SYLLABUS (2016-17)

WEB ENGINEERING (ETCS-308)

Instructions to Paper Setters:

1. *Question No.1 should be compulsory and cover the entire syllabus. Thus question should have objective or short answer type questions. It should be of 25 marks.*
2. *Apart from Question No.1, rest of the paper shall consists of four units as per the syllabus. Every unit should have two question. However, student may be asked to attempt only 1 question from each unit,. Each question should be of 12.5 marks.*

UNIT–I

History of the Internet, Basic internet protocols, World Wide Web (W3C), HTTP: Hypertext Transfer Protocol.

Markup languages-XHTML: Introduction to HTML, basics of XTHML, HTML elements, HTML tags, lists, tables, frames, forms, defining XHTML's abstract syntax, defining HTML documents.

CSS style sheets: Introduction, CSS core syntax, text properties, CSS box model, normal flow box layout, other properties like list, tables, DHTML, XML, XML documents & vocabulary, XML versions and declarations, Introduction to WML.

[T1,T2][No. of hrs. 10]

UNIT–II

Client Side Programming: JAVA Scripts, basic syntax, variables & data-types, literals, functions, objects, arrays, built-in objects, JAVA Script form programming, Intrinsic event handling, modifying element style, document trees,

Server side programming: Java Servlets: Servlet architecture, life cycle, parameter data, sessions, cookies, servlets capabilities, servlets and concurrency. Introduction to JSP, JSP Tags, JSP life cycle, custom tags.

[T1,T2][No. of hrs. 12]

UNIT–III

Security Threats, Security risks of a site, Web attacks and their prevention, Web security model, Session management, authentication, HTTPS and certificates, Application vulnerabilities and defenses.

Client-side security, Cookies security policy, HTTP security extensions, Plugins, extensions, and web apps, Web user tracking.

Server-side security tools, Web Application Firewalls (WAFs) and Fuzzers.

[T1,T2][No. of hrs. 10]

UNIT–IV

Introduction to Web 2.0 and Web 3.0, Concepts and Issues, Latest Trends in Web Technologies. Web Security concerns. Applications of Web Engineering Technologies in distributed systems etc. Case studies using different tools.

[T1,T2][No. of hrs. 12]

MODEL PAPER-I
SIXTH SEMESTER (B. TECH.)
FIRST TERM EXAMINATION
WEB ENGINEERING (ETCS-308)

Time : 1 Hour **MM: 30**

Q1. (*a*) Explain HTTP in detail.

Hypertext Transfer Protocol (HTTP) is an application protocol for distributed, collaborative, hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web.

Hypertext is structured text that uses logical links (hyperlinks) between nodes containing text. HTTP is the protocol to exchange or transfer hypertext.

HTTP functions as a request-response protocol in the client-server computing model. A web browser, for example, may be the client and an application running on a computer hosting a web site may be the server. The client submits an HTTP request message to the server. The server, which provides resources such as HTML files and other content, or performs other functions on behalf of the client, returns a response message to the client. The response contains completion status information about the request and may also contain requested content in its message body.

A web browser is an example of a user agent (UA). Other types of user agent include the indexing software used by search providers (web crawlers), voice browsers, mobile apps, and other software that accesses, consumes, or displays web content.

HTTP is designed to permit intermediate network elements to improve or enable communications between clients and servers. High-traffic websites often benefit from web cache servers that deliver content on behalf of upstream servers to improve response time. Web browsers cache previously accessed web resources and reuse them when possible to reduce network traffic. HTTP proxy servers at private network boundaries can facilitate communication for clients without a globally routable address, by relaying messages with external servers.

Q.1. (*b*) What is W3C explain.

Ans. The World Wide Web Consortium (W3C) is an international community where Member organizations, a full-time staff, and the public work together to develop Web standards.

W3C does not have a typical organizational structure, nor is it incorporated. There are at least two ways to think about how W3C is organized:

1. in administrative terms
2. in process terms

In administrative terms: W3C is administered via a joint agreement among these "Host Institutions": MIT , ERCIM , Keio University, and Beihang University. The W3C staff (many of whom work physically at one of these institutions) is led by a Director and CEO. A small management team is responsible for resource allocation and strategic planning on behalf of the staff. Regional offices play an important role in W3C being an international organization.

In process terms: the W3C Process Document, Member Agreement, Patent Policy, and a few others documents establish the roles and responsibilities of the parties

involved in the making of W3C standards. Some key components of the organization are.

- the Advisory Committee, composed of one representative from each W3C Member. The Advisory Committee has a number of review roles in the W3C Process, and they elect the Advisory Board and TAG.
- the Advisory Board, an advisory body elected by the Advisory Committee
- the Technical Architecture Group (TAG), which primarily seeks to document Web Architecture principles
- the W3C Director and CEO, who assess consensus for W3C-wide decisions
- the chartered groups, populated by Member representatives and invited experts, and which produce most of W3C's deliverables according to the steps of the W3C Process.

Q.2. (*a*) Explain WML.

Ans. Wireless Markup Language (WML), based on XML, is a markup language intended for devices that implement the Wireless Application Protocol (WAP) specification, such as mobile phones. It provides navigational support, data input, hyperlinks, text and image presentation, and forms, much like HTML (HyperText Markup Language). It preceded the use of other markup languages now used with WAP, such as HTML itself, and XHTML.

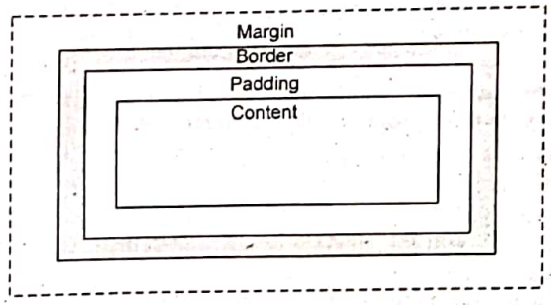
WML is a free and extensible Webdesigner's off-line HTML generation toolkit for Unix, distributed under the GNU General Public License (GPL v2). It is written in ANSI C and Perl 5, built via a GNU Autoconf based source tree and runs out-of-the-box on all major Unix derivates. It can be used free of charge both in educational and commercial environments.

WML consists of a control frontend driving up to nine backends in a sequential pass-oriented filtering scheme. Each backend provides one particular core language. For maximum power WML additionally ships with a well-suited set of include files which provide higher-level features built on top of the backends core languages.

Q.2. (*b*) Explain CSS box model.

Ans. The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content.

The image below illustrates the box model:



Explanation of the different parts:

- **Content**: The content of the box, where text and images appear
- **Padding**: Clears an area around the content. The padding is transparent
- **Border**: A border that goes around the padding and content

- **Margin**: Clears an area outside the border. The margin is transparent

The box model allows us to add a border around elements, and to define space between elements.

Q.3. (a) Difference between HTML and XML.

Ans. • XML is the acronym from Extensible Markup Language (meta-language of noting/marking). XML is a resembling language with HTML. It was developed for describing data.

- The XML tags are not pre-defined in XML. You will have to create tags according to your needs.
- XML is self descriptive.
- XML uses DDT principle (Defining the Document Type) to formally describe the data.
- The main difference between XML and HTML: XML is not a substitute for HTML.
- XML was developed to describe data and to focalize on what the data represent.
- HTML was developed to display data about to focalize on the way that data looks.
- HTML is about displaying data, XML is about describing information.
- XML is extensible.

Q.3. (*b*) Define HTML and explain 5 HTML tags.

Ans.

<html>	Defines an HTML document
<title>	Defines a title for the document
<body>	Defines the document's body
<h1> to <h6>	Defines HTML headings
<p>	Defines a paragraph

Q.4. (*a*) Write short note on history of internet.

Ans. The US Department of Defense awarded contracts as early as the 1960s for packet network systems, including the development of the Apranet. Packet switching networks such as Apranet, NPL network, Cyclades. Merit Network, Tymnet, and Telenet, were developed in the late 1960s. Access to the Arpanet was expanded in 1981 when the National Science Foundation (NSF) funded the Computer Science Network (CSNET). In 1982, the Internet protocol suite (TCP/IP) was introduced as the standard networking protocol on the Arpanet. In the early 1980s the NSF funded the establishment for national supercomputing centers at several universities, and provided interconnectivity in 1986 with the Nsfnet project, which also created network access to the supercomputer sites in the United States from research and education organizations. Commercial Internet service providers (ISPs) began to emerge in the very late 1980s. The ARPANET was decommissioned in 1990. Limited private connections to parts of the Internet by officially commercial entities emerged in several American cities by late 1989 and 1990, and the Nsfnet was decommissioned in 1995, removing the last restrictions on the use of the Internet to carry commercial traffic.

Q.4. (*b*) Difference between internet and intranet.

Ans. Internet

1. Internet is wide network of computers and is open for all.
2. Internet itself contains a large number of intranets.

3. The number of users who use internet is Unlimited.
4. The Visitors traffic is unlimited.
5. Internet contains different source of information and is available for all.

Intranet

1. Intranet is also a network of computers designed for a specific group of users.
2. Intranet can be accessed from Internet but with restrictions.
3. The number of users is limited.
4. The traffic allowed is also limited.
5. Intranet contains only specific group information.

Therefore the Internet is an open, public space, while an intranet is designed to be a private space. An intranet may be accessible from the Internet, but it is protected by a password and accessible only to authorized users.

MODEL PAPER-I
SIXTH SEMESTER (B. TECH.)
SECOND TERM EXAMINATION
WEB ENGINEERING (ETCS-308)

Time. 1 Hrs. **MM: 30**

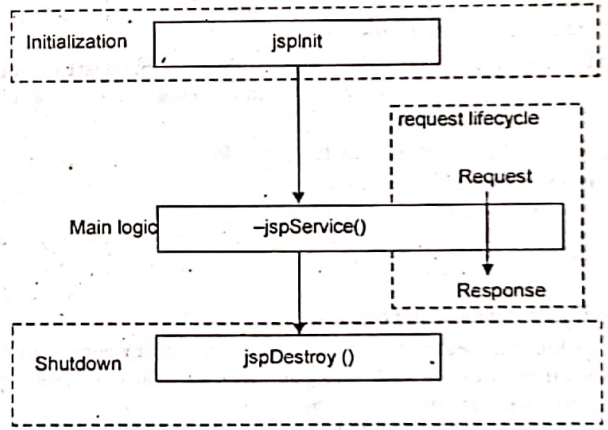
Q.1. (a) Explain JSP life cycle. **(5)**

Ans. A JSP life cycle can be defined as the entire process from its creation till the destruction which is similar to a servlet life cycle with an additional step which is required to compile a JSP into servlet.

The following are the paths followed by a JSP

- Compilation
- Initialization
- Execution
- Cleanup

The four major phases of JSP life cycle are very similar to Servlet Life Cycle and they are as follows:



JSP Compilation: When a browser asks for a JSP, the JSP engine first checks to see whether it needs to compile the page. If the page has never been compiled, or if the JSP has been modified since it was last compiled, the JSP engine compiles the page.

The compilation process involves three steps:

- Parsing the JSP.
- Turning the JSP into a servlet.
- Compiling the servlet.

JSP Initialization: When a container loads a JSP it invokes the jspInit() method before servicing any requests. If you need to perform JSP-specific initialization, override the jspInit() method:

```
public void jspInit(){
```

```
// Initialization code...
}
```

Typically initialization is performed only once and as with the servlet init method, you generally initialize database connections, open files, and create lookup tables in the jspInit method.

JSP Execution: This phase of the JSP life cycle represents all interactions with requests until the JSP is destroyed.

Whenever a browser requests a JSP and the page has been loaded and initialized, the JSP engine invokes the _jspService() method in the JSP.

The_jspService() method takes an **HttpServletRequest** and **anHttpServlet Response** as its parameters as follows:

```
void _jspService(HttpServletRequest request,
 HttpServletResponse response)
{
// Service handling code...
}
```

The _jspService() method of a JSP is invoked once per a request and is responsible for generating the response for that request and this method is also responsible for generating responses to all seven of the HTTP methods *i.e.* get, post, delete etc.

JSP Cleanup: The destruction phase of the JSP life cycle represents when a JSP is being removed from use by a container.

The **jspDestroy**() method is the JSP equivalent of the destroy method for servlets. Override jspDestroy when you need to perform any cleanup, such as releasing database connections or closing open files.

The jspDestroy() method has the following form:

```
public void jspDestroy()
{
// Your cleanup code goes here.
}
```

Q.1. (*b*) Explain session, cookies.

Ans. A cookie is a small piece of text stored on a user's computer by their browser. Common uses for cookies are authentication, storing of site preferences, shopping cart items, and server session identification.

Each time the users' web browser interacts with a web server it will pass the cookie information to the web server. Only the cookies stored by the browser that relate to the domain in the requested URL will be sent to the server. This means that cookies that relate to www.example.com will not be sent to www.exampledomain.com.

In essence, a cookie is a great way of linking one page to the next for a user's interaction with a web site or web application.

A session can be defined as a server-side storage of information that is desired to persist throughout the user's interaction with the web site or web application.

Instead of storing large and constantly changing information via cookies in the user's browser, only a unique identifier is stored on the client side (called a "session id"). This session id is passed to the web server every time the browser makes an HTTP request (*i.e* a page link or AJAX request). The web application pairs this session id with it's

internal database and retrieves the stored variables for use by the requested page.

Q.2. What are custom tags? Explain.

Ans. Custom Elements enable developers to create their own custom HTML tags, let them use those tags in their sites and apps, and enable easier component reuse.

call document.registerElement()with its tag name as the first argument. var XComponent = document.registerElement('x-component');

Now you can use <x-component> wherever you want in the document.

<x-component></x-component>

Note that <x-component> can appear in the document before the definition of the custom element execution.

To detect the availability of Custom Elements, check ifdocument.registerElement is available. Otherwise, you can simply loadwebcomponents.js to polyfill it.

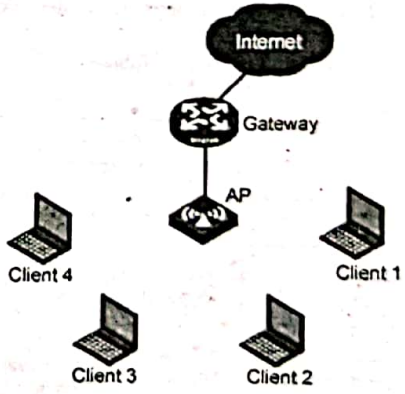
<script src="bower_components/webcomponentsjs/webcomponents.min.js"></script>

Q.3. (*a*) Explain WLAN.

A wireless local area network (WLAN) is a wireless distribution method for two or more devices that use high-frequency radio waves and often include an access point to the Internet. A WLAN allows users to move around the coverage area, often a home or small office, while maintaining a network connection.

A WLAN is sometimes call a local area wireless network (LAWN).

Every component that connects to a WLAN is considered a station and falls into one of two categories: access points (APs) and clients. APs transmit and receive radio frequency signals with devices able to receive transmitted signals; they normally function as routers. Clients may include a variety of devices such as desktop computers, workstations, laptop computers, IP phones and other cell phones and Smartphones. All stations able to communicate with each other are called basic service sets (BSSs), of which there are two types: independent and infrastructure. Independent BSSs (IBSS) exist when two clients communicate without using APs, but cannot connect to any other BSS. Such WLANs are called a peer-to-peer or an ad-hoc WLANs. The second BSS is called an infrastructure BSS. It may communicate with other stations but only in other BSSs and it must use APs.



Q.3. (b) What are document trees.

Ans. The Document Object Model (DOM) is a programming API for HTML and XML documents. It defines the logical structure of documents and the way a document is accessed and manipulated. In the DOM specification, the term "document" is used in the broad sense - increasingly, XML is being used as a way of representing many different kinds of information that may be stored in diverse systems, and much of this would traditionally be seen as data rather than as documents. Nevertheless, XML presents this data as documents, and the DOM may be used to manage this data.

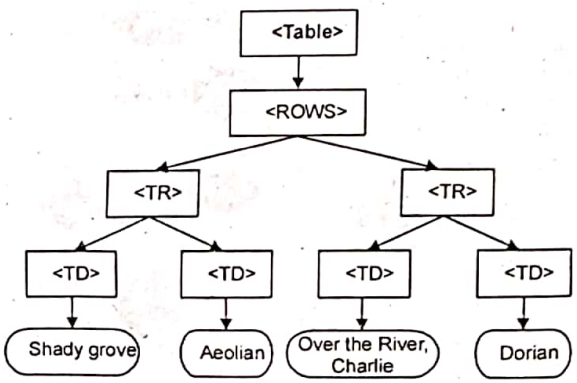
With the Document Object Model, programmers can create and build documents, navigate their structure, and add, modify, or delete elements and content. Anything found in an HTML or XML document can be accessed, changed, deleted, or added using the Document Object Model, with a few exceptions - in particular, the DOM interfaces for the internal subset and external subset have not yet been specified.

As a W3C specification, one important objective for the Document Object Model is to provide a standard programming interface that can be used in a wide variety of environments and applications. The Document Object Model can be used with any programming language.

The Document Object Model is a programming API for documents. The object model itself closely resembles the structure of the documents it models. For instance, consider this table, taken from an HTML document:

```
<TABLE>
<ROWS>
<TR>
<TD>Shady Grove</TD>
<TD>Aeolian</TD>
</TR>
<TR>
<TD>Over the River, Charlie</TD>
<TD>Dorian</TD>
</TR>
</ROWS>
</TABLE>
```

The Document Object Model represents this table like this:



Q.4. (a) What are built in objects?

Ans. The term "global objects" (or standard built-in objects) here is not to be confused with the **global object**. Here, global objects refer to **objects in the global scope. The global object** itself can be accessed using the this operator in the global scope. In fact, the global scope **consists of** the properties of the global object, including inherited properties.

JavaScript String Object: The String object is used to manipulate a stored piece of text.

Examples of use: The following example uses the length property of the String object to find the length of a string:

```
var txt="Hello world!"
document.write(txt.length)
```

The following example uses the toUpperCase() method of the String object to convert a string to uppercase letters:

```
var txt="Hello world!"
document.write(txt.toUpperCase())
```

The code above will result in the following output:

HELLO WORLD!

Complete String Object Reference: For a complete reference of all the properties and methods that can be used with the String object, go to our complete String object reference.

JavaScript Date Object

Defining Dates

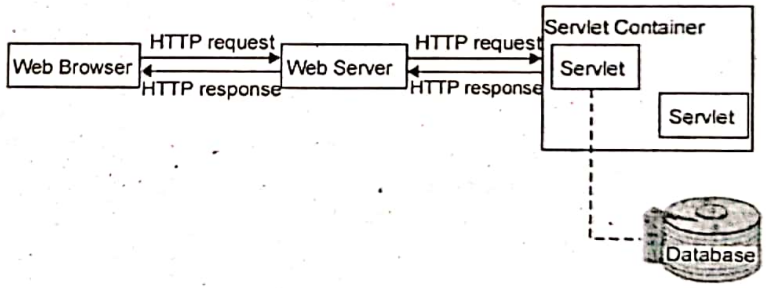
The Date object is used to work with dates and times.

We define a Date object with the new keyword. The following code line defines a Date object called myDate:

```
var myDate=new Date()
```

Note: The Date object will automatically hold the current date and time as its initial value!

Q.4. (b) Explain servlet architecture.



- Servlets read the explicit data sent by the clients (browsers). This includes an HTML form on a Web page or it could also come from an applet or a custom HTTP client program.
- Read the implicit HTTP request data sent by the clients (browsers). This includes cookies, media types and compression schemes the browser understands, and so forth.

• Process the data and generate the results. This process may require talking to a database, executing an RMI or Corba call, invoking a Web service, or computing the response directly.

• Send the explicit data (*i.e.*, the document) to the clients (browsers). This document can be sent in a variety of formats, including text (HTML or XML), binary (GIF images), Excel, etc.

• Send the implicit HTTP response to the clients (browsers). This includes telling the browsers or other clients what type of document is being returned (*e.g.*, HTML), setting cookies and caching parameters, and other such tasks.

