHTML document is required to be saved in the default directory i.e., htdocs directory
2. In windows platform, the director is c:\programfiles\apachegroup\apache in which the test .html file should be present
3. Launch the internet explorer or equivalent UNIX or LINUX web browser.
4. Put the location of HTML /XHTML document in the "address field" .
5. Press 'enter key and view the result of requested document (test .php) from the apache web server.

Requesting PHP documents:

The following are steps to request a PHP document from the web server such as IIS,PWS and apache:-

- 1.Initially ,launch the internet explorer web browser.
- 2.Put the address location of the PHP document in the 'address field'.
- 3.Press'enter' key and view the result of requested document (test.php) from the web server.