MODEL PAPER-I
SIXTH SEMESTER (B. TECH.)
END TERM EXAMINATION
WEB ENGINEERING (ETCS-308)

Time. 3 Hrs. **MM: 75**

Note: *Question 1 is compulsory and attempt any 4 questions from the rest*

Q.1. (*a*) Explain CSS core syntax. **(2.5)**

Ans. CSS files are filled with styling statements like this. The Selector determines the name and where the style can be applied. The Declaration Block is contained between the curly braces - it contains all the styling Properties. As you can see, a ruling can have more than one property and if so they are separated by semi-colons. Here are some styles from the osstyle.css used to format theOpenSourcery.com files:

.gogreen { background-color: #00FF66; color: #000000; }

.indent { font-weight: bold; color:white; background-color: #6600CC; margin-left: 20px; }

.underbold { font-weight: bolder; text-decoration: underline; }

All of 3 of these styles have been used just above or below. .underbold is used at the beginning of this paragraph, . indent is used to create the purple box above, and .gogreen is used to highlight the comments below. It is important to note that in CSS there is no programming: no if-then constructs, no looping nor other programmatic statements. The only other legitimate CSS statement is a comment like so:

/* Here is a valid CSS comment and then a CSS statement - the comment may continue onto many lines until matched by a closing=> */

.serifBigBold { font-family: Georgia, Times, serif; font-size: 18px; font-weight: bold; color: white; }

/* Selector Declaration with 4 property:value(s) pairs */

of course, the green background above is just my own .gogreen CSS styling - and is nota part of a CSS comment. So with this you see the essential simplicity of CSS. One basic statement type with two elements. First, there is a Selector that determines where and to what the styling rule will apply - this is also called targetting the style. Second, there is a Declaration Block with a series of property:value pairs that specify what and how the styling is to be done. That's all Folks! You are done for CSS syntax.

Q.1. (*b*) Explain flowbox layout in CSS. **(2.5)**

Ans. All elements are treated as boxes by the browser's rendering engine (either "inline" or "block" boxes).

For example the margin is a transparent area around the box - the background of the box does not apply to it. It separates the box from other elements.

15 pixels on all sides:

margin: 15px;

10px on top, 5px right, 3px bottom, 20px left:

margin: 10px 5px 3px 20px;

10px on top:

margin-top: 10px;

If the margin is set to "auto" on a box that has a width, it will take up as much space as possible.

A centered box:

```
<div style="margin:auto; width:300px;">Hi</div>
```

Hi

A flush-right box:

```
<div style="margin-left:auto; margin-right:5px;
width:300px;"> Hi</div>
```

Q.1. (c) What do we understand by parameter data. **(2.5)**

Ans. The parameters are the way in which a client or user can send information to the Http Server. For example, in a login screen, we need to send to the server, the user and the password so that it validates them.

One of the nice features of Java servlets is that all of this form parsing is handled automatically. You simply call the getParameter method of the HttpServletRequest, supplying the parameter name as an argument. Note that parameter names are case sensitive. You do this exactly the same way when the data is sent via GET as you do when it is sent via POST. The return value is a String corresponding to the uudecoded value of the first occurrence of that parameter name. An empty String is returned if the parameter exists but has no value, and null is returned if there was no such parameter. If the parameter could potentially have more than one value, as in the example above, you should call getParameterValues instead of getParameter. This returns an array of strings. Finally, although in real applications your servlets probably have a specific set of parameter names they are looking for, for debugging purposes it is sometimes useful to get a full list. Use getParameterNames for this, which returns an Enumeration, each entry of which can be cast to a String and used in a getParameter call.

Q.1. (d) Explain CIDR. **(2.5)**

Ans. Classless Inter-Domain Routing, an IP addressing scheme that replaces the older system based on classes A, B, and C. With CIDR, a single IP address can be used to designate many unique IP addresses. A CIDR IP address looks like a normal IP address except that it ends with a slash followed by a number, called the *IP network prefix*. For example:

[172.200.0.0/16]

The IP network prefix specifies how many addresses are covered by the CIDR address, with lower numbers covering more addresses. An IP network prefix of /12, for example, can be used to address 1,048,576 former Class C addresses.

To illustrate the problems with the class system, consider that one of the most commonly used classes was Class B. An organization that needed more than 254 host machines would often get a Class B license, even though it would have far fewer than 65,534 hosts. This resulted in most of the block of addresses allocated going unused. The inflexibility of the class system accelerated IPv4 address pool exhaustion. With IPv6, addresses grow to 128 bits, greatly expanding the number of possible addresses on the Internet. The transition to IPv6 is slow, however, so IPv4 address exhaustion continues to be a significant issue.

CIDR reduced the problem of wasted address space by providing a new and more flexible way to specify network addresses in routers. CIDR lets one routing table entry represent an aggregation of networks that exist in the forward path that don't need to

be specified on that particular gateway. This is much like how the public telephone system uses area codes to channel calls toward a certain part of the network. This aggregation of networks in a single address is sometimes referred to as a supernet.

Using CIDR, each IP address has a network prefix that identifies either one or several network gateways. The length of the network prefix in IPv4 CIDR is also specified as part of the IP address and varies depending on the number of bits needed, rather than any arbitrary class assignment structure. A destination IP address or route that describes many possible destinations has a shorter prefix and is said to be less specific. A longer prefix describes a destination gateway more specifically. Routers are required to use the most specific, or longest, network prefix in the routing table when forwarding packets. (In IPv6, a CIDR block always gets 64 bits for specifying network addresses.)

Q.1. (e) . Explain WWW consortium. **(2.5)**

Ans. The World Wide Web Consortium (W3C) is an international community where Member organizations, a full-time staff, and the public work together to develop Web standards.

W3C does not have a typical organizational structure, nor is it incorporated. There are at least two ways to think about how W3C is organized:

1. in administrative terms
2. in process terms

In administrative terms: W3C is administered via a joint agreement among these "Host Institutions": MIT , ERCIM , Keio University, and Beihang University. The W3C staff (many of whom work physically at one of these institutions) is led by a Director and CEO. A small management team is responsible for resource allocation and strategic planning on behalf of the staff. Regional offices play an important role in W3C being an international organization.

In process terms: the W3C Process Document, Member Agreement, Patent Policy, and a few others documents establish the roles and responsibilities of the parties involved in the making of W3C standards. Some key components of the organization are.

- the Advisory Committee, composed of one representative from each W3C Member. The Advisory Committee has a number of review roles in the W3C Process, and they elect the Advisory Board and TAG.
- the Advisory Board, an advisory body elected by the Advisory Committee
- the Technical Architecture Group (TAG), which primarily seeks to document Web Architecture principles
- the W3C Director and CEO, who assess consensus for W3C-wide decisions
- the chartered groups, populated by Member representatives and invited experts, and which produce most of W3C's deliverables according to the steps of the W3C Process.

Q.1. (f) What is the use of HTML. **(2.5)**

Ans. HTML (HyperText Markup Language) is a descriptive language that is used to structure a webpage's content (*e.g.* text, images, links).

An HTML document is a text document containing tags, which must be properly used to describe the document's structure. Tags tell the browser something about how to display the document, and with tags you can also embed various media (such as images, video, sound) within the text.

The browser does not display the tags themselves. When users visit a webpage, their browser *parses* the document and *interprets* the tags in order to display the webpage correctly. For example, if there is an <img> tag within the document, the browser loads the associated image and displays that image in place of the HTML tag.

Q.1. (g) What are fuzzers? **(2.5)**

Fuzz testing or Fuzzing is a Black Box software testing technique, which basically consists in finding implementation bugs using malformed/semi-malformed data injection in an automated fashion.

Lets's consider an integer in a program, which stores the result of a user's choice between 3 questions. When the user picks one, the choice will be 0, 1 or 2. Which makes three practical cases. But what if we transmit 3, or 255 ? We can, because integers are stored a static size variable. If the default switch case hasn't been implemented securely, the program may crash and lead to "classical" security issues: (un)exploitable buffer overflows, DoS, ...

Fuzzing is the art of automatic bug finding, and it's role is to find software implementation faults, and identify them if possible.

A fuzzer is a program which injects automatically semi-random data into a program/ stack and detect bugs.

The data-generation part is made of generators, and vulnerability identification relies on debugging tools. Generators usually use combinations of static fuzzing vectors (known-to-be-dangerous values), or totally random data. New generation fuzzers use genetic algorithms to link injected data and observed impact.

The number of possible tryable solutions is the explorable solutions space. The aim of cryptanalysis is to reduce this space, which means finding a way of having less keys to try than pure bruteforce to decrypt something.

Most of the fuzzers are:

- protocol/file-format dependant
- data-type dependant

. •First, because the fuzzer has to connect to the input channel, which is bound to the target.

• Second, because a program only understands structured-enough data. If you connect to a web server in a raw way, it will only respond to listed commands such as GET (or eventually crash). It will take less time to start the string with "GET ", and fuzz the rest, but the drawback is that you'll skip all the tests on the first verb.

In this regard, Fuzzers try to reduce the number of unuseful tests, i.e. the values we already know that there's little chance they'll work: you reduce unpredictability, in favor of speed

Q.1. (h) Define HTTP. **(2.5)**

Ans. HTTP (Hypertext Transfer Protocol) is the set of rules for transferring files (text, graphic images, sound, video, and other multimedia files) on the World Wide Web. As soon as a Web user opens their Web browser, the user is indirectly making use of HTTP. HTTP is an application protocol that runs on top of the TCP/IP suite of protocols (the foundation protocols for the Internet).

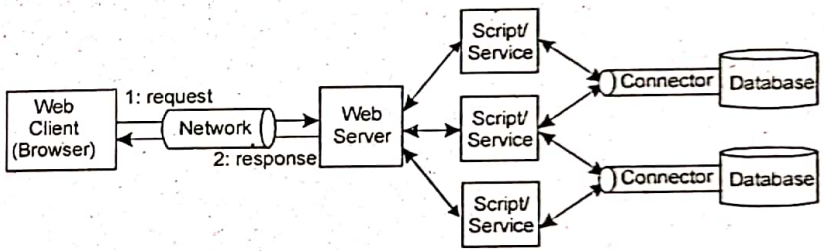
HTTP concepts include (as the Hypertext part of the name implies) the idea that files can contain references to other files whose selection will elicit additional transfer requests. Any Web server machine contains, in addition to the Web page files it can

serve, an HTTP daemon, a program that is designed to wait for HTTP requests and handle them when they arrive. Your Web browser is an HTTP client, sending requests to server machines. When the browser user enters file requests by either "opening" a Web file (typing in a Uniform Resource Locator or URL) or clicking on a hypertext link, the browser builds an HTTP request and sends it to the Internet Protocol address (IP address) indicated by the URL. The HTTP daemon in the destination server machine receives the request and sends back the requested file or files associated with the request.

Q.2. (a) Difference between web 2.0 and web 3.0 architecture. **(8)**

Ans. Web 2.0 describes World Wide Web sites that emphasize user-generated content, usability, and interoperability. Web 2.0 suggests a new version of the World Wide Web, it does not refer to an update to any technical specification, but rather to cumulative changes in the way Web pages are made and used.

A Web 2.0 site may allow users to interact and collaborate with each other in a social media dialogue as creators of user-generated content in a virtual community, in contrast to Web sites where people are limited to the passive viewing of content. Examples of Web 2.0 include social networking sites, blogs, wikis, folksonomies, video sharing sites, hosted services, Web applications, and mashups.



The server side functionality is partitioned in this figure. There are scripts and other services that the application, the web server is now accessing in order to create the complete web application and we see much more sophistication in terms of the data access as well.

We'll spend quite a bit of time to talk about how the server side is organized to create sophisticated web application architectures. One of the things we talk about are the design patterns that allow you to build more sophisticated web application architectures. By understanding these design patterns, you'll be better prepared to understand web applications in general. And the entire web stack, I should mention too, when moving to Web 2.0 and 3.0 applications. There's, there are many more standards that exist throughout the web stack. And this makes it much easier to build these web applications as well. So what do mean when we say the web stack? When I use the term web stack, I'm referring to the protocols, the standards and the technologies that exist throughout a web application architecture. And these standards are defined on the browser side. All the way to the deepest parts of the server side. What we see in Web 2.0 and 3.0 application architectures is that the browser is more capable and again with better standards support. If you think back to the Web 1.0 application days, you used to see web pages that said, best viewed with Netscape Navigator, or best viewed with Internet Explorer. You don't see those nowadays, and that's because browsers, in general, are better about complying to the standards that

have been created since that time. Let's look at the context for Web 2.0 applications. Here we see, by this point in time, the mid 2000 time frame, there are close to 100 million web pages out there, and a billion users. And we see that the web is a bit more interactive. There's more ability to write and save information on the web. When you went to a a site like Amazon, for example, you could save a wish list, or your previous purchases, and look at them. So you're able to put content up, up on the web. Every tweet that is created actually creates a webpage. So we'll see a lot more of that happening with Web 2.0 application architectures. The context for Web 3.0 application architectures is we're per, we're approaching now a billion indexed webpages on the world wide web. And we see that there's a lot more interactivity that's taking place.

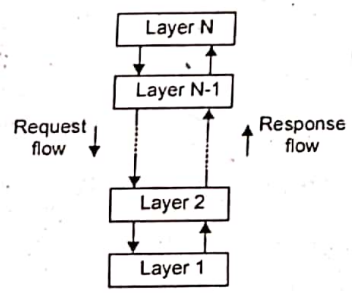
Q.2.(*b*). Explain distributed system architecture in detail. **(4.5)**

Ans. Distributed System Architecture Styles

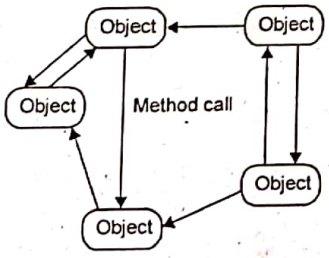
- Formulated in terms of components, and the way they are connected:
- A component is a modular unit with well-defined interfaces; replaceable; reusable
- A connector is a communication link between modules; mediates coordination or cooperation among components
- Four styles that are most important:

1. Layered architecture
2. Object-based architecture
3. Data-centered architecture: processes communicate through a common repository (passive or active).
4. Event-based architecture: processes communicate through the propagation of events

- Organize into logically different components, and distribute those components over the various machines.



(a)

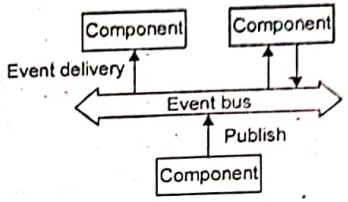


(b)

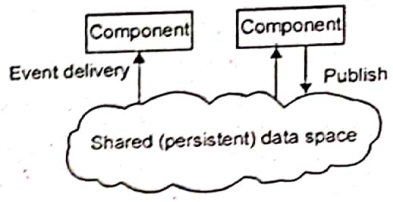
(*a*) Layered style is used for client-server system

(*b*) Object-based style for distributed object systems.

- less structured
- component = object
- connector = RPC or RMI
- Decoupling processes in space (anonymous) and also time (asynchronous) has led to alternative styles.



(a)



(b)

Event-based arch. supports several communication styles:

- Publish-subscribe
- Broadcast
- Point-to-point
- Decouples sender and receiver; asynchronous communication
- Event-driven architecture (EDA) promotes the production, detection, consumption of,and reaction to events.
- An event can be defined as "a significant change in state". For example, when a consumer purchases a car, the car's state changes from "for sale" to "sold". A car dealer's system architecture may treat this state change as an event to be produced, published, detected and consumed by various applications within the architecture.The main advantage of this architecture is that they are loosely coupled; they need not explicitly refer to each other. For example, If we have an alarm system that records information when the front door opens, the door itself doesn't know that the alarm system will add information when the door opens, just that the door has been opened.

Q.3(*a*). Describe about applications of web engineering in various systems. (10)

Modeling disciplines

- Design Manufacturing of Steel Plant equipments
- Process Modelling of Web applications
- Requirements Engineering for Web applications
- B2B applications

Design disciplines, tools and methods

- UML and the Web
- Conceptual Modeling of Web Applications (aka. Web modeling)
- Prototyping Methods and Tools
- Web design methods
- CASE Tools for Web Applications
- Web Interface Design
- Data Models for Web Information Systems

Implementation disciplines

- Integrated Web Application Development Environments
- Code Generation for Web Applications
- Software Factories for/on the Web
- Web 2.0, AJAX, E4X, ASP.NET, PHP and Other New Developments
- Web Services Development and Deployment

Testing disciplines

- Testing and Evaluation of Web systems and Applications
- Testing Automation, Methods and Tools

Applications categories disciplines

- Semantic Web applications
- Ubiquitous and Mobile Web Applications
- Mobile Web Application Development
- Device Independent Web Delivery
- Localization and Internationalization of Web Applications

Q.3. (*b*) Describe web application firewall. **(2.5)**

Ans. A WAF protects a Web application by controlling its input and output and the access to and from the application. Running as an appliance, server plug-in or cloud-based service, a WAF inspects every HTML, HTTPS, SOAP and XML-RPC data packet. Through customizable inspection, it is able to prevent attacks such as XSS, SQL injection, session hijacking and buffer overflows, which network firewalls and intrusion detection systems are often not capable of doing. A WAF is also able to detect and prevent new unknown attacks by watching for unfamiliar patterns in the traffic data.

A WAF can be either network-based or host-based and is typically deployed through a proxy and placed in front of one or more Web applications. In real time or near-real time, it monitors traffic before it reaches the Web application, analyzing all requests using a rule base to filter out potentially harmful traffic or traffic patterns. Web application firewalls are a common security control used by enterprises to protect Web applications against zero-day exploits, impersonation and known vulnerabilities and attackers.

WAFs started to gain attention when the PCI Security Standards Council formed and PCI DSS compliance was mandated by the credit card brands for merchants that process payment card transactions. PCI DSS requires that Web applications be fortified through either a code security review or a WAF.

Q.4(*a*) Define XHTML abstract syntax. **(8)**

Ans. Extensible Hypertext Markup Language (XHTML) is part of the family of XML markup languages. It mirrors or extends versions of the widely used Hypertext Markup Language (HTML), the language in which Web pages are formulated. Modularization provides an abstract collection of components through which XHTML can be subsetted and extended. The feature is intended to help XHTML extend its reach onto emerging platforms, such as mobile devices and Web-enabled televisions. It implements the following abstract modules: Base, Basic Forms, Basic Tables, Image, Link, Metainformation, Object, Style Sheet, and Target.

XHTML syntax is very similar to HTML syntax and almost all the valid HTML elements are valid in XHTML as well. But when you write an XHTML document, you need to pay a bit extra attention to make your HTML document compliant to XHTML.

Here are the important points to remember while writing a new XHTML document or converting existing HTML document into XHTML document "

- Write a DOCTYPE declaration at the start of the XHTML document.
- Write all XHTML tags and attributes in lower case only.
- Close all XHTML tags properly.
- Nest all the tags properly.
- Quote all the attribute values.
- Forbid Attribute minimization.
- Replace the name attribute with the id attribute.
- Deprecate the language attribute of the script tag.

Doctype Declaration: All XHTML documents must have a Doctype declaration at the start. There are three types of DOCTYPE declarations, which are discussed in detail

in XHTML Doctypes chapter. Here is an example of using DOCTYPE "

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

Case Sensitivity: XHTML is case sensitive markup language. All the XHTML tags and attributes need to be written in lower case only.

```
<!— This is invalid in XHTML —>
<A Href="/xhtml/xhtml_tutorial.html">XHTML Tutorial</A>
<!— Correct XHTML way of writing this is as follows —>
<a href="/xhtml/xhtml_tutorial.html">XHTML Tutorial</a>
```

In the example, Href and anchor tag A are not in lower case, so it is incorrect.

Closing the Tags: Each and every XHTML tag should have an equivalent closing tag, even empty elements should also have closing tags. Here is an example showing valid and invalid ways of using tags "

```
<!— This is invalid in XHTML —>
<p>This paragraph is not written according to XHTML syntax.
<!— This is also invalid in XHTML —>
<img src="/images/xhtml.gif" >
```

The following syntax shows the correct way of writing above tags in XHTML. Difference is that, here we have closed both the tags properly.

```
<!— This is valid in XHTML —>
<p>This paragraph is not written according to XHTML syntax.</p>
<!— This is also valid now —>
<img src="/images/xhtml.gif" />
```

Attribute Quotes: All the values of XHTML attributes must be quoted. Otherwise, your XHTML document is assumed as an invalid document. Here is the example showing syntax "

```
<!— This is invalid in XHTML —>
<img src="/images/xhtml.gif" width=250 height=50 />
<!— Correct XHTML way of writing this is as follows —>
<img src="/images/xhtml.gif" width="250" height="50" />
```

Attribute Minimization

XHTML does not allow attribute minimization. It means you need to explicitly state the attribute and its value. The following example shows the difference "

```
<!— This is invalid in XHTML —>
<option selected>
<!— Correct XHTML way of writing this is as follows —>
<option selected="selected">
```

Q.4.(*b*) Explain web application vulnerabilities and defenses. **(4.5)**

Ans. Applications are often installed with default settings that attackers can use to attack them. This is particularly an issue with third party software where an attacker has easy access to a copy of the same application or framework you are running. Hackers know the default account names and passwords. For example, looking at the contents you know that there's a default administrator account named 'admin' with the password 'secret'.

Configuration vulnerabilities also include features that increase attack surface. A common example is a feature that is on by default but you are not using, so you didn't configure it and the default configuration is vulnerable. It also includes debug features like status pages or dumping stack traces on failures.

Web threats can be divided into two primary categories, based on delivery method – push and pull. Push-based threats use spam, phishing, or other fraudulent means to lure a user to a malicious (often spoofed) website which then collects information and/or injects malware. Push attacks use phishing, DNS poisoning (or pharming), and other means to appear to originate from a trusted source.

Precisely-targeted push-based web threats are often referred to as spear phishing to reflect the focus of their data gathering attack. Spear phishing typically targets specific individuals and groups for financial gain. In other push-based web threats, malware authors use social engineering such as enticing subject lines that reference holidays, popular personalities, sports, pornography, world events and other hot topics to persuade recipients to open the email and follow links to malicious websites or open attachments with malware that accesses the Web.

Pull-based web threats are often referred to as "drive-by" threats by experts (and more commonly as "drive-by downloads" by journalists and the general public), since they can affect any website visitor. Cybercriminals infect legitimate websites, which unknowingly transmit malware to visitors or alter search results to take users to malicious websites. Upon loading the page, the user's browser passively runs a malware downloader in a hidden HTML frame (IFRAME) without any user interaction.

***(iii)* Trojan horse:** A Trojan horse, or Trojan, in computing is any malicious computer program which misrepresents itself to appear useful, routine, or interesting in order to persuade a victim to install it. The term is derived from the Ancient Greek story of the wooden horse that was used to help Greek troops invade the city of Troy by stealth.

Trojans are generally spread by some form of social engineering, for example where a user is duped into executing an e-mail attachment disguised to be unsuspicious, (e.g., a routine form to be filled in), or by drive-by download. Although their payload can be anything, many moderns forms act as a backdoor, contacting a controller which can then have unauthorized access to the affected computer. While Trojans and backdoors are not easily detectable by themselves, computers may appear to run slower due to heavy processor or network usage.

Q.6. Explain some latest trends in web technologies. **(12.5)**

Ans. 1. React: React isn't new, react means virtual DOM-diffing and unidirectional data flows are a proven pattern.

2. Flux: MVC is here to stay — just not on the client. JavaScript frameworks seem like they're in a constant state of reinvention. That can be a sign the problem is being solved the wrong way. Flux is simple, easy to work with, and pairs great with React.

3. Microservices: Loosely typed, dynamic languages are great for rapid iteration on the web, but it makes a lot of sense to split off parts of a large project into more focused, smaller applications. Break up your monolithic application with API-driven, modular components.

4. Not jQuery: Query has been the most important, impactful library in web development for the past decade, but manual DOM manipulation is on the way out. Instead, think about user interactions, data state, and components — then have something like React manage the rest for you.

5. Server side JSX: Node and JSX fit together naturally thanks to V8, but for teams not heavily invested in the Node ecosystem, we think there is room for interfaces between V8 and other languages.

6. Go: It feels simpler and safer than C and can pick up the slack where PHP falls short.

FIRST TERM EXAMINATION [MARCH 2016]
SIXTH SEMESTER [B.TECH]
WEB ENGINEERING [ETCS–308]

Time : 1½ Hrs. **M.M. : 30**

Note: *Q.No. 1 is compulsory. Attempt any two more Questions from the rest.*

Q.1. Attempt All **(10×1 = 10)**

Q.1. (*a*) Who is known as the father of World Wide Web?

Ans. Tim Berners-Lee is know as the father of world wide Web.

Q.1. (*b*) What is the standard port for HTTP connections?

Ans. The standard port for Hyper Text Transfer Protocol is 80.

Q.1. (*c*) Differentiate between DIV and SPAN.

Ans.) The difference between span and div is that a span element is in-line and usually used for a small chunk of HTML inside a line (such as inside a paragraph) whereas a div (division) element is block-line (which is basically equivalent to having a line-break before and after it) and used to group larger chunks of code.

Q.1. (*d*) What are the current technologies used to develop dynamic web applications?

Ans. The two current technologies for creating dynamic web pages are as follows :Ajax use a combination of both client-side scripting and server-side requests. It is a web application development technique for dynamically interchanging content, and it sends requests to the server for data in order to do so. Example- Google maps makes use of this technology.

Javascript is for adding dynamics to your website. You can do things like drop down menus and changing things after your page loads .

Q.1. (*e*) Which commond allows a client to remove a file from a Web Server using HTTP?

Ans. HTTP provides a method DELETE which allows a client to delete a file on the web server. An attacker can exploit it as a very simple and direct way to deface a web site or to mount a DoS attack.

The following example requests the server to delete the given file hello.htm at the root of the server:

DELETE /hello.htm HTTP/1.1
User-Agent: Mozilla/4.0
Host: www. Tutors. com

Q.1. (*f*) Define GET() and POST() methods.

Ans. A GET request retrieves data from a web server by specifying parameters in the URL portion of the request. This is the main method used for document retrieval. The following example makes use of GET method to fetch hello.htm:

GET /hello.htm HTTP/1.1
User-Agent: Mozilla/4.0
Host: www.tutors .com
Accept-Language: english

Accept-Encoding: gzip

The POST method is used when you want to send some data to the server, for example, file update, form data, etc. The following example makes use of POST method to send a form data to the server, which will be processed by a process.cgi and finally a response will be returned:

POST /cgi-bin/process.cgi HTTP/1.1
User-Agent: Mozilla/4.0
Host: www.tutorialspoint.com
Content-Type: text/xml; charset=utf-8
Content-Length: 88

Q.1. (g) Differentiate between XML and HTML.

Ans.

HTML	XML
HTML is an abbreviation for HyperText Markup Language.	XML stands for extensible Markup Language.
HTML is a markup language itself.	XML provides a framework for defining markup languages.
HTML was designed to display data with focus on how data looks.	XML was designed to be a software and hardware independent tool used to transport and store data, with focus on what data is.

Q.1. (h) What is WML Script used for?

Ans. WMLScript (Wireless Markup Language Script) is the client-side scripting language of WML (Wireless Markup Language). A scripting language is similar to a programming language, but is of lighter weight. With WML Script, the wireless device can do some of the processing and computation. This reduces the number of requests and responses to/from the server.

Q.1. (i) ..... is the space between the content and cell boundary in table cell.

Ans. The space between the table cells is controlled by the CELLSPACING attribute in the TABLE tag. By setting CELLSPACING to zero, you can remove all the space between the cells of your table.

Example-

<TABLE BGCOLOR = "#FFFFCC" BORDER=0 CELLSPACING = 0 >

Q.1. (j) Which Tag is used for making a drop-down list in HTML?

Ans. The <select> element is used to create a drop-down list.

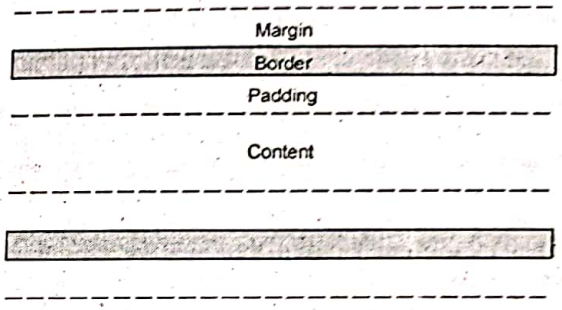
Example-

```
<select>
 <option value="volvo">Volvo</option>
 <option value="saab">Saab</option>
</select>
```

Q.2. (a) What is the purpose of CSS Box model and mention its parts also? (5)

Ans. a)All HTML elements can be considered as boxes. In CSS, the term "box model" is used when talking about design and layout.

The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content. The image below illustrates the box model:



Explanation of the different parts:

Content - The content of the box, where text and images appear

Padding - Clears an area around the content. The padding is transparent

Border - A border that goes around the padding and content

Margin - Clears an area outside the border. The margin is transparent

The box model allows us to add a border around elements, and to define space between elements.

Example

```
div {
  width: 300px;
  border: 25px solid green;
  padding: 25px;
  margin: 25px;
}
```

Q.2. (b) Compare Servelts with CGI. (5)

Ans. Both Java servlets and CGI are used for creating dynamic web applications that accept a user request, process it on the server side and return responses to the user. However, Java servlets provide a number of advantages over traditional CGI which are as follows:

• **Efficient:** Unlike traditional CGI where a new process is started for each client request, a servlet processes each request as a thread inside of a process. Thus servlets improve the performance as it removes the overhead of creating a new process for request every time.

Also, unlike CGI program which terminates after handling a request, the servlets remains in memory even after they complete a response and destroyed only when the servlet container is shutdown. Thus servlets make it easier to cache computations, keep database connections open and perform other optimizations that rely on persistent data.

• **Powerful:** Servlets support several capabilities that are difficult or impossible to accomplish with traditional CGI. These capabilities include talking directly to the web server, sharing data between multiple servlets, session tracking and caching of previous computations.

• **Portable:** Since servlets are written in the Java programming language and follow a standard API, so a servlet can be moved from one servlet compatible web server to another very easily. For example, servlets written for Apache web server can run unchanged on other severs such as Microsoft Internet Information (IIS) Server, IBM Web Sphere etc. On the other hand, CGI programs may be platform dependent, need to be recompiled or web server dependent.

• **Inexpensive:** A number of free or very inexpensive web servers are available these days. Once you have a web server, adding servlet support to it costs very little. On the other hand, CGI alternatives require a significant initial investment to purchase a proprietary package.

• **Language dependency:** Since servlets can be written only in Java so they are language dependent. On the other hand, CGI programs can be written in any programming language like C, C++, Perl, Visual Basic or even Java so they are language independent.

• **Secure:** As Java was designed from the ground up as a secure language, so a servlet can be run by a servlet engine or servlet container in restrictive sandbox just like an applet runs in a web browser's JVM, which increases the server security. On the other hand, CGI scripts are not subject to same degree of security sandboxing as Java servlets, they are significantly less secure.

• **Convenient:** Servlets have an extensive infrastructure for automatically parsing and decoding HTML form data, reading and setting HTTP headers, handling cookies, session handling etc. On the other hand, CGI does not support such infrastructure.

Q.3. (*a*) Write a program to validate the content of an HTML form. (5)

Ans. HTML form validation can be done by JavaScript.

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

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

The function can be called when the form is submitted:

```
HTML Form Example
<form name="myForm" action="demo_form.asp" onsubmit="return validate
Form()" method="post">
Name: <input type="text" name="fname">
<input type="submit" value="Submit">
</form>
```


Q.3. (*b*) Differentiate between POP3 and IMAP.

Ans.

POP3-POST OFFICE PROTOCOL	IMAP- Internet Messaging Access Protocol
You can use only one computer to check your email (no other devices)	You can use multiple computers and devices to check your email
Your mails are stored on the computer that you use	Your mails are stored on the server
Sent mail is stored locally on your PC, not on a mail server	Sent mail stays on the server so you can see it from any device.
The POP protocol, by default, is set to download all the messages from the e-mail server onto your computer. This means that all the actions performed on the messages (reading, moving, deleting...) will be performed on one's computer.	The IMAP protocol,by default, allows the user to keep all messages on the server. It constantly synchronizes the e-mail program with the server and displays what messages are currently present. All the actions performed on the messages (reading, moving, deleting...) will be done directly on the server.
Because everything is kept on the user's computer, the user will not be able to reopen messages from any location other than the computer where the messages have been downloaded.	
Messages are deleted on the desktop PC. Comparatively, it is inconvenient to clean up your mailbox on the server. Once e-mail is downloaded it can be accessed only using the same computer.	Message can be deleted directly on the server to make it more convenient to clean up your mailbox on the server. Because everything is kept on the server the user will be able to access the e-mail in the inbox from any computer in the world connected to the internet and can will always find the same settings in their e-mail account.

Q.4. (a) Explain different types of Style Sheets? (5)

Ans. When a browser reads a style sheet, it will format the HTML document according to the information in the style sheet.

There are three types of CSS styles:

Inline styles

Inline styles are styles that are written directly in the tag on the document. Inline styles affect only the tag they are applied to.An inline style may be used to apply a unique style for a single element.

To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

```
<a href="" style="text-decoration: none;">
```

Embedded styles

Embedded styles are styles that are embedded in the head of the document. Embedded styles affect only the tags on the page they are embedded in.An internal style sheet may be used if one single page has a unique style.

Internal styles are defined within the <style> element, inside the <head> section of an HTML page

```
<style type="text/css">
p { color: #00f; }
</style>
```

External styles

External styles are styles that are written in a separate document and then attached to various Web documents. External style sheets can affect any document they are attached to.With an external style sheet, you can change the look of an entire website by changing just one file.

Each page must include a reference to the external style sheet file inside the <link> element.

```
<link rel="stylesheet" type="text/css" href="styles.css" />
```

Q.4. (*b*) Wirte a program. **(5)**

(*i*) to add an ordered list in a web page

Ans. The ordered list element, <ol>, works very much like the unordered list element; individual list items are created in the same manner. The main difference between an ordered list and an unordered list is that with an ordered list, the order in which items are presented is important.

```
<ol>
 <li>Head north on N Halsted St</li>
 <li>Turn right on W Diversey Pkwy</li>
 <li>Turn left on N Orchard St</li>
</ol>
```

(ii) to create frames in a web page.

Ans. *The <frameset> tag is not supported in HTML5.*

The <frameset> tag defines a frameset.

The <frameset> element holds one or more <frame> elements. Each <frame> element can hold a separate document.

The <frameset> element specifies HOW MANY columns or rows there will be in the frameset, and HOW MUCH percentage/pixels of space will occupy each of them.

```
<frameset cols="25%,*,25%">
 <frame src="frame_a.htm">
 <frame src="frame_b.htm">
 <frame src="frame_c.htm">
</frameset
```

SECOND TERM EXAMINATION [APRIL 2016]
SIXTH SEMESTER [B.TECH]
WEB ENGINEERING [ETCS–308]

Time : 1½ Hrs. **M.M. : 30**

Note: *Q.No. 1 is compulsory. Attempt any two more Questions from the as rest.*

Q.1. Attempt All **(5×2 = 10)**

Q.1. (*a*) What are the different types of Secuirty Threats?

Ans. The types of computer security threats are as follows:–

1. Trojan. Trojan is one of the most complicated threats among all. It has the ability to hide itself from antivirus detection and steal important banking data to compromise your bank account. If the Trojan is really powerful, it can take over your entire security system as well.

2. Virus. It is a malicious program where it replicates itself and aim to only destroy a computer. The ultimate goal of a virus is to ensure that the victim's computer will never be able to operate properly or even at all.

3. Worms. One of the most harmless threats where it is program designed only to spread. It does not alter your system to cause you to have a nightmare with your computer, but it can spread from one computer to another computer within a network or even the internet.

4. Spyware. Is a Malware which is designed to spy on the victim's computer. If you are infected with it, probably your daily activity or certain activity will be spied by the spyware and it will find itself a way to contact the host of this malware.

Q.1. (*b*) Differentiate

(*i*) Cookies and DDOS

Ans. A "session" is set for maintaining the user data as the person is browsing through the site. A session is very useful in e-commerce websites, social networking sites etc. In PHP, session variables are used to set the sessions. Anything can be set / stored in a session like the user's id, username, some encrypted password string etc.

Example: $_SESSION['customer_name'] = 'John';

A "cookie" is however different from a session. It stores some information like the username, last visited pages etc. So that when the customer visits the site again, he may have the same environment set for him

Example: setcookie("username", "John", time()+2400);

(*ii*) DOS and DDOS

Ans. A DoS Attack is a Denial of Service attack.

This means that one computer and one internet connection is used to flood a server with packets (TCP / UDP). The point of such a denial of service attack is to overload the targeted server's bandwidth and other resources. This will make the server inaccessible to others, thereby blocking the website or whatever else is hosted there.

DDoS

A DDoS Attack is a Distributed Denial of Service Attack.

In most respects it is similar to a DoS attack but the results are much, much different. Instead of one computer and one internet connection the DDoS attack utilises many computers and many connections. The computers behind such an attack are often

distributed around the whole world and will be part of what is known as a botnet. The main difference between a DDoS attack vs a DoS attack, therefore, is that the target server will be overload by hundreds or even thousands of requests in the case of the former as opposed to just one attacker in the case of the latter.

Q.1. (c) Define virus, worms, trapdoor, trojan

Ans. VIRUS : It is a malicious program where it replicates itself and aim to only destroy a computer. The ultimate goal of a virus is to ensure that the victim's computer will never be able to operate properly

WORM: It does not alter your system to cause you to have a nightmare with your computer, but it can spread from one computer to another computer within a network or even the internet.

TRAPDOOR: A computer trapdoor, also known as a back door, provides a secret — or at least undocumented — method of gaining access to an application, operating system or online service.

TROJAN: is one of the most complicated threats among all. It has the ability to hide itself from antivirus detection and steal important banking data to compromise your bank account. If the Trojan is really powerful, it can take over your entire security system as well.

Q.1. (d) Explain HTTPS.

Ans. HTTPS: also called HTTP over TLS,HTTP over SSL,and HTTP Secure is a protocol for secure communication over a computer network which is widely used on the Internet. HTTPS consists of communication over Hypertext Transfer Protocol (HTTP) within a connection encrypted by Transport Layer Security or its predecessor, Secure Sockets Layer. The main motivation for HTTPS is authentication of the visited website and protection of the privacy and integrity of the exchanged data.

Q.1. (e) Compare Web 1.0, Web2.0 and Web3.0

Ans. Web 1.0

It is the "readable" phrase of the World Wide Web with flat data. In Web 1.0, there is only limited interaction between sites and web users. Web 1.0 is simply an information portal where users passively receive information without being given the opportunity to post reviews, comments, and feedback.

Web 2.0

It is the "writable" phrase of the World Wide Web with interactive data. Unlike Web 1.0, Web 2.0 facilitates interaction between web users and sites, so it allows users to interact more freely with each other. Web 2.0 encourages participation, collaboration, and information sharing. Examples of Web 2.0 applications are Youtube, Wiki, Flickr, Facebook, and so on.

Web 3.0

It is the "executable" phrase of World Wide Web with dynamic applications, interactive services, and "machine-to-machine" interaction. Web 3.0 is a semantic web which refers to the future. In Web 3.0, computers can interpret information like humans and intelligently generate and distribute useful content tailored to the needs of users.

Q.2. (a) Explain different application of Plug-in. (2×5=10)

Ans. Applications support plug-ins for many reasons. Some of the main reasons include:

To enable third-party developers to create abilities which extend an application

To support easily adding new features

To reduce the size of an application

To separate source code from an application because of incompatible software licenses.

Types of applications and why they use plug-ins:

(*i*) Audio editors use plug-ins to generate, process or analyse sound. Ardour and Audacity are examples of such editors.

(*ii*) Email clients use plug-ins to decrypt and encrypt email. Pretty Good Privacy is an example of such plug-ins.

(*iii*) Graphics software use plug-ins to support file formats and process images. (c.f. Photoshop plugin)

(*iv*) Media players use plug-ins to support file formats and apply filters. foobar2000, GStreamer, Quintessential, VST, Winamp, XMMS are examples of such media players.

(*v*) Packet sniffers use plug-ins to decode packet formats. OmniPeek is an example of such packet sniffers.

(*vi*) Remote sensing applications use plug-ins to process data from different sensor types; e.g., Opticks.

(*vii*)Text editors and Integrated development environments use plug-ins to support programming languages or enhance development process e.g., Visual Studio, RAD Studio, Eclipse, IntelliJ IDEA, jEdit and MonoDevelop support plug-ins.

(*viii*) Web browsers use browser extensions to expand their functionality. Examples include Adobe Flash Player, Java SE, QuickTime, Microsoft Silverlight and Unity.

Q.2. (b) What is seven layer security model?

Ans. The Open Systems Interconnect (OSI) model has seven layers.

The layers are stacked this way:

Application
Presentation
Session
Transport
Network
Data Link
Physical

PHYSICAL LAYER: The physical layer, the lowest layer of the OSI model, is concerned with the transmission and reception of the unstructured raw bit stream over a physical medium. It describes the electrical/optical, mechanical, and functional interfaces to the physical medium, and carries the signals for all of the higher layers.

DATA LINK LAYER: The data link layer provides error-free transfer of data frames from one node to another over the physical layer, allowing layers above it to assume virtually error-free transmission over the link.

NETWORK LAYER: The network layer controls the operation of the subnet, deciding which physical path the data should take based on network conditions, priority of service, and other factors.

TRANSPORT LAYER: The transport layer ensures that messages are delivered error-free, in sequence, and with no losses or duplications. It relieves the higher layer protocols from any concern with the transfer of data between them and their peers.

SESSION LAYER: The session layer allows session establishment between processes running on different stations.

PRESENTATION LAYER: The presentation layer formats the data to be presented to the application layer. It can be viewed as the translator for the network. This layer may translate data from a format used by the application layer into a common format at the sending station, then translate the common format to a format known to the application layer at the receiving station.

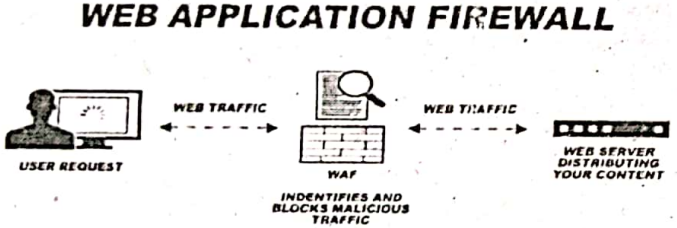
APPLICATION LAYER: The application layer serves as the window for users and application processes to access network services.

Q.3. (a) Explain Web application Firewalls and its types. **(2×5 = 10)**

Ans. A Web application firewall (WAF) is a firewall that monitors, filters or blocks the HTTP traffic to and from a Web application.

A WAF protects a Web application by controlling its input and output and the access to and from the application.A WAF is also able to detect and prevent new unknown attacks by watching for unfamiliar patterns in the traffic data.A WAF can be either network-based or host-based and is typically deployed through a proxy and placed in front of one or more Web applications.

Some examples of Web application firewalls include Citrix Systems Inc.'s NetScaler AppFirewall.



The types are as follows:

(i) ModSecurity (Trustwave SpiderLabs): Is one of the oldest and widely used open source web application firewall which can detect application level threats on internet, and provides security against a range of security issues to web applications.

(ii) AQTRONIX WebKnight: An open source application firewall designed specifically for web servers and IIS, and it is licensed through the GNU – General Public License. It provides the features of buffer overflow, directory traversal, encoding and SQL injection to identify / restrict the attacks.

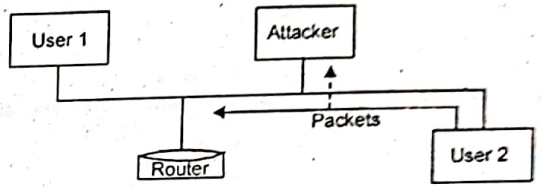
(iii) ESAPI WAF: Designed to provide protection at the application layer instead of network layer. It is a Java based WAF which provides complete security from online attacks. Some of the unique features of the solution include outbound filtering features which reduce information leakage.

(iv) WebCastellum: Is a Java based web application firewall which can protect application against cross site scripting.

(v) Binarysec: Is web application software firewall, and it protects applications against illegitimate HTTP and blocks suspicious requests as well. It provides protection against cross site scripting, command injections, parameter tampering, buffer overflow, directory traversal, SQL injection and attack obstruction.

Q.3. (b) Explain Packet sniffing, Packet spoofing and Phising.

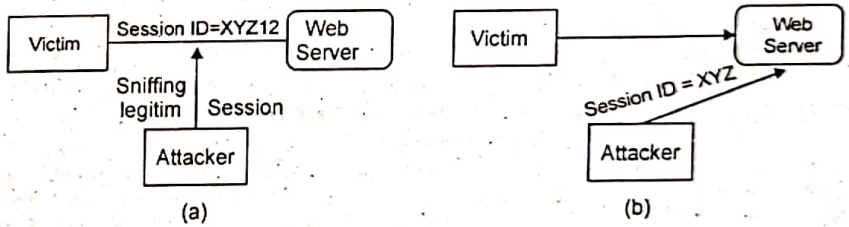
Ans. PACKET SNIFFING: A packet analyzer (also known as a network analyzer, protocol analyzer or packet sniffer—or, for particular types of networks, an Ethernet sniffer or wireless sniffer) is a computer program or piece of computer hardware that can intercept and log traffic that passes over a digital network or part of a network. As data streams flow across the network, the sniffer captures each packet and, if needed, decodes the packet's raw data, showing the values of various fields in the packet, and analyzes its content.



PACKET SPOOFING: Packet spoofing or IP spoofing is the creation of Internet Protocol (IP) packets having a source IP address, with the purpose of concealing the identity of the sender or impersonating another computing system.

A spoofing attack is when a malicious party impersonates another device or user on a network in order to launch attacks against network hosts, steal data, spread malware, or bypass access controls.

The attacker creates a IP packet and sends to the server which is known as SYN request. The difference in the IP packet and normal packet is that the attacker puts the own source address as another computers IP address in the newly created IP packet. The server responds back with a SYN ACK response which travels to the forged IP address. The attacker somehow gets this SYN ACK response sent by the server and acknowledges it so as to complete a connection with the server.



PHISHING : Phishing is the attempt to obtain sensitive information such as usernames, passwords, and credit card details (and sometimes, indirectly, money), often for malicious reasons, by masquerading as a trustworthy entity in an electronic communication.Phishing is typically carried out by email spoofing or instant messaging, and it often directs users to enter details at a fake website whose look and feel are almost identical to the legitimate one.

Q.4. (a) What are the different applications of WE Technologies in distributed system? **(2×5=10)**

Ans. Distributed system is a a system in which hardware or software components located at networked computers communicate and coordinate their actions only by message passing defines it and it can also be defined by a collection of independent computers that appear to the users of the system as a single computer.

There are various types of distributed systems, such as Clusters , Grids , P2P (Peer-to-Peer) networks , distributed storage systems and so on.

Over the years, technologies such as CORBA and DCOM have provided the means to build distributed component-based systems. Such technologies allow systems to interoperate at the component level, by providing asoftware layer and protocols that offer the interoperability needed for components developed in different programming languages to exchange messages. However, such technologies present scalability issues when applied to, for instance, the Internet and some restrict the developer to aspecific programming language. Hence, approaches based on Web protocols and XML(eXtensible Markup Language) have been proposed toallow interoperable distributed systems irrespective the programming language in which they are developed.Web Services are based on XML and provide a means to develop distributed systems that follow a Service Oriented Architecture (SOA).

Services are described in an XML-based dialect (WSDL). In a similar fashion, the request and reply messages exchangedin such systems are formatted according to the Simple Object Access Protocol (SOAP).SOAP messages can be encoded and transmitted by using Web protocols such as theHypertext Transfer Protocol (HTTP). Various industrial technologies and applicationplatforms such as .NET from Microsoft, J2EE from Sun, WebSphere from IBM aretargeted at supporting the development of applications based on Web Services. .

Q.4. (*b*) How HTTPS cerificate works for authentication?

Ans. HTTPS Client Authentication requires the client to possess a Public Key Certificate (PKC). If you specify client authentication, the web server will authenticate the client using the client's public key certificate.

HTTPS Client Authentication is a more secure method of authentication than either basic or form-based authentication. It uses HTTP over SSL (HTTPS), in which the server authenticates the client using the client's Public Key Certificate (PKC). Secure Sockets Layer (SSL) technology provides data encryption, server authentication, message integrity, and optional client authentication for a TCP/IP connection. You can think of a public key certificate as the digital equivalent of a passport. It is issued by a trusted organization, which is called a certificate authority (CA), and provides identification for the bearer.

Before using HTTP Client Authentication, you must make sure that the following actions have been completed:

(*i*) Make sure the client has a valid Public Key Certificate. For more information on creating and using public key certificates, read Working with Digital Certificates.

(*ii*) Make sure that SSL support is configured for your server. If your server is the Sun GlassFishEnterprise Server v3, SSL support is already configured. If you are using another server, consult the documentation for that server for information on setting up SSL support. More information on configuring SSL support on the application server can be found in Establishing a Secure Connection Using SSL and the Sun GlassFish Enterprise Server v3 Administration Guide.

The following example shows how to declare HTTPS client authentication in your deployment descriptor:

```
<login-config>
  <auth-method>CLIENT-CERT</auth-method>
</login-config>
```

END TERM EXAMINATION [MAY 2016]
SIXTH SEMESTER [B.TECH]
WEB ENGINEERING [ETCS–308]

Time : 3 Hrs. **M.M. : 75**

Note: *Attempt any five questions including Q.No.1 which is compulsory.*

Q.1. Attempt All

Q.1. (*a*) How does XHTML differ from HTML? **(5)**

Ans.

HTML	XHTML
HTML or HyperText Markup Language the main markup language for creating web pages and other information that can be displayed in a web browser.	XHTML (Extensible HyperText Language) is a family of XML markup languages that mirror or extend versions of the widely used Hypertext Markup Language (HTML), the language in which web pages are written.
Filename extension .html, .htm	**Filename extension** xhtml, .xht, .xml, .html, .htm
Internet media type text/html	application/xhtml+xml
Developed by W3C &WHATWG	World Wide Web Consortium
Type of format Document file format	Markup language
Function Web pages are written in HTML.	Extended version of HTML that is stricter and XML-based.
Origin Proposed by Tim Berners-Lee in 1987.	World Wide Web Consortium Recommendation in 2000.
Nature Flexible framework requiring lenient HTML specific parser.	Restrictive subset of XML and needs to be parsed with standard XML parsers.
Application Application of Standard Generalized Markup Language (SGML).	Application of XML

Q.1. (*b*) Differentiate between server side scripting and client side scripting. **(5)**

Ans. There are two main ways to customise Web pages and make them more interactive. The two are often used together because they do very different things.

Scripts: A script is a set of instructions. For Web pages they are instructions either to the Web browser (client-side scripting) or to the server (server-side scripting). These are explained more below.

Scripts provide change to a Web page. Think of some Web pages you have visited. Any page which changes each time you visit it (or during a visit) probably uses scripting.

Client-side: The client is the system on which the Web browser is running. JavaScript is the main client-side scripting language for the Web. Client-side scripts are interpreted by the browser. The process with client-side scripting is:

- The user requests a Web page from the server
- The server finds the page and sends it to the user
- The page is displayed on the browser with any scripts running during or after display

So client-side scripting is used to make Web pages change after they arrive at the browser. It is useful for making pages a bit more interesting and user-friendly. It can also provide useful gadgets such as calculators, clocks etc. but on the whole is used for appearance and interaction.

Server-side: The server is where the Web page and other content lives. The server sends pages to the user/client on request. The process is:

- The user requests a Web page from the server
- The script in the page is interpreted by the server creating or changing the page content to suit the user and the occasion and/or passing data around
- The page in its final form is sent to the user and then cannot be changed using server-side scripting

Server-side scripting tends to be used for allowing users to have individual accounts and providing data from databases. It allows a level of privacy, personalisation and provision of information that is very powerful. E-commerce, MMORPGs and social networking sites all rely heavily on server-side scripting. PHP and ASP.net are the two main technologies for server-side scripting.

Q.1. (c) Explain the usage of plugins, extensions, and web apps. (5)

Ans. PLUG-INS: In computing, a plug-in (or plugin, add-in, addin, add-on, addon, or extension) is a software component that adds a specific feature to an existing computer program. When a program supports plug-ins, it enables customization. The common examples are the plug-ins used in web browsers to add new features such as search-engines, virus scanners, or the ability to use a new file type such as a new video format. Well-known browser plug-ins include the Adobe Flash Player, the QuickTime Player, and the Java plug-in, which can launch a user-activated Java applet on a web page to its execution on a local Java virtual machine.

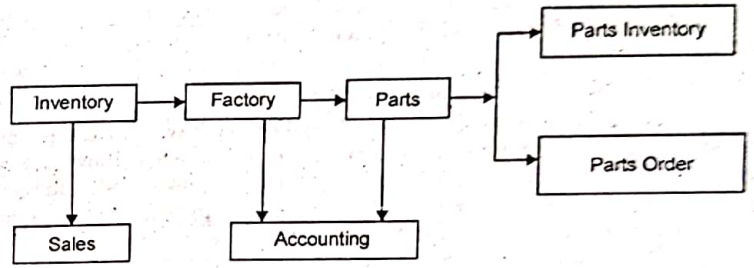
EXTENSIONS: A browser extension is a plug-in that extends the functionality of a web browser in some way. Some extensions are authored using web technologies such as HTML, JavaScript, and CSS.Browser extensions can change the user interface of the web browser without directly affecting viewable content of a web page; for example, by adding a "toolbar."Browser extensions are most commonly used for improving security, accessibility, blocking advertisements, and generally improving a browser's user interface and adding various other features to make browsing the internet more easy and pleasant.

WEB APPS: In computing, a web application or web app is a client–server software application in which the client (or user interface) runs in a web browser.Common web applications include webmail, online retail sales, online auctions, wikis, instant messaging services and many other functions.

Web sites most likely to be referred to as "web applications" are those which have similar functionality to a desktop software application, or to a mobile app. HTML5 introduced explicit language support for making applications that are loaded as web pages, but can store data locally and continue to function while offline.

Q.1. (d) How does the JMS API Work with the JAVA EE Platform? (5)

Ans. The Message Service is a Java API that allows applications to create, send, receive, and read messages. Designed by Sun and several partner companies, the JMS API defines a common set of interfaces and associated semantics that allow programs written in the Java programming language to communicate with other messaging implementations.



When the JMS API was introduced in 1998, its most important purpose was to allow Java applications to access existing messaging-oriented middleware (MOM) systems, such as MQSeries from IBM. Since that time, many vendors have adopted and implemented the JMS API, so a JMS product can now provide a complete messaging capability for an enterprise.

Beginning with the 1.3 release of the Java EE platform, the JMS API has been an integral part of the platform, and application developers have been able to use messaging with Java EE components.

The JMS API in the Java EE platform has the following features:Application clients, Enterprise JavaBeans (EJB) components, and web components can send or synchronously receive a JMS message. Application clients can in addition receive JMS messages asynchronously. (Applets, however, are not required to support the JMS API.) Message-driven beans, which are a kind of enterprise bean, enable the asynchronous consumption of messages. A JMS provider can optionally implement concurrent processing of messages by message-driven beans.

The JMS API enhances the Java EE platform by simplifying enterprise development, allowing loosely coupled, reliable, asynchronous interactions among Java EE components and legacy systems capable of messaging. A developer can easily add new behavior to a Java EE application that has existing business events by adding a new message-driven bean to operate on specific business events.

Q.1. (e) List two advantages and two disadvantages of dynamic script loading (5)

Ans. Some of the benefits include :

- It reduces the load on the user's computer, as it does not require plugins or browser scripting technology (such as Javascript).
- You can use it to dynamically create pages on the fly. New pages can even be instantly created based on certain user interaction.

Some disadvantages are:

- It requires the scripting software to be installed on the server
- Many scripts and CMS tools require databases in order to store dynamic data

The nature of dynamic scripts creates new security concerns, in some cases making it easier for hackers to gain access to servers exploiting code flaws.

Q.2. (a) Discuss the application of Web Engineering Technologies in distributed systems. **(6)**

Ans. Distributed system is a a system in which hardware or software components located at networked computers communicate and coordinate their actions only by message passing defines it and it can also be defined by a collection of independent computers that appear to the users of the system as a single computer.

There are various types of distributed systems, such as Clusters , Grids , P2P (Peer-to-Peer) networks , distributed storage systems and so on.

Over the years, technologies such as CORBA and DCOM have provided the means to build distributed component-based systems. Such technologies allow systems to interoperate at the component level, by providing a software layer and protocols that offer the interoperability needed for components developed in different programming languages to exchange messages. However, such technologies present scalability issues when applied to, for instance, the Internetand some restrict the developer to aspecific programming language. Hence, a pproaches based on Web protocols and XML(eXtensible Markup Language) have been proposed toallow interoperable distributed systems irrespective the programming language in which they are developed.Web Services are based on XML and provide a means to develop distributed systems that follow a Service Oriented Architecture (SOA).

Services are described in an XML-based dialect (WSDL). In a similar fashion, the request and reply messages exchanged in such systems are formatted according to the Simple Object Access Protocol (SOAP).SOAP messages can be encoded and transmitted by using Web protocols such as the Hypertext Transfer Protocol (HTTP). Various industrial technologies and application platforms such as .NET from Microsoft, J2EE from Sun, WehSphere from IBM are targeted at supporting the development of applications based on Web Services.

Q.2. (b) Explain and discuss the various issues in WEB Security. **(6.5)**

Ans. Web security is very complex – with a lot of unknowns. As an executive running a business with a lot of moving parts, I'm sure you can relate. There are numerous areas – both operational and technical – where web security is lacking in practically every organization regardless of skills and budget.

1. Untested systems: The web security focus is typically on the latest applications essential to running the business or increasing sales. The thing is, there are other (likely dozens) of other websites and applications running in your environment that are creating as much, if not more, business risk simply because they have not been tested for vulnerabilities.

2. Production data being used in development and QA: Developers and QA professionals often use a copy of production databases when writing and testing their code. It's typically an honest oversight but it can have serious ramifications. The thing is, the systems they're running are often under-secured.

3. Exposed source code: Developers are some of the smartest and most reasonable people I work with. Yet they're still human and make mistakes like everyone else.

4.Weak passwords: It's the bane of web security. You can have the most secure code, strongest encryption, and the fanciest web application firewall, yet all it takes to expose your entire application (and data) to the world is one weak password. Often times, developers will build in strong password enforcement features.

5. Input validation flaws: On the more technical side of web security are websites and applications that do not "cleanse" or validate what's entered into the URL or form fields. When these flaws exist, things like cross-site scripting and SQL injection can occur which allow an attacker or malware to manipulate the vulnerable web pages and gain access to things like local web browser information, user login sessions, or even the entire database.

Q.3. Create a HTML form that has the following controls. **(12.5)**

- **A TEXT control called firstName to collect the first name.**
- **A TEXT control called lastName to collect the last name.**
- **A TEXT control called email to collect the email address.**
- **A TEXT control called phone to collect the phone number.**
- **A SELECT control called software for displaying a combo box with software list.**
- **A SELECT control called os for displaying a combo box with operating systems.**
- **A TEXTAREA control called txtArea for displaying problem description.**
- **A SUBMIT control called submit for submitting the information.**

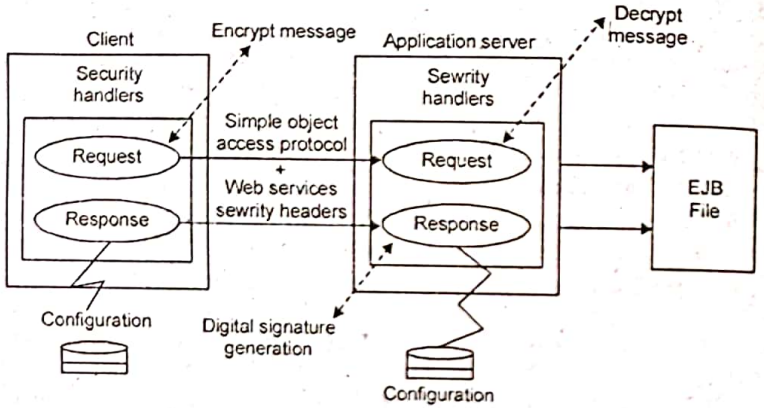
Ans.

```
<html>
<body>
<form action="action_page.php">
First name:<br>
<input type="text" name="firstname"><br>
Last name:<br>
<input type="text" name="lastname">
Email:<br>
<input type="text" name="email"><br>
Phone:<br>
<input type="text" name="phone"><br>
<select>
 <option value="software">MS WORD</option>
 <option value="software">MS EXCEL</option>
 <option value="software">ORACLE</option>
 <option value="software">MYSQL</option>
</select>
<select>
 <option value="os">windows</option>
 <option value="os">linux</option>
 <option value="os">mac</option>
 </select>
<input type="submit" value="Submit">
```

</form>
</body>
</html>

Q.4. (a) Discuss and explain Web Secuirty model in detail. **(6.5)**

Ans.



Web security has two sides:

• Web Browser (client side): Attacks target browser security weakness uninstallation which results in malware uninstallation and loss of private data.

• Web Application code (server side): Writtern in PHP, ASP, ----- Attacks lead to defaced sites.

So a browser is used as a security interface. HTTP protocol is simple, stateless and unencrypted which provides HTTP request and HTTP response as a security mechanism. Various components of browser security are:

1. FRAMCE to FRAME RELATIONSHIPS: can script (A, B)
2. FRAME to COOKIE RELATIONSHIPS: Read cookie (A, B) write cookie (A, S)
3. Security Indicator (W): (SSl lock icon)

Q.4. (b) Explain the following: **(6)**

(i) HTTPS and certificates

(ii) HTTP security extensions

Ans. HTTPS and Certificates: Hyper Text Transfer Protocol Secure (HTTPS) is the secure version of HTTP, the protocol over which data is sent between your browser and the website that you are connected to. The 'S' at the end of HTTPS stands for 'Secure'. It means all communications between your browser and the website are encrypted. HTTPS is often used to protect highly confidential online transactions like online banking and online shopping order forms. Web browsers such as Internet Explorer, Firefox and Chrome also display a padlock icon. When you request a HTTPS connection to a webpage, the website will initially send its SSL certificate to your browser. This certificate contains the public key needed to begin the secure session. Based on this initial exchange, your browser and the website then initiate the 'SSL handshake'. The SSL handshake involves the generation of shared secrets to establish a uniquely secure connection between yourself and the website. When a trusted SSL(Secure Socket Layer) Digital Certificate

is used during a HTTPS connection, users will see a padlock icon in the browser address bar. All communications sent over regular HTTP connections are in 'plain text' and can be read by any hacker that manages to break into the connection between your browser and the website. This presents a clear danger if the 'communication' is on an order form and includes your credit card details or social security number. With a HTTPS connection, all communications are securely encrypted. This means that even if somebody managed to break into the connection, they would not be able decrypt any of the data which passes between you and the website.

HTTPS security exteusious: A Reporting Services security extension enables the authentication and authorization of users or groups; that is, it enables different users to log on to a report server and, based on their identities, perform different tasks or operations. By default, Reporting Services uses a Windows-based authentication extension, which uses Windows account protocols to verify the identities of users who claim to have accounts on the system. Reporting Services uses a role-based security system to authorize users. The Reporting Services role-based security model is similar to the role-based security models of other technologies.

Because security extensions are based on an open and extensible API, you can create new authentication and authorization extensions in Reporting Services. The following is an example of a typical security extension implementation that uses Forms-based authentication and authorization:

It is recommended that you use Windows Authentication if at all possible. However, custom authentication and authorization for Reporting Services may be appropriate in the following two cases:

• You have an Internet or extranet application that cannot use Windows accounts.

• You have custom-defined users and roles and need to provide a matching authorization scheme in Reporting Services.

Q.5. (a) Explain the following in respect to JAVA **(12.5)**

Ans. Java servelets: A **Java servlet** is a Java program that extends the capabilities of a server. Although servlets can respond to any types of requests, they most commonly implement applications hosted onWeb servers. Such Web servlets are the Java counterpart to other dynamic Web content technologies such as PHP and ASP.NET. To deploy and run a servlet, a web container must be used. A web container (also known as a servlet container) is essentially the component of a web server that interacts with the servlets. The web container is responsible for managing the lifecycle of servlets, mapping a URL to a particular servlet and ensuring that the URL requester has the correct access rights.The Servlet API, contained in the Java package hierarchy javax.servlet, defines the expected interactions of the web container and a servlet. A Servlet is an object that receives a request and generates a response based on that request. The basic Servlet package defines Java objects to represent servlet requests and responses, as well as objects to reflect the servlet's configuration parameters and execution environment. The packagejavax.servlet.http defines HTTP-specific subclasses of the generic servlet elements, including session management objects that track multiple requests and responses between the web server and a client. Servlets may be packaged in a WAR file as a web application.

Q.5. (b) Intrinsic event handling

Ans. Intrinsic event handlers are ways to attach specific scripts to your documents that are executed only when something happens to an element. Not all event handlers apply to all elements, but here's the lot:

Intrinsic event handlers:

• ONLOAD (Script)

This event occurs when the browser finishes loading a document or a all frames in a frameset. It applies to BODY and FRAMESET elements.

• ONUNLOAD (Script)

This event occurs when the browser stops displaying a document or a frame. It applies to BODY and FRAMESET elements.

•ONCLICK (Script)

This event occurs when a mouse button is clicked over an element.

• ONDBLCLICK (Script)

This event occurs when a mouse button is double-clicked over an element.

• ONMOUSEDOWN (Script)

In order to use intrinsic event handlers, you *must* define the default scripting language using the Content-Style-Type header. Most scripting languages, including JavaScript, offer a way to define event handlers in scripts instead of using HTML attributes, and this is a better approach from a design point of view.

Q.5. (c) JSP

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

There are many advantages of JSP over servlet. They are as follows:

(1) Extension to Servlet: JSP technology is the extension to servlet technology. We can use all the features of servlet in JSP. In addition to, we can use implicit objects, predefined tags, expression language and Custom tags in JSP, that makes JSP development easy.

(2) Easy to maintain: JSP can be easily managed because we can easily separate our business logic with presentation logic. In servlet technology, we mix our business logic with the presentation logic.

(3) Fast Development: No need to recompile and redeploy: If JSP page is modified, we don't need to recompile and redeploy the project. The servlet code needs to be updated and recompiled if we have to change the look and feel of the application.

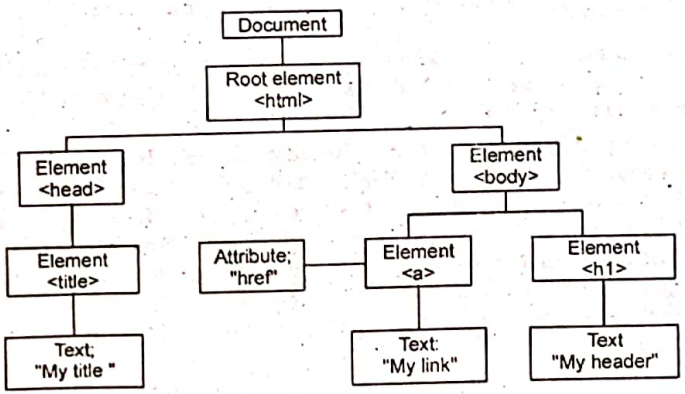
(4) Less code than Servlet: In JSP, we can use a lot of tags such as action tags, jstl, custom tags etc. that reduces the code. Moreover, we can use EL, implicit objects etc.

Q.6. Define the following terminology. (12.5)

Q.6. (a) Document tress

Ans. When a web page is loaded, the browser creates a Document Object Model of the page.

HTML DOM model is constructed as a tree of Objects:



With the object model, JavaScript gets all the power it needs to create dynamic HTML:

- JavaScript can change all the HTML elements in the page
- JavaScript can change all the HTML attributes in the page
- JavaScript can change all the CSS styles in the page
- JavaScript can remove existing HTML elements and attributes

Q.6. (b) CSS style sheets

Ans. CSS stands for Cascading Style Sheets.CSS describes how HTML elements are to be displayed on screen, paper, or in other media. It is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages , user interfaces for web applications, and user interfaces for many mobile applications.

CSS saves a lot of work. It can control the layout of multiple web pages all at once.CSS can be added to HTML elements in 3 ways:

- **Inline** - by using the style attribute in HTML elements
- **Internal** - by using a <style> element in the <head> section
- **External** - by using an external CSS file

The most common way to add CSS, is to keep the styles in separate CSS files.

Q.6. (c) DNS and URL

Ans. The **Domain Name System (DNS)** is a hierarchical decentralized naming system for computers, services, or any resource connected to the Internet or a private network. It associates various information with domain names assigned to each of the participating entities. Most prominently, it translates more readily memorized domain names to the numerical IP addresses needed for the purpose of locating and identifying computer services and devices with the underlying network protocols. By providing a worldwide, distributed directory service, the Domain Name System is an essential

it crash. If a vulnerability is found, a tool called a fuzz tester (or fuzzer), indicates potential causes. Fuzz testing was originally developed by Barton Miller at the University of Wisconsin in 1989.Fuzzers work best for problems that can cause a program to crash, such as buffer overflow, cross-site scripting, denial of service attacks, format bugs and SQL injection.

Q.8. (*b*) Latest Trends in Web Technologies

Ans. Internet of Things (IoT) will have a huge impact: The Internet of Things has taken app development to a new level. According to Technavio, IoT will grow by 31.72% (CAGR) between 2015 and 2019. By linking smart objects to the internet, IoT enables the exchange of data that was never possible before.

As more and more devices are being connected and accessible to the network, we'll find web developers coming up with upgraded solutions to help users control and communicate with their everyday gadgets and equipment.

Browser based IDEs: Odds are that you have your own favorite development environment. Maybe you fell in love with VIM years ago or you're an IntelliJ fanatic. That's going to change soon as more and more people are starting to use cloud-based versions of IDEs.

They're fast and they're accessible, and some of them have a huge community behind them. Flexibility is key here, and while you may not use these tools full time, it's definitely good to know that they're available if you want to do a quick test of a bootstrap code of Jade without having to download a single file.

Full-screen navigation design: Full-screen navigation design is a feature that improves the user experience on mobile devices. Let's say a user is navigating a website on his mobile phone and he comes across a registration form. As he taps on the registration form, the form jumps to a full-screen size enabling the user to fill out the form in a more natural way. More and more web developers and designers are developing sites for full-screen navigation designs and this trend is going to continue.

Q.8. (*c*) Web attacks and their prevention.

Ans. The most common web application threats include:

• **Cross site scripting (XSS):** XSS (Cross-Site Scripting) is regarded as the most common type of computer security vulnerability, with a huge number of web applications that are online today being vulnerable to this type of malicious script.

TOP PREVENTION TIP: An intelligent Web Application Firewall (WAF) can shield these vulnerabilities, working in conjunction with the behavioural firewall, blocking sophisticated and dangerous attacks.

• **SQL injection:** SQL Injections are one of the most serious type of attack on the internet. These attacks take advantage of web application vulnerabilities to gain control of databases and all of the information contained within them.

TOP PREVENTION TIP: In order to keep your databases secure you should practice regular auditing and remediation of your application to ensure that any vulnerability are discovered and dealt with as quickly as possible.

•**DDoS attacks:** DDoS stands for a denial-of-service or as it's more commonly known, a distributed denial-of-service (DDoS). This type of attack is an attempt to make a machine or network resource unavailable to its intended users.

TOP PREVENTION TIP: A reliable and well-reviewed DDoS protection tool is the best defence against DDos Attacks; there are plenty of tools to choose from.

FIRST TERM EXAMINATION [FEB. 2017]
SIXTH SEMESTER [B.TECH]
WEB ENGINEERING [ETCS–308]

Time : 1.5 Hrs. **M.M. : 30**

Note: Q. No. 1 is compulsory. Attempt any two more Questions from the rest.

Q.1. Attempt All:

Q.1. (a) What is CDATA section? **(1)**

Ans. CDATA stands for Character Data and it means that the data in between these strings includes data that *could* be interpreted as XML markup, but should not be.

The key differences between CDATA and comments are:

• As Richard points out, CDATA is still part of the document, while a comment is not.

• In CDATA you cannot include the string]]> (CDEnd), while in a comment -- is invalid.

• Parameter Entity references are not recognized inside of comments.

This means given these three snippets of XML from one well-formed document:

<!ENTITY MyParamEntity "Has been expanded">

Q.1. (b) Define eval() function. **(1)**

Ans. eval() is a function property of the global object. The argument of the **eval()** function is a string. If the string represents an expression, **eval()** evaluates the expression. If the argument represents one or more **JavaScript** statements, **eval ()** evaluates the statements.

Q.1. (c) Differentiate between SGML and HTML. **(1)**

Ans. Comparison between SGML and HTML:

	SGML	HTML
Full Form	It stands for the Standard Generalized Markup Language.	It stands for Hyper Text Markup Language.
Type	application/sgml, text/sgml	text/html
Type code	Text	Text
Uniform type	public.xml	public.html
Developed by	ISO	WWW Consotium
Format type	It is a mark up language.	It is a mark up language.

Q.1. (d) What are sessions and cookies? **(1)**

Ans. A session as you probably mean it is a server-side object which stores state. You use it in servlets to store and retrieve data. You keep hearing people saying HTTP is a stateless protocol, right? They mean when you load a page, you're finished as far as the web server is concerned. If you reload a page, the new request isn't associated in any way with the previous one.

A cookie is a small piece of information a browser sends to a server with every request.

Q.1. (e) Differentiate between indexOF() and last Index Of(). **(1)**

Ans. The **String** class provides two methods that allow you to search a string for a specified character or substring:

- **indexOf( )** Searches for the first occurrence of a character or substring.
- **lastIndexOf( )** Searches for the last occurrence of a character or substring.

Q.1. (f) Write the names of two application server and two web server. (1)

Ans. WEB SERVERS:

Apache web server – the HTTP web server

Apache Tomcat

APPLICATION SERVER:

BASIC

C

Q.1. (g) Briefly describe WML and XML. (1)

Ans. XML (Extensible Markup Language) is an increasingly popular format for sharing data that allows Web developers to create customized tags, as well as use predefined tags, used for developing a single Web site whose content can be formatted to display appropriately on various devices. Wireless devices use a subset of XML called WML. WML (wireless markup language) allows Web developers to design pages specifically for microbrowsers.

Q.1. (h) Differentiate between JSP and Servlet. (1)

Ans.

SERVLET	JSP
A servlet is a server-side program and written purely on Java.	JSP is an interface on top of Servlets. In another way, we can say that JSPs are extension of servlets to minimize the effort of developers to write User Interfaces using Java programming.
Servlets run faster than JSP	JSP runs slower because it has the transition phase for converting from JSP page to a Servlet file. Once it is converted to a Servlet then it will start the compilation

Q.1. (i) Differentiate between Inline, External and Internal Style Sheets. (1)

Ans. External Style Sheet

With an external style sheet, you can change the look of an entire website by changing just one file

Internal Style Sheet

An internal style sheet may be used if one single page has a unique style.

Inline Styles

An inline style may be used to apply a unique style for a single element.

Q.1. (j) .....are the block of JavaScript code that performs a specific task and return a value. (1)

Ans. Functions

Q.2. (a) Explain any two Email protocols. (5)

Ans. E-mail Protocols are set of rules that help the client to properly transmit the information to or from the mail server. Here in this tutorial, we will discuss various protocols such as **SMTP, POP,** and **IMAP.**

SMPTP

SMTP stands for **Simple Mail Transfer Protocol.** It was first proposed in 1982. It is a standard protocol used for sending e-mail efficiently and reliably over the internet.

Key Points:

- SMTP is application level protocol.
- SMTP is connection oriented protocol.

IMAP

IMAP stands for **Internet Mail Access Protocol.** It was first proposed in 1986. There exist versions of IMAP as follows:

1. Original IMAP
2. IMAP2

Key Points:

- IMAP allows the client program to manipulate the e-mail message on the server without downloading them on the local computer.
- The e-mail is hold and maintained by the remote server.

It enables us to take any action such as downloading, delete the mail without reading the mail.It enables us to create, manipulate and delete remote message folders called mail boxes.

Q.2. (b) Write a program to show the use of all tags and attributes used to create a table. (5)

Ans.

```
<table>
 <thead>
 <tr>
   <th>Month</th>
   <th>Savings</th>
 </tr>
 </thead>
 <tfoot>
 <tr>
    <td>Sum</td>
    <td>$180</td>
 </tr>
 </tfoot>
 <tbody>
 <tr>
   <td>January</td>
   <td>$100</td>
 </tr>
 <tr>
    <td>February</td>
    <td>$80</td>
 </tr>
```

```
</tbody>
</table>
```

Q.3. (a) What are the types of Pop-up boxes supported by JavaScript. Explain them with the help of a program. **(5)**

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

Alert Box

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

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

Syntax

window.alert("*sometext*");

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

Example

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

Confirm Box

A confirm box is often used if you want the user to verify or accept something.

When a confirm box pops up, the user will have to click either "OK" or "Cancel" to proceed.

If the user clicks "OK", the box returns true. If the user clicks "Cancel", the box returns false.

Syntax

window.confirm("*sometext*");

Q.3. (b) Explain the types of selectors used in CSS with example. **(5)**

Ans. CSS selectors are the part of **CSS rules** that determine what HTML elements that are affected by the CSS rule. Here is an example CSS rule:

```
div {
  border: 1px solid black;
}
```

The CSS selector part of the above CSS rule is this:

```
div
```

This selector means that all div elements should be targeted by the CSS rule.

There are several different types of CSS selectors. Both CSS 1.0, CSS 2.1 and CSS 3.0 added selectors to the CSS standard. The rest of this text will go through these CSS selectors.

Universal Selector

The universal CSS selector is used to select all elements. It is marked with a *. Here is a universal CSS selector example:

```
* {
  font-size: 18px;
}
```

This example selects all HTML elements and set their font-size CSS property.

The universal CSS selector is not so often used alone. It is more often used with a child selector or descendant selector.

Element Selector

The element selector is the most basic CSS selector. It selects all the HTML elements of the same type. For instance, all div elements or p elements.

With the element CSS selector you simply write the element name of the elements to apply the CSS rule to. Here are three examples:

```
div {
  border: 1px solid black;
}
p {
  font-size: 18px;
}
input {
  border: 1px solid #cccccc;
}
```

These three CSS rules each have a selector that selects all of a certain type of HTML elements. The first CSS selector selects all div elements. The second CSS selector selects all p elements. The third CSS selector selector selects all input elements.

You can select any HTML element using the element selector. All elements of that type / name will be affected by the CSS rule having the element selector.

Q.4. (a) Write a program to accept an Email address from the user in an HTML form and then validates it. **(6)**

Ans.

```
• <form>
• <input type= "email" placeholder= "Enter your email">
• <input type= "submit" value= "Submit">
• </form>
• <input
• type="text"
• pattern="/^[a-zA-Z0-9.!#$%&'*+/=?^_`{|}~-]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)*$/"
• required
• >
```

Q.4. (b) Explain the lifecycle and architecture of a Servlet. **(4)**

Ans. A servlet life cycle can be defined as the entire process from its creation till the destruction. The following are the paths followed by a servlet.

- The servlet is initialized by calling the **init()** method.
- The servlet calls **service()** method to process a client's request.
- The servlet is terminated by calling the **destroy()**method.
- Finally, servlet is garbage collected by the garbage collector of the JVM.

The init() Method

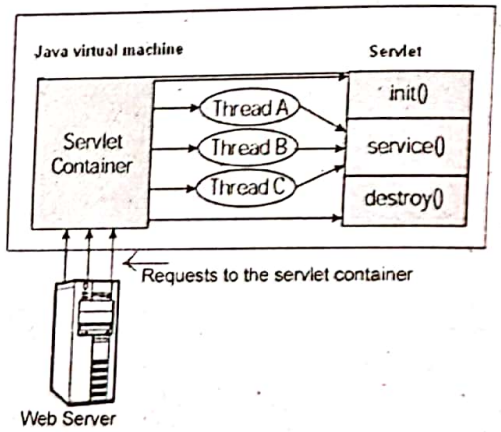
- The init method is called only once. It is called only when the servlet is created, and not called for any user requests afterwards. So, it is used for one-time initializations, just as with the init method of applets.

The service() Method

- The service() method is the main method to perform the actual task. The servlet container (i.e. web server) calls the service() method to handle requests coming from the client(browsers) and to write the formatted response back to the client.

The destroy() Method

• he destroy() method is called only once at the end of the life cycle of a servlet. This method gives your servlet a chance to close database connections, halt background threads, write cookie lists or hit counts to disk, and perform other such cleanup activities.



END TERM EXAMINATION [MAY-JUNE 2017]
SIXTH SEMESTER [B.TECH.]
WEB ENGINEERING [ETCS–308]

Time : 3 Hrs. **M.M. : 75**

Note: Attempt any five questions including Q. No. 1 which is compulsory.

Q.1. (a) State the difference between Internet and World Wide Web. **(2)**

Ans.

Comparison chart of internet and WWW

INTERNET	WWW
1. Internet originated sometimes in late 1960s.	English scientist Tim Berners-Lee invented the World Wide Web in 1989
2. Nature of internet is hardware.	Nature of www is software.
3. Internet consists of computers, routers, cables, bridges, servers, cellular towers, satellites, etc.	www consists of information link text, images, audio, video
4. The first version of the Internet was known as ARPANET	In the beginning WWW was known as NSFNET
5. Internet works on the basis of Internet Protocol (IP)	WWW works on the basis of Hyper Text Transfer Protocol (HTTP)

Q.1. (b) What is the difference between HTML and XHTML? **(2)**

Ans. Comparison chart

HTML versus XHTML comparison chart

	HTML	XHTML
Introduction	HTML or HyperText Markup Language is the main markup language for creating web pages and other information that can be displayed in a web browser.	XHTML (Extensible HyperText Markup Language) is a family of XML markup languages that mirror or extend versions of the widely used Hypertext Markup Language (HTML), the language in which web pages are written.
Filename extension	.html, .htm	.xhtml, .xht, .xml, .html, .htm
Internet media type	text/html	application/xhtml+xml
Developed by	W3C & WHATWG	World Wide Web Consortium
Type of format	Document file format	Markup language
Extended from	SGML	XML, HTML

Q.1. (c) What are the different components of CSS? **(3)**

Ans. Common sections:

- HEADER
- FOOTER
- BODY
- SIDEBAR

Common properties:

- FONTS
- COLOURS
- SIZING

Q.1. (d) What is NaN? **(1)**

Ans. In computing, **NaN**, standing for **not a number**, is a numeric data type value representing an undefined or unrepresentable value, especially in floating-point calculation. Two separate kinds of NaNs are provided, termed quiet NaNs and signaling NaNs. Quiet NaNs are used to propagate errors resulting from invalid operations or values, whereas signaling NaNs can support advanced features such as mixing numerical and symbolic computation or other extensions to basic floating-point arithmetic.

Q.1. (e) Name any three predefined JSP tags. **(3)**

Ans.

jsp:forward	forwards the request and response to another resource.
jsp:include	includes another resource.
jsp:useBean	creates or locates bean object.

Q.1. (f) Briefly explain lifecycle of a servlet? **(3)**

Ans. Servlets follow a three-phase life: *initialization*, *service*, and *destruction*, with initialization and destruction typically performed once, and service performed many times. Initialization is the first phase of the Servlet life cycle and represents the creation and initialization of resources the Servlet may need to service requests. The service phase of the Servlet life cycle represents all interactions with requests until the Servlet is destroyed. The destruction phase of the Servlet life cycle represents when a Servlet is being removed from use by a container.

Q.1. (g) What are Plugins? **(2)**

Ans. In computing, a **plug-in** (or **plugin, add-in, addin, add-on, addon,** or **extension**) is a software component that adds a specific feature to an existing computer program. When a program supports plugins, it enables customization. The common examples are the plug-ins used in web browsers to add new features such as search-engines, virus scanners, or the ability to use a new file type such as a new video format.

Q.1. (h) What are the technologies used in Web 2.0? **(3)**

Ans. The client-side (Web browser) technologies used in Web 2.0 development include Ajax and JavaScript frameworks. Ajax programming uses JavaScript and the Document Object Model to update selected regions of the page area without undergoing a full page reload. To allow users to continue to interact with the page,

communications such as data requests going to the server are separated from data coming back to the page (asynchronously). Otherwise, the user would have to routinely wait for the data to come back before they can do anything else on that page, just as a user has to wait for a page to complete the reload. This also increases overall performance of the site, as the sending of requests can complete quicker independent of blocking and queueing required to send data back to the client. The data fetched by an Ajax request is typically formatted in XML or JSON (JavaScript Object Notation) format, two widely used structured data formats.

Q.1. (i) What are widgets? **(2)**

Ans. Widgets are an essential aspect of home screen customization. You can imagine them as "at-a-glance" views of an app's most important data and functionality that is accessible right from the user's home screen. Users can move widgets across their home screen panels, and, if supported, resize them to tailor the amount of information within a widget to their preference.

Q.1. (j) What do you mean by parameter tampering? **(2)**

Ans. Parameter tampering is a form of Web-based attack in which certain parameters in the Uniform Resource Locator (URL) or Web page form field data entered by a user are changed without that user's authorization. This points the browser to a link, page or site other than the one the user intends (although it may look exactly the same to the casual observer).

Q.1. (k) What is fuzzer? **(2)**

Ans. Fuzz testing (fuzzing) is a quality assurance technique used to discover coding errors and security loopholes in software, operating systems or networks. It involves inputting massive amounts of random data, called fuzz, to the test subject in an attempt to make it crash. If a vulnerability is found, a software tool called a fuzzer can be used to identify potential causes.

Q.2. (a) What are the limitations of HTML? **(4)**

Ans. HTML is also known as HyperText Markup Language provides the creation of the web pages.

- The HTML pages are the documents that can be read by the server, and are not the best fit to be read by humans.

- HTML forms have the dependency on scripting languages and it results in complex document creation that consumes more time.

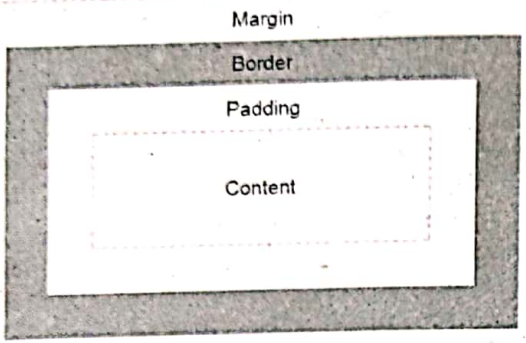
- HTML doesn't initialize the form data properly and doesn't make it easier for the users to enter the information once.

- HTML is having some limitations with the use of forms that doesn't allow encoding formats, url encoded or multipart forms.

Q.2. (b) Describe CSS Box model with an example. **(5)**

Ans. All HTML elements can be considered as boxes. In CSS, the term "box model" is used when talking about design and layout.

The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content. The image below illustrates the box model:



Explanation of the different parts:

- **Content** - The content of the box, where text and images appear
- **Padding** - Clears an area around the content. The padding is transparent
- **Border** - A border that goes around the padding and content
- **Margin** - Clears an area outside the border. The margin is transparent

The box model allows us to add a border around elements, and to define space between elements.

Example

```
div {
  width: 300px;
  border: 25px solid green;
  padding: 25px;
  margin: 25px;
}
```

Q.2. (c) Explain DHTML and its features. **(3.5)**

Ans. DHTML stands for Dynamic HTML. The first thing that we need to clear about DHTML is that it is neither a language like HTML, JavaScript etc. nor a web standard. It is just a combination of HTML, JavaScript and CSS. It just uses these languages features to build dynamic web pages. DHTML is a feature of Netscape Communicator 4.0, and Microsoft Internet Explorer 4.0 and 5.0 and is entirely a "client-side" technology.

Features of DHTML:

1. Simplest feature is making the page dynamic.
2. Can be used to create animations, games, applications, provide new ways of navigating through web sites.
3. DHTML use low-bandwidth effect which enhance web page functionality.
4. Dynamic building of web pages is simple as no plug-in is required.
5. Facilitates the usage of events, methods and properties and code reuse.

Q.3. (a) Discuss XML and XSLT with an example. **(7.5)**

Ans. XML stands for eXtensible Markup Language. XML was designed to store and transport data.

XML was designed to be both human- and machine-readable.

XML Example 1

```
<?xml version= "1.0" encoding= "UTF-8"?>
<note>
 <to>Tove</to>
 <from>Jani</from>
 <heading>Reminder</heading>
 <body>Don't forget me this weekend!</body>
</note>
```

XSL (eXtensible Stylesheet Language) is a styling language for XML.

XSLT stands for XSL Transformations.

This example will teach you how to use XSLT to transform XML documents into other formats (like transforming XML into HTML).

XSLT Example

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match= "/">
 <html>
 <body>
  <h2>My CD Collection</h2>
  <table border="1">
   <tr bgcolor="#9acd32">
    <th>Title</th>
    <th>Artist</th>
   </tr>
   <xsl:for-each select="catalog/cd">
    <tr>
     <td><xsl:value-of select= "title"/></td>
     <td><xsl:value-of select="artist"/></td>
    </tr>
   </xsl:for-each>
  </table>
 </body>
 </html>
</xsl:template>
</xsl:stylesheet>
```

Q.3. (b) What is DTD? **(2.5)**

Ans. A DTD is a Document Type Definition. A DTD defines the structure and the legal elements and attributes of an XML document.

With a DTD, independent groups of people can agree on a standard DTD for interchanging data.

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

If the DTD is declared inside the XML file, it must be wrapped inside the <!DOCTYPE> definition:

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

Q.3. (c) What is WML? **(2.5)**

Ans. WML (Wireless Markup Language), formerly called HDML (Handheld Devices Markup Languages), is a language that allows the text portions of Web pages to be presented on cellular telephones and personal digital assistants (PDAs) via wireless access. WML is part of the Wireless Application Protocol (WAP) that is being proposed by several vendors to standards bodies. The Wireless Application Protocol works on top of standard data link protocols, such as Global System for Mobile communication, code-division multiple access, and Time Division Multiple Access, and provides a complete set of network communication programs comparable to and supportive of the Internet set of protocols.

Q.4. (a) Create a student registration form in HTML and validate the name and email field using Javascript. **(7.5)**

Ans. StudentRegistration.html

```
<html>
<head>
<script type="text/javascript" src="validate.js"></script>
</head>
<body>
<form action="#" name="StudentRegistration" onsubmit= "return(validate());">
<table cellpadding= "2" width= "20%" bgcolor= "99FFFF" align="center"
cellspacing="2">
<tr>
<td colspan=2>
```



```
<center><font size=4><b>Student Registration Form</b></font></center>
</td>
</tr>
<tr>
<td>Name</td>
<td><input type=text name=textnames id="textname" size="30"></td>
</tr>
<tr>
<td>Father Name</td>
<td><input type="text" name="fathername" id="fathername"
size="30"></td>
</tr>
<tr>
<td>Postal Address</td>
<td><input type="text" name="paddress" id="paddress" size="30"></td>
</tr>
<tr>
<td>Personal Address</td>
<td><input type="text" name="personaladdress"
id="personaladdress" size="30"></td>
</tr>
<tr>
<td>Sex</td>
<td><input type="radio" name="sex" value="male" size="10">Male
<input type="radio" name="sex" value="Female" size="10">Female</td>
</tr>
<tr>
<td>EmailId</td>
<td><input type="text" name="emailid" id="emailid" size="30"></td>
</tr>
<tr>
<td>DOB</td>
<td><input type="text" name="dob" id="dob" size="30"></td>
</tr>
<tr>
<td>MobileNo</td>
<td><input type="text" name="mobileno" id="mobileno" size="30"></td>
</tr>
<tr>
<td><input type="reset"></td>
```

```
<td colspan="2"><input type="submit" value="Submit Form" /></td>
</tr>
</table>
</form>
</body>
</html>
```

Form Validation

```
function validate()
{
 if( document.StudentRegistration.textnames.value == "" )
 {
  alert( "Please provide your Name!" );
  document.StudentRegistration.textnames.focus() ;
  return false;
 }
 }
var email = document.StudentRegistration.emailid.value;
 atpos = email.indexOf("@");
 dotpos = email.lastIndexOf(".");
if (email == "" || atpos < 1 || ( dotpos - atpos < 2 ))
{
  alert("Please enter correct email ID")
  document.StudentRegistration.emailid.focus() ;
  return false;
}
```

Q.4. (b) Explain the concept of event handling in Java Script. **(5)**

Ans. An event occurs when something happens in a browser window. The kinds of events that might occur are due to:

- A document loading
- The user clicking a mouse button
- The browser screen changing size

When a function is assigned to an event handler, that function is run when that event occurs.

A handler that is assigned from a script used the syntax '[element].[event] = [function];', where [element] is a page element, [event] is the name of the selected event and [function] is the name of the function that occurs when the event takes place.

For example:

```
document.onclick = clickHandler;
```

This handler will cause the function clickHandler() to be executed whenever the user clicks the mouse anywhere on the screen. Note that when an event handler is assigned,

the function name does not end with parentheses. We are just pointing the event to the name of the function. The clickHandler() function is defined like this:

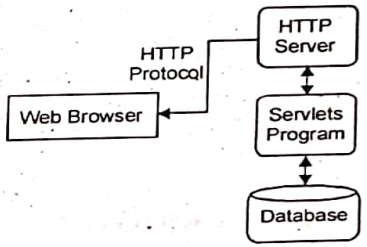
```
function clickHandler(event) {
 //some code here
}
```

By convention the event is represented by the variable 'event()'. In some browsers the event must be explicitly passed to the function, so as a precaution it's often best to include a conditional to test that the event() variable has been passed, and if it hasn't then to use an alternative method that works on those other browsers:

```
function clickHandler(event) {
 event = event || window.event;
 //some code here
}
```

Q.5. (a) Explain the architecture of Servlets? Differentiate between do post () and do Get(). **(3.5+4=7.5)**

Ans. The following diagram shows the position of Servlets in a Web Application.



Servlets Tasks

Servlets perform the following major tasks

- Read the explicit data sent by the clients (browsers). This includes an HTML form on a Web page or it could also come from an applet or a custom HTTP client program.
- Read the implicit HTTP request data sent by the clients (browsers). This includes cookies, media types and compression schemes the browser understands, and so forth.
- Process the data and generate the results. This process may require talking to a database, executing an RMI or CORBA call, invoking a Web service, or computing the response directly.

doGet() and doPost() are HTTP requests handled by servlet classes.

In doGet(), the parameters are appended to the URL and sent along with header information. This does not happen in case of doPost(). In doPost(), the parameters are sent separately. Since most of the web servers support only a limited amount of information to be attached to the headers, the size of this header should not exceed 1024 bytes. doPost() does not have this constraint. Usually programmers find it difficult to choose between doGet() and doPost().

doGet() shall be used when small amount of data and insensitive data like a query has to be sent as a request. doPost() shall be used when comparatively large amount of

sensitive data has to be sent. Examples are sending data after filling up a form or sending login id and password.

Q.5. (b) Write a program to print HELLO WORLD using JSP. **(5)**

Ans.

```
<html>
 <head>
  <title>Sample Application JSP Page</title>
 </head>
 <body bgcolor=white>
 <table border= "0" cellpadding= "10">
  <tr>
   <td align=center>
    <img src= "images/springsource.png">
   </td>
   <td>
    <h1>Sample Application JSP Page</h1>
   </td>
  </tr>
 </table>
 <br />
 <p>This is the output of a JSP page that is part of the HelloWorld application.<p>
 <%= new String("Hello!") %>
 </body></html>
```

Q.6. (a) What are the primary security controls? **(4.5)**

Ans. Security controls are safeguards or countermeasures to avoid, detect, counteract, or minimize security risks to physical property, information, computer systems, or other assets.

They can be classified by several criteria. For example, according to the time that they act, relative to a security incident:

- Before the event, **preventive controls** are intended to prevent an incident from occurring e.g. by locking out unauthorized intruders;
- During the event, **detective controls** are intended to identify and characterize an incident in progress e.g. by sounding the intruder alarm and alerting the security guards or police;
- After the event, **corrective controls** are intended to limit the extent of any damage caused by the incident e.g. by recovering the organization to normal working status as efficiently as possible.

According to their nature, for example:

- **Physical controls** *e.g.* fences, doors, locks and fire extinguishers;
- **Procedural controls** *e.g.* incident response processes, management oversight, security awareness and training;
- **Technical controls** *e.g.* user authentication (login) and logical access controls, antivirus software, firewalls;

- **Legal and regulatory or compliance controls** *e.g.* privacy laws, policies and clauses.

A similar categorization distinguishes control involving people, technology and operations/processes.

In the field of information security, such controls protect the **confidentiality, integrity** and/or **availability** of information - the so-called CIA Triad

Systems of controls can be referred to as frameworks or standards. Frameworks can enable an organization to manage security controls across different types of assets with consistency.

Q.6. (b) What are the different types of security threats and what are their possible solutions? **(4+4=8)**

Ans. 1. Viruses:

- A computer program developed intentionally to corrupt the files, applications, data, etc. of a computer. It gets back door entry (from storage devices, internet, USB etc.) without the knowledge of the user, and exploits the system mercilessly.

Prevention:

- 1. Beware of downloading applications, files (mp3, mp4, gif, etc) from the sites and also from the attachments of the e-mails.

2. Use/buy certified and secured products from the vendors.

2. Hackers:

- An intruder or probably an enemy of a particular entity with malicious intentions creates and injects malicious content to steal sensitive information or money or sometimes to destroy some part of data or applications.

Prevention:

- 1. Initiate strong encryption technology on the website.

2. Secure your websites with digital SSL certificates.

3. Avoid exposure to unauthenticated access, unnecessary access to employees or users.

3. Phishing Threats:

- Phishing means, when any website impersonates itself as a trustworthy and well established brand most probably to steal the information as well as money by misleading the online users. DNS farming attack, another type of phishing attack corrupts the DNS server because of which the client is automatically transferred to an imposter website (an illegal website having the look and feel of the original website).

Prevention:

1. Install updated version of antivirus tool.

2. Do not click blindly on the hyperlinks appearing in the e-mail that came from the unknown sources.

3. Secure your website with anti spam and phishing detection tools.

4. Always look for the "https:" before trusting the website especially, before providing credit card information and personal

4. Infected Websites:

- Be it in emails or ads on the websites or the normal looking website, you might never know what it is stored in it. These can be the sources of viruses, Trojans and malwares;by clicking on the link you install them in your system.

Prevention:

Avoid visiting to the suspicious websites specially those, which are not secured with digital certificates, install appropriate antivirus, anti malware, anti phishing tools.

5. Spywares, Adware, Trojans:

• Spywares, as the name suggests itself, are software that secretly tracks one's online behavior, and installs malicious software without user's concern. Adwares, Trojans also interpret the same behavior. Downloaded applications, corrupted CDs can be counted as their sources. They may display loads of disturbing ads and in turn slowing down one's internet, they can pass one's information to others.

Prevention:

1. Present your website with Symantec SSL certificates that come with Norton Secured Seal (available separately) that scans regularly your website against viruses, spyware, trojan horses, worms, adware or other malicious programs and will notify you by email.

2. Be very much careful before downloading any content from the suspicious or even from the unsecured website.

6. Insecure Wireless access points:

• Connecting to a wireless network like say connecting to a broadband router is not that safe. Devices like, laptop, PDA and mobile those connect with wireless connections are prone to get affected by several threats.

Prevention:

1. Keep your Service Set Identifier ID hidden.

2. Block the non-approved MAC addresses so as to avoid sniffing of the MAC addresses and spoof the address.

• **7. Social engineering:**

Tricks like pretexting; quid pro quo, tailgating etc. are accomplished in social engineering

• **Prevention:**

1. Implement strict measures against unauthorized access either from the user side or even from the employee side.

2. Educate your employees as well customers regarding various tricks and techniques of social engineering, and warn them to not providing any kind of personal information to irrelevant entity.

Q.7. (a) What is session management? What is session hijacking? How can it be prevented? (6.5)

Ans. Session Management is a mechanism used by the **Web container** to store session information for a particular user. There are four different techniques used by Servlet application for session management. They are as follows:

1. **Cookies**
2. **Hidden form field**
3. **URL Rewriting**
4. **HttpSession**

Session is used to store everything that we can get from the client from all the requests the client makes.

In computer science, **session hijacking**, sometimes also known as **cookie hijacking** is the exploitation of a valid computer session—sometimes also called a *session key*—to gain unauthorized access to information or services in a computer system. In particular, it is used to refer to the theft of a magic cookie used to authenticate a user to a remote server. It has particular relevance to web developers, as the HTTP cookies used to maintain a session on many web sites can be easily stolen by an attacker using an intermediary computer or with access to the saved cookies on the victim's computer (see HTTP cookie theft.

Methods to prevent session hijacking include:

• Encryption of the data traffic passed between the parties by using SSL/TLS; in particular the session key (though ideally all traffic for the entire session). This technique is widely relied-upon by web-based banks and other e-commerce services, because it completely prevents sniffing-style attacks. However, it could still be possible to perform some other kind of session hijack. In response, scientists from the Radboud University Nijmegen proposed in 2013 a way to prevent session hijacking by correlating the application session with the SSL/TLS credentials.

• Use of a long random number or string as the session key. This reduces the risk that an attacker could simply guess a valid session key through trial and error or brute force attacks.

• Regenerating the session id after a successful login. This prevents session fixation because the attacker does not know the session id of the user after s/he has logged in.

• Some services make secondary checks against the identity of the user. For instance, a web server could check with each request made that the IP address of the user matched the one last used during that session. This does not prevent attacks by somebody who shares the same IP address, however, and could be frustrating for users whose IP address is liable to change during a browsing session.

Q.7. (b) What are the features Wireless Application Firewalls? (6)

Ans. Developed in the early 1990s, WAFs were a new species of firewall initially created to respond to threats beyond the scope of traditional firewalls. These threats were dangerous because they utilized authorized protocols (such as HTTP), but attacked the application or underlying infrastructure over that protocol. WAFs are available in three rather broad categories: network-based, application-based and cloud-hosted.

Network-based WAFs are the traditional implementation of the technology. It offers several benefits and drawbacks. The largest benefit is that network-based WAFs are usually hardware-based and, being local, it reduces latency and negative performance impacts. The largest drawback is that this type of WAF product tends to be more expensive to both purchase and implement.

Application-based WAFs are generally installed closest to the application, such as on the hosting platform, and often times are fully integrated into the application code itself. The benefits of this type of WAF implementation are increased performance and customization options. As an example, since ModSecurity (an open source WAF) can be installed as a module in Apache, an application can take full advantage of the features while allowing the overhead to be handled by the server locally. The cost of deploying an application-based WAF is typically low as well, but the flexibility and scalability can leave something to be desired for larger organizations.

Cloud-hosted WAFs, meanwhile, offer a low-cost/low-effort application firewall implementation opportunity for organizations that want a turnkey product. These are easy to deploy, as they often require only a simple DNSchange to redirect application traffic, and are available on a subscription basis. While customization and performance limitations are usually drawbacks of cloud-based WAF products, they are often a viable stop-gap product that can be deployed rapidly.

Q.8. (a) Explain web syndication. **(6)**

Ans. Web syndication is a form of syndication in which content is made available from one website to other sites. Most commonly, websites are made available to provide either summaries or full renditions of a website's recently added content. The term may also describe other kinds of content licensing for reuse. Web syndication involves a website providing content to an arbitrary number of subscribing websites that redistribute it. For the subscribing sites, syndication is an effective way of adding greater depth and immediacy of information to their pages, making them more attractive to users. For the providing site, syndication increases exposure. This generates new traffic for the providing site—making syndication an easy and relatively cheap, or even free, form of advertisement. Commercial web syndication can be categorized in three ways:

- by **business models**
- by **types of content**
- by **methods for selecting distribution partners**

Commercial web syndication involves partnerships between content producers and distribution outlets. There are different structures of partnership agreements. One such structure is licensing content, in which distribution partners pay a fee to the content creators for the right to publish the content. Another structure is ad-supported content, in which publishers share revenues derived from advertising on syndicated content with that content's producer. A third structure is free, or barter syndication, in which no currency changes hands between publishers and content producers. This requires the content producers to generate revenue from another source, such as embedded advertising or subscriptions. Alternatively, they could distribute content without remuneration. Typically, those who create and distribute content free are promotional entities, vanity publishers, or government entities. Web syndication has been used to distribute product content such as feature descriptions, images, and specifications. As manufacturers are regarded as authorities and most sales are not achieved on manufacturer Web sites, manufacturers allow retailers or dealers to publish the information on their sites. Through syndication, manufacturers may pass relevant information to channel partners. Such web syndication has been shown to increase sales

Q.8. (b) Why Web 3.0 is referred as semantic web? How is web 3.0 different from web 2.0? **(2.5 + 4 = 6.5)**

Ans. The Semantic Web is an idea of World Wide Web inventor Tim Berners-Lee that the Web as a whole can be made more intelligent and perhaps even intuitive about how to serve a user's needs. Berners-Lee observes that although search engines index much of the Web's content, they have little ability to select the pages that a user really wants or needs. He foresees a number of ways in which developers and authors, singly or in collaborations, can use self-descriptions and other techniques so that context-understanding programs can selectively find what users want.

Web 2.0 uses the **read write web**, blogs, web applications, rich media, viral media, tagging or **Folksonomy** while sharing content and focusing on communities.

The *Web 3.0* standard uses **semantic web**, drag n drop mash ups, widgets, user behavior or **Me-onomy**, advertisement, user engagement, consolidates dynamic content and focuses on individuals.

Web 3.0 uses the **'*Data Web*' technology** featuring structures data records that are publishable and reusable on the web through query-able formats like RDF, XML and micro formats. It is the stepping-stone to complete semantic web, which enables new levels of application operability, data integration and makes data openly linkable and accessible in the form of web pages. The complete semantic web stage expands the scope of both structured and un-structured content through OWL and RDF semantic formats.

The *web 3.0* standard also describes the **latest trends for artificial intelligence**. Mass level use of technology is promptly visible here as in the case of an application that makes hit song predictions based on music websites of various colleges available on the net. Web 3.0 aims to highlight intelligence in an organic fashion through the interaction of people.

Can *web 3.0* extend itself to the **semantic web concept using artificial intelligence**? There is plenty on ongoing research to develop software, which uses reasoning based on intelligent agents and description logic. These applications perform all logical and reasoning operations using the set of rules, which expresses logical relationship between the data on the net and their concepts.Website development is a very booming and challenging field. If you are not grabbing latest technologies, you will out from information technologies market. As a popular website development company of India Hans Cyber technologies web designer experts always share there views for learner. Recently our experts analyzed that website designing strategy completely changed as web 3.0 launched which has various advance terms user compatible on the comparison of web 2.0. As per our experts analysis web 2.0 is read write base technology in which user or on line viewer can read, write and post their comments. But in web 3.0 website became more user friendly as it has various other features such as custom search, portability, dynamic content, widgets drag and drop etc.

As per our web designer experts research following difference between web 2.0 and web 3.0

WEB 2.0 features

- The Widely Read-Write Web
- Focused on Communities
- Blogs
- Sharing Content
- XML, RSS
- Web Applications
- Tagging (Folksonomy)
- Google
- Cost per click
- Rich Media, Viral

WEB 3.0 features

- The Portable Personal Web
- Focused on Individuals
- Lifestream
- Consolidating Dynamic Content
- Widgets, Drag & Drop Mashups
- The Semantic Web
- User Behavior
- Netvibes
- User Engagement
- Advertisement

FIRST TERM EXAMINATION [FEB. 2018]
SIXTH SEMESTER [B.TECH]
WEB ENGINEERING [ETCS–308]

Time : 1.5 hrs. **M.M. : 30**

Note: *Question 1 is compulsory. Attempt two questions from the remaining.*

Q. 1. (a) Differentiate between DIV and SPAN. **(2)**

Ans. (i) SPAN is a inline element whereas DIV is a block level element.

(ii) a SPAN allows you to separate things from the other elements around them on a page or within a document, it does not cause a line break. SPAN is perfect for in-line styling, like coloring a single word in a sentence to draw more attention to it. On the other hand, DIV, by default, creates a line break because it is used to make separate containers or boxes within a page or document.

Q. 1. (b) Explain the four possible keyword in a DTD declaration with suitable examples. **(2)**

Ans. There are four possible declaration keywords:

ELEMENT: Specifies the name of the tag that you use to build XML document. . General XML element declare by following way,

<!ELEMENT element_name (inside_element)>

element_name specifies the general identifier and inside_element specifies what are content inside the element.

ATTLIST: Specifies the attributes that you use inside element. General declaration

<!ATTLIST element_name attr_name attr_token_type attr_declaration>

element_name specifies the element name.

attr_name specifies the element attribute name.

attr_token_type specifies the structure/character string value.

attr_declaration specifies the default behavior of this attributes.

ENTITY: Specifies the pieces of text that are represent as a specific entity name.

XML ENTITY use to represent specific character that are difficult to write in standard keyboard.

General Declaration: <!ENTITY name definition>

Where name defines the entity name and definition defines specific code snippets that you want to actually represent.

NOTATION: Specifies the format type of non XML files like audio, image file. General declaration: <!NOTATION notation_name PUBLIC url>

<!NOTATION notation_name SYSTEM url>

where *notation_name* is name to identify declared notation. PUBLIC is keyword to specify the publisher URL. SYSTEM is keyword to specify URL for limited access.

Q. 1. (c) Differentiate between do get () and do post () method. **(2)**

Ans.

	doget()	dopost()
(i)	doGet is called when a HTTP GET request is made.	doPost is when a HTTP POST request is made.

(ii)	In doGet Method the parameters are appended to the URL and sent along with header information.	In doPost method, form data is sent in separate line in the body.
(iii)	Maximum size of data that can be sent is 240 bytes.	Data that can be sent is not limited.
(iv)	Parameters sent are not encrypted	Parameters are sent in encrypted form.
(v)	DoGet is faster if we set the response content length since the same connection is used. Thus increasing the performance	DoPost is slower compared to doGet since doPost does not write the content response length.

Q. 1. (d) Explain WML. (2)

Ans. WML stands for Wireless Markup Language and is intended for devices that implement the Wireless Application Protocol (WAP) specification, such as mobile phones. It provides navigational support, data input, hyperlinks, text and image presentation, and forms, much like HTML. It is an application of XML, which is defined in a document-type definition. WML takes care of the small screen and the low bandwidth of transmission.It is the markup language defined in the WAP(Wireless Application Protocol) specification. WAP sites are written in WML.WML is very similar to HTML. Both of them use tags and are written in plain text format. WML files have the extension ".wml". The MIME type of WML is "text/vnd.wap.wml". WML supports client-side scripting.

Q. 1. (e) Differentiate between servlets and CGI. (2)

Ans. Both Java servlets and CGI are used for creating dynamic web applications that accept a user request, process it on the server side and return responses to the user.

	Servlets	CGI(common gateway interface)
1.	Unlike CGI where a new process is started for each client request, a servlet processes each request as a thread inside of a process. Thus servlets improve the performance as it removes the overhead of creating a new process for request every time.	CGI is used to provide dynamic content to the user.
2.	In Servlets each request is handled by lightweight Java Thread	In CGI each request is handled by heavy weight OS process
3.	Servlets can link directly to the Web server	CGI cannot directly link to Web server.
4.	Servlets are platform independant.	CGI is platform dependent
5.	Servlets can resist attacks	CGI is vulnerable to attacks

(6)

Q. 2. (a) Write a Java Script program for login form validation.

(1) Text box not be blank put message in alert box on form.

(2) Reset will clear all the contents on click event.

(3) Max length should not be greater than 6 on form submit event.

Ans. Javascript program for login form validation


```
<script>
function loginform()
{
  var name = document.forms["RegForm"]["Name"];
  var email = document.forms["RegForm"]["EMail"];
  var phone = document.forms["RegForm"]["Telephone"];
  var password = document.forms["RegForm"]["Password"];
  if (name.value == "")
  {
    window.alert("Please enter your name.");
    name.focus();
    return false;
  }
  if (email.value == "")
  {
    window.alert("Please enter a valid e-mail address.");
    email.focus();
    return false;
  }
  if (email.value.indexOf("@", 0) < 0)
  {
    window.alert("Please enter a valid e-mail address.");
    email.focus();
    return false;
  }
  if (email.value.indexOf(".", 0) < 0)
  {
    window.alert("Please enter a valid e-mail address.");
    email.focus();
    return false;
  }
  if (phone.value == "")
  {
    window.alert("Please enter your telephone number.");
    phone.focus();
    return false;
  }
  if (password.value == "")
  {
    window.alert("Please enter your password");
    password.focus();
    return flase;
  }
  if (what.selectedIndex < 1)
  {
    alert("Please enter your course.");
    what.focus();
```

```
    return false;
  }
  return true;
}</script>
```

Q. 2. (b) Explain different types of style sheets by using list and frames. **(4)**

Ans. CSS is the language that defines the presentation of a web page. It is used to add color, background images, and textures, and to arrange elements on the page

Internal CSS Stylesheet

An internal stylesheet holds the CSS code for the webpage in the head section of the particular file. This makes it easy to apply styles like classes or id's in order to reuse the code. The downside of using an internal stylesheet is that changes to the internal stylesheet only effect the page the code is inserted into.

```
<head>
<style type= "text/css">
h1 {color:blue;}
h2 {color:red;}
p {color:green;}
</style>
</head>
```

External CSS Stylesheet

The External Stylesheet is a .css file that you link your website to. This makes it so that whatever you change in the .css sheet, will affect every page in your website. This prevents you from having to make many code changes in each page. This is for *"global"* site changes.

```
<head><br />
<link rel= "stylesheet" type="text/css" href="/support/style.css" /><br />
</head>
```

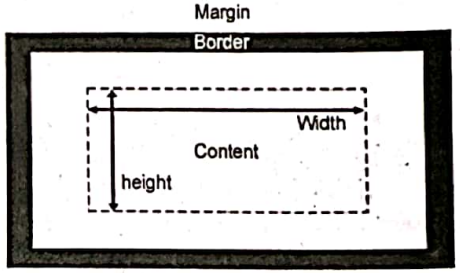
Inline CSS Styles

The Inline style is specific to the tag itself. The inline style uses the HTML *"style"* attribute to style a specific tag. This is not recommended, as every CSS change has to be made in every tag that has the inline style applied to it. The Inline style is good for one an individual CSS change that you do not use repeatedly through the site.

Q. 3. (a) Explain CSS box Model. **(4)**

Ans. The CSS box model is the foundation of layout on the Web — each element is represented as a rectangular box, with the box's content, padding, border, and margin built up around one another like the layers of an onion

Box properties



Width and height: The width and height properties set the width and height of the **content box**, which is the area in which the content of the box is displayed — this content includes both text content set inside the box, and other boxes representing nested child elements.

Padding: Padding refers to the *inner* margin of a CSS box — between the outer edge of the content box and the inner edge of the border. The size of this layer can be set on all four sides at once with the padding shorthand property, or one side at a time with the padding-top, padding-right, padding-bottom and padding-left properties.

Border: The **border** of a CSS box exists between the outer edge of the padding and the inner edge of the margin. By default the border has a size of 0 — making it invisible — but one can set the thickness, style and color of the border to make it appear. The border shorthand property allows to set all of these on all four sides at once, for example border: 1px solid black. This can be broken down into numerous different longhand properties for more specific styling needs:

- border-top, border-right, border-bottom, border-left: Set the thickness, style and color of one side of the border.
- border-width, border-style, border-color: Set only the thickness, style, or color individually, but for all four sides of the border.
- One can also set one of the three properties of a single side of the border individually, using border-top-width, border-top-style, border-top-color, etc.

Margins: The margin surrounds a CSS box, and pushes up against other CSS boxes in the layout. It behaves rather like padding; the shorthand property is margin and the individual properties are margin-top, margin-right, margin-bottom, and margin-left.

Q. 3. (b) Write a JSP menu driver program to perform Addition and Subtraction. **(6)**

Ans. JSP menu driven program to perform addition and subtraction

```
import java.util.*
public class menuExample
{
public static void main(String args[])
{
Scanner userInput = new Scanner(System.in);
int choice;
int value1, value2
double.number
do
{
System.out.println("*** Calculator v1.0***");
System.out.println("1, Addition");
System.out.println("2, Subtration");
System.out.println("3. Exit");
System.out.println("**********************");
System.out.println("Please enter your choice:");
choice = userInput.nextInt();
switch(choice)
```

```
{
case 1://addition
System.out.println("addition");
System.out.println("Please enter two values to be added");
value1 = userInput.nextInt();
value2 = userInput.nextInt();
System.out.println(value1 + " + " + value2 + " = " + (value1+value2);
break;
case 2://subtraction
System.out.println("subraction");
System.out.println("Please enter two values to be subtrated");
value1 = userInput.nextInt();
value2 = userInput.nextInt();
System.out.println(value1 + " – " + value2 + " = " + (value1 – value2);
break;
case 3://exit
System.out.println("You have chose exit!");
break;
default://default
System.out.println("You entered an invalid choice");
}
}while(choice != 3);
}//main
}//class
```

Q. 4. (a) Make the following table Using Table Tag **(5)**

S.No	Name of the student	III Year	
		Sem V	Sem Vi
1.	XYZ	b. DCN c. MP	D. WT E. OS

Ans.

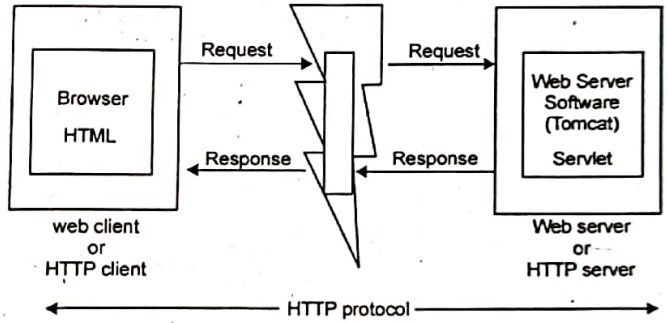
```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML Table Cellpadding</title>
  </head>
  <body>
    <table border = "1" cellpadding = "5" cellspacing = "5">
      <tr>
        <th>S. No.</th>
        <th>Name of student</th>
                <th>III Year</th>
      </tr>
      <tr>
        <td>1.</td>
        <td>XYZ</td>
      </tr>
    </table>
  </body>
</html>
```


Q. 4. (b) Explain the architecture of Servlets in detail. **(5)**

Ans. Java Servlets are programs that run on a Web or Application server and act as a middle layer between a requests coming from a Web browser or other HTTP client and databases or applications on the HTTP server.

Using Servlets, you can collect input from users through web page forms, present records from a database or another source, and create web pages dynamically.

A Servlet is a class, which implements the javax.servlet.Servlet interface. However instead of directly implementing the javax.servlet.Servlet interface we extend a class that has implemented the interface like javax.servlet.GenericServlet



1. In Web communication, the client and server uses Web (Internet) as an interface of communication. That is, both client and server should connect to Web to communicate, else, impossible to communicate. When connected, the simple client system can be called as "Web client" and the server can be called as "Web server".

2. The standard protocol used in the Web, now-a-days, is HTTP (HyperText Transfer Protocol) protocol. For this reason, the client and server are also can be called as "HTTP client" and "HTTP server" .

3. When connected, the client sends a request and server responses. In Web terminology, use only the words of request and response only. It is known as request/ response paradigm

4. To send a request, the software to be loaded on the Web cleint is "Browser". Similarly, the software required on the Web server is "Web server software". To get the Web server software, most commonly used server is Tomcat ,Weblogic and WebSphere (of IBM) etc.

5. Responsibilities of Web client

1.Should be able to take request from the client by displaying some GUI environment(like user name and passowrd as in Login screen).

2. To extract the data entered by the user and send it to the Web server as request.

3. Should be able to receive what the server responses and display to the user.

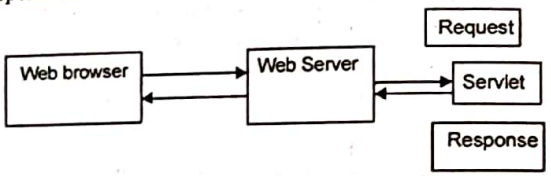
6. Now choose such a software on the client which can fulfill the above requirements. Obviously, it is browser. Browser represents the client system. When I say the client, it means I am talking about the browser on the client. Client sends request means, the browser on the client sends. Write a program which can take care of the above requirements in such a language understood by the browser. The easiet language programmer prefers is HTML.

7. Responsibilities of Web server

a. Should be able to receive the request send by the Web client.

b. As per the request, load an appropriate servlet, execute it and send the output of execution as response to client.

c. When response is delivered close the connection.



END TERM EXAMINATION [MAY-JUNE 2018]
SIXTH SEMESTER [B.TECH]
WEB ENGINEERING [ETCS–308]

Time : 3 hrs. **M.M. : 75**

Note: Attempt any five questions including Q. no. 1. which is cimpulsory.

Q. 1. Attempt all

Q. 1. (a) Describe basic HTML tags which are used in HTML head and HTML body. (Provide Example) **(5)**

Ans. The HTML <head> tag is used for indicating the head section of the HTML document. Tags included inside head tags are not displayed on browser window.

```
<!DOCTYPE html>
<html>
 <head>
  <title>HTML head Tag</title>
 </head>
 <body>
  actual content goes here
 </body>
</html>
```

- **The HTML <title> Element:** The <title> element defines the title of the document, and is required in all HTML/XHTML documents. The <title> element:
  - defines a title in the browser tab
  - provides a title for the page when it is added to favorites
  - displays a title for the page in search engine results

Example:

```
<!DOCTYPE html>
<html>
<head>
 <title>Page Title</title>
</head>
<body>
The content of the document.....
</body>
</html>
```

The HTML <style> Element: The <style> element is used to define style information for a single HTML page: Example:

```
<style>
 body {background-color: powderblue;}
 h1 {color: red;}
 p {color: blue;}
</style>
```

The HTML <link> Element: The <link> element is used to link to external style sheets:

```
<link rel = "stylesheet" href="mystyle.css">
```

Body Tags:

The HTML <body> tag is used for indicating the main content section of the HTML document.

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML body Tag</title>
  </head>
  <body>
    Body of the document...
  </body>
</html>
```

Heading Tags: Any document starts with a heading. You can use different sizes for your headings. HTML also has six levels of headings, which use the elements <h1>, <h2>, <h3>, <h4>, <h5>, and <h6>. While displaying any heading, browser adds one line before and one line after that heading. Example:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Heading Example</title>
  </head>
    <body>
    <h1>This is heading 1</h1>
    <h2>This is heading 2</h2>
    <h3>This is heading 3</h3>
    <h4>This is heading 4</h4>
    <h5>This is heading 5</h5>
    <h6>This is heading 6</h6>
  </body>
</html>
```

Paragraph Tag: The tag offers a way to structure your text into different paragraphs. Each paragraph of text should go in between an opening and a closing tag. Example:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Paragraph Example</title>
  </head>
  <body>
    <p>Here is a first paragraph of text.</p>
```



```
    <p>Here is a second paragraph of text.</p>
    <p>Here is a third paragraph of text.</p>
  </body>
</html>
```

Line Break Tag: Whenever you use the
 element, anything following it starts from the next line. This tag is an example of an empty element, where you do not need opening and closing tags, as there is nothing to go in between them. The
 tag has a space between the characters br and the forward slash.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Line Break  Example</title>
  </head>
  <body>
    <p>Hello<br />
      You delivered your assignment ontime.<br />
      Thanks<br />
      Mahnaz</p>
  </body>
</html>
```

Centering Content: You can use tag to put any content in the center of the page or any table cell. Example:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Centring Content Example</title>
  </head>
        <body>
    <p>This text is not in the center.</p>
      <center>
    <p>This text is in the center.</p>
  </center>
  </body>
</html>
```

Horizontal Lines: Horizontal lines are used to visually break-up sections of a document. The <hr> tag creates a line from the current position in the document to the right margin and breaks the line accordingly. Example:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Horizontal Line Example</title>
  </head>
        <body>
```

```
<p>This is paragraph one and should be on top</p>
<hr />
<p>This is paragraph two and should be at bottom</p>
</body>
</html>
```

Preserve Formatting: Sometimes, you want your text to follow the exact format of how it is written in the HTML document. In these cases, you can use the preformatted tag <pre>.

Any text between the opening <pre> tag and the closing </pre> tag will preserve the formatting of the source document. Example:

```
<!DOCTYPE html>
<html>
<head>
<title>Preserve Formatting Example</title>
</head>
<body>
<pre>
function testFunction( strText ){
alert (strText)
}
</pre>
</body>
</html>
```

Q. 1. (b) Explain the scope of global and local variables in JavaScript. (Provide Example) **(5)**

Ans. Java Script has two scopes – *global* and *local*.

Global variable: Any variable declared outside of a function belongs to the global scope, and is therefore accessible from anywhere in your code. Each function has its own scope, and any variable declared within that function is only accessible from that function and any nested functions.

```
function myFunction()
{
var carName = "Volvo";
}
```

Local variable: Because local scope in JavaScript is created by functions, it's also called function scope. When we put a function inside another function, then we create nested scope. For example:

```
var carName= "Volvo";
function myFunction()
{
}
```

Q. 1. (c) What is CSS box model? Explain with a proper diagram. **(5)**

Ans. Refer Q. 3. (a) of First Term 2018.

Q. 1. (d) What are the types of security threats possible at front and level (browser) and back end level (server side)? **(5)**

Ans. Browser side attacks or Client-Side Attacks

The Client-Side Attacks section focuses on the abuse or exploitation of a web site's users. When a user visits a web site, trust is established between the two parties both technologically and psychologically. A user expects web sites they visit to deliver valid content. A user also expects the web site not to attack them during their stay. By leveraging these trust relationship expectations, an attacker may employ several techniques to exploit the user.

1. Content Spoofing: Content Spoofing is an attack technique used to trick a user into believing that certain content appearing on a web site is legitimate and not from an external source. This attack exploits the trust relationship established between the user and the web site. The technique has been used to create fake web pages including login forms, defacements, false press releases, and so on.

2. Cross-Site Scripting: Cross-site Scripting (XSS) is an attack technique that forces a web site to echo attacker-supplied executable code, which loads in a user's browser. The code itself is usually written in HTML/JavaScript, but may also extend to VBScript, ActiveX, Java, Flash, or any other browser-supported technology. When an attacker gets a user's browser to execute his code, the code will run within the security context (or zone) of the hosting web site. With this level of privilege, the code has the ability to read, modify, and transmit any sensitive data accessible by the browser. A Cross-site Scripted user could have his account hijacked (cookie theft), his browser redirected to another location, or possibly shown fraudulent content delivered by the web site he is visiting. Cross-site Scripting attacks essentially compromise the trust relationship between a user and the web site. There are two types of Cross-site Scripting attacks: non-persistent and persistent.

Non-persistent attacks require a user to visit a specially crafted link laced with malicious code. Upon visiting the link, the code embedded in the URL will be echoed and executed within the user's web browser. Persistent attacks occur when the malicious code is submitted to a web site where it's stored for a period of time. Examples of an attacker's favorite targets often include message board posts, web mail messages, and web chat software. The unsuspecting user is not required to click on any link, just simply view the web page containing the code.

Server side security threats:

1. Brute Force Attack: In a brute force attack, the intruder attempts to gain access to a server by guessing a user password (usually the root administrator) through the SSH server, Mail server, or other service running on your system. The attacker will normally use software that will check every possible combination to find the one that works. Brute force detection software will alert you when multiple failed attempts to gain access are in progress and disable access from the offending IP address.

2. Open Relay: A Mail Transfer Agent (MTA) normally uses an SMTP server to send email from your server's users to people around the world. With an open relay, anyone can use your SMTP server, including spammers. Not only is it bad to give access to people who send spam, it could very well get your server placed on a DNS blacklist that some ISPs will use to block mail from your IP. It is very easy to close an open relay. Just follow the documentation for your MTA.

3. Botnet: Attackers use botnets to automatically run and distribute malicious software on "agent" servers. They then use the agent machines to attack or infect others. Because all of this can be done automatically without user intervention, botnets can

spread very quickly and be deadly for large networks. They are commonly used in DDoS attacks and spam campaigns.

4. **DoS :** DoS stands for Denial of Service, and is a technique attackers will use to effectively shut off access to your site. They accomplish this by increasing traffic on your site so much that the victim's server becomes unresponsive. While some DoS attacks come from single attackers, others are coordinated and are called Distributed Denial of Service (DDoS) attacks.

5. **SQL Injection:** Like XSS, SQL injection requires a vulnerability to be present in the database associated with a web application. The malicious code is inserted into strings that are later passed to the SQL server, parsed, and executed. As with other vulnerability-dependent attacks, you can prevent it by scanning for problem code and fixing it.

6. **Malware:** Malware can take many forms, but as the name implies, it is malicious software. It can take the form of viruses, bots, spyware, worms, trojans, rootkits, and any other software intended to cause harm. In most cases, malware is installed without the user's direct consent. It may attack the user's computer and/or attack other computers through the user's own system.

Q. 1. (e) Describe how Web 1.0, web 2.0 and Web 3.0 are different. (5)

Ans. Difference between Web 1.0, Web 2.0, Web 3.0:

Web 1.0

Web 1.0 was the first iteration and was called 'The Internet'. This iteration was used before 1999 and it is the "readable" phrase of the World Wide Web with flat data. In Web 1.0, there is only limited interaction between sites and web users. Web 1.0 is simply an information portal where users passively receive information without being given the opportunity to post reviews, comments, and feedback. The examples include static websites.

Web 2.0

It is the "writable" phrase of the World Wide Web with interactive data. Unlike Web 1.0, Web 2.0 facilitates interaction between web users and sites, so it allows users to interact more freely with each other. Web 2.0 encourages participation, collaboration, and information sharing. Examples of Web 2.0 applications are Youtube, Wiki, Flickr, Facebook, and so on.

Web 3.0

It is the "executable" phrase of Word Wide Web with dynamic applications, interactive services, and "machine-to-machine" interaction. Web 3.0 is a semantic web which refers to the future. In Web 3.0, computers can interpret information like humans and intelligently generate and distribute useful content tailored to the needs of users. One example of Web 3.0 is Tivo, a digital video recorder. Its recording program can search the web and read what it finds to you based on your preferences.

Q. 2. (a) How can we perform internal page linking and external page linking? Explain with examples for both the cases. (6)

Ans. Internal Links : Internal Links are hyperlinks that point at (target) the same domain as the domain that the link exists on (source). Internal links are links that go from one page on a domain to a different page on the same domain. They are commonly used in main navigation. These type of links are useful for three reasons:

- They allow users to navigate a website.
- They help establish information hierarchy for the given website.
- They help spread link equity (ranking power) around websites.

HTML internal link name is followed by head sign(#). HTML <a> tag is use for anchor point name, which is referred to a internal link into a same page.

Example:

```
<html>
<head>
</head>
<body>
 <a href= "#Lession.1">Lession.1</a> <br />
 <a href="#Lession.2">Lession.2</a> <br />
 <br />
 <a name="Lession.1">Introduction of Lession.1</a>
 <p>This is sub topic.1</p>
 <p>This is sub topic.2</p>
  <br />
 <br />
 <a name="Lession.2">Introduction of Lession.2</a>
 <p>This is sub topic.1</p>
 <p>This is sub topic.2</p>
 <br />
 <br />
 </body>
</html>
```

External Links : External links which link your pages to other web sites. This link is may be absolute path or relative link path. <a> tag is used for anchor name which is referred link to another web page. External link is great future to drive a webpage one to another and useful for surf many webpage in website. For Example:

```
<html>
<head>
</head>
<body>
 <a href="html.php">HTML</a>
 <br />
 <br />
 <a href="css.php">CSS</a>
 <br />
 <br />
 <a href="javascript.php">Java Script</a>
 <br />
</body>
</html>
```

Q. 2. (b) Write HTML code and use internal styling to create basic table structure as below: (6.5)

Monday		Wednesday	
9-11(AM)	2-4(PM)	9-11(AM)	2-4(PM)
C	C++	Java	Python

Ans. HTML code for table:

Monday		Wednesday	
9-11(AM)	2-4(PM)	9-11(AM)	2-4(PM)
C	C++	Java	Python

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML Tables</title>
  </head>
  <body>
    <table border = "1">
      <tr>
        <td>Monday</td>
        <td></td>
              <td>Wednesday</td>
        <td></td>
      </tr>
        <tr>
        <td>9-11(AM)</td>
        <td>2-4(PM)</td>
        <td>9-11(AM)</td>
        <td>2-4(PM)</td>
      </tr>
      <tr>
        <td>C</td>
        <td>C++</td>
        <td>Java</td>
        <td>Python</td>
    </table>
    </body>
</html>
```

Q. 3. Write code for document named as filename, html and filename, js script and perform following steps: . (12.5)

Create a form element in filename, html. Include below control elements inside the form:

(i) One control element for accepting the-full name.

(ii) One control element for accepting the phone number.

(iii) One control element which accepts the date as the date of birth.

(iv) One combo box (dropdown) for accepting multiple hobbies from the list

(v) One button to submit the form.

Perform following validation on pressing submit option in filename, js:

(vi) Name control should not be empty,

(vii) Phone number control should contain only numbers of length 10 digits.

Ans.

```
<html>
<head>
<title>Student Registration Form</title>
</head>
 <body>
<td>FIRST NAME</td>
<td><input type= "text" name="First_Name" maxlength="30"/>
(max 30 characters a-z and A-Z)
</td>
</tr>
 <tr>
<td>DATE OF BIRTH</td>
 <td>
<select name="Birthday_day" id="Birthday_Day">
<option value="-1">Day:</option>
<option value="1">1</option>
<option value="2">2</option>
<option value="3">3</option>
<option value="4">4</option>
<option value="5">5</option>
<option value="6">6</option>
<option value="7">7</option>
<option value="8">8</option>
<option value="9">9</option>
<option value="10" >10</option>
<option value="11">11</option>
<option value="12">12</option>
<option value="13">13</option>
<option value="14">14</option>
<option value="15">15</option>
<option value="16">16</option>
<option value="17">17</option>
<option value="18">18</option>
<option value="19">19</option>
<option value="20">20</option>
<option value="21">21</option>
 <option value="22">22</option>
<option value="23">23</option>
<option value="24">24</option>
<option value="25">25</option>
```

```
<option value="26">26</option>
<option value="27">27</option>
<option value="28">28</option>
<option value="29">29</option>
<option value="30">30</option>
<option value="31">31</option>
</select>
<select id="Birthday_Month" name="Birthday_Month">
<option value="-1">Month:</option>
<option value="January">Jan</option>
<option value="February">Feb</option>
<option value="March">Mar</option>
<option value="April">Apr</option>
<option value="May">May</option>
<option value="June">Jun</option>
<option value="July">Jul</option>
<option value="August">Aug</option>
<option value="September">Sep</option>
<option value="October">Oct</option>
<option value="November">Nov</option>
<option value="December">Dec</option>
</select>
<select name="Birthday_Year" id="Birthday_Year">
<option value="-1">Year:</option>
<option value="2012">2012</option>
<option value="2011">2011</option>
<option value="2010">2010</option>
<option value="2009">2009</option>
<option value="2008">2008</option>
<option value="2007">2007</option>
<option value="2006">2006</option>
<option value="2005">2005</option>
<option value="2004">2004</option>
<option value="2003">2003</option>
<option value="2002">2002</option>
<option value="2001">2001</option>
<option value="2000">2000</option>
<option value="1999">1999</option>
<option value="1998">1998</option>
<option value="1997">1997</option>
<option value="1996">1996</option>
<option value="1995">1995</option>
```



```
<option value="1994">1994</option>
<option value="1993">1993</option>
<option value="1992">1992</option>
<option value="1991">1991</option>
<option value="1990">1990</option>
<option value="1989">1989</option>
<option value="1988">1988</option>
<option value="1987">1987</option>
<option value="1986">1986</option>
<option value="1985">1985</option>
<option value="1984">1984</option>
<option value="1983">1983</option>
<option value="1982">1982</option>
<option value="1981">1981</option>
<option value="1980">1980</option>
</select>
</td>
</tr>
<tr>
<td>EMAIL ID</td>
<td><input type="text" name="Email_Id" maxlength="100" /></td>
</tr>
<tr>
<td>MOBILE NUMBER</td>
<td>
<input type="text" name="Mobile_Number" maxlength="10" />
(10 digit number)
</td>
</tr>
<tr>
<td>GENDER</td>
<td>
Male <input type="radio" name="Gender" value="Male" />
Female <input type="radio" name="Gender" value="Female" />
</td>
</tr>
<tr>
<td>HOBBIES <br /><br /><br /></td>
<td>
Drawing
<input type="checkbox" name= "Hobby_Drawing" value="Drawing" />
Singing
```

```
<input type="checkbox" name="Hobby_Singing" value="Singing" />
Dancing
<input type="checkbox" name="Hobby_Dancing" value="Dancing" />
Sketching
<input type="checkbox" name="Hobby_Cooking" value="Cooking" />
<br />
Others
<input type="checkbox" name="Hobby_Other" value="Other">
<input type="text" name="Other_Hobby" maxlength="30" />
</td>
</tr>
<tr>
<td colspan="2" align="center">
<input type= "submit" value="Submit">
<input type="reset" value="Reset">
</td>
</tr>
</table>
</form>
</body>
</html>
```

Q. 4. (a) Differentiate between DOS and DDOS attacks. **(4)**

Ans. Following is the difference between DOS and DDOS attacks:

DOS	DDOS
It is known as Denial of Service (DoS) attack	It is known as Distributed Denial of Service
The DoS attack is a cyber-attack in which the perpetrator seeks to make a machine or network resource unavailable to its intended	DDoS attack is a cyber-attack in which the incoming traffic flooding the victim originates from many different sources.
The DoS attack typically uses one computer and one Internet connection to flood a targeted system or resource.	The DDoS attack uses multiple computers and Internet connections to flood the targeted resource.
There is no malware involvement.	Use of malware to affect multiple machines via botnets
DOS attacks are comparatively less complicated	DDOS attacks are difficult to prevent and are more complicated
The DoS attack is a cyber-attack in which the perpetrator seeks to make a machine or network resource unavailable to its intended users by temporarily or indefinitely disrupting services of a host connected to the internet.	

Q. 4. (b) Differentiate between HTTP and HTTPS. **(4.5)**

Ans. Difference between HTTP and HTTPS:

HTTP	HTTPS
Stands for hyper text transfer protocol	Stands for Hyper text transfer protocol secure
In HTTP, URL begins with "http://"	URL starts with "https://"
HTTP is not secure	HTTPS is a secure protocol which uses TLS/SSL certificate to ensure the authentication.
HTTP Works at Application Layer	HTTPS works at Transport Layer
HTTP does not require any certificates	HTTPS needs SSL Certificates
HTTP uses port number 80 for communication	HTTPS uses 443 for communication

Q. 4. (c) Explain mechanisms available in HTTP to maintain the state of the session. **(4)**

Ans. Session tracking methods:

User Authorization: Users can be authorized to use the web application in different ways. Basic concept is that the user will provide username and password to login to the application. Based on that the user can be identified and the session can be maintained.

Hidden Fields : <INPUT TYPE="hidden" NAME="technology" VALUE="servlet">

Hidden fields like the above can be inserted in the webpages and information can be sent to the server for session tracking. These fields are not visible directly to the user, but can be viewed using view source option from the browsers. This type doesn't need any special configuration from the browser of server and by default available to use for session tracking.

URL Rewriting:

Original URL: http://server:port/servlet/ServletNam

Rewritten URL: http://server:port/servlet/ServletName?sessionid=7456

When a request is made, additional parameter is appended with the url. In general added additional parameter will be sessionid or sometimes the userid. It will suffice to track the session. This type of session tracking doesn't need any special support from the browser.

Cookies: Cookies are the mostly used technology for session tracking. Cookie is a key value pair of information, sent by the server to the browser. This should be saved by the browser in its space in the client computer. Whenever the browser sends a request to that server it sends the cookie along with it. Then the server can identify the client using the cookie.

In java, following is the source code snippet to create a cookie:

```
Cookie cookie = new Cookie("userID", "7456");
res.addCookie(cookie);
```

Session tracking is easy to implement and maintain using the cookies.

Session tracking API: Session tracking API is built on top of the first four methods. This is in order to help the developer to minimize the overhead of session tracking. This type of session tracking is provided by the underlying technology. Then, the servlet container manages the session tracking task and the user need not do it explicitly using the java servlets. Then, the servlet container manages the session tracking task and the user need not do it explicitly using the java servlets.

Q. 5. (a) Explain uses of the handler in making an Ajax call? (Provide Example) **(4.5)**

Ans. The biggest challenge in handling Ajax call is knowing the loading time for the web page. These methods register handlers to be called when certain events, such as initialization or completion, take place for any Ajax request on the page. The global events are fired on each Ajax request if the global property in jQuery.ajaxSetup() is true, which it is by default.

• ajaxComplete()

Whenever an Ajax request completes, jQuery triggers the ajaxComplete event. Any and all handlers that have been registered with the .ajaxComplete() method are executed at this time.

set up a basic Ajax load request:

```
<div class="trigger">Trigger</div>
<div class="result"></div>
<div class="log"></div>
```

Attach the event handler to the document:

```
$( document ).ajaxComplete(function() {
  $( ".log" ).text( "Triggered ajaxComplete handler." );
});
```

Now, make an Ajax request using any jQuery method:

```
$( ".trigger" ).click(function() {
$( ".result" ).load( "ajax/test.html" );
});
```

When the user clicks the element with class trigger and the Ajax request completes, the log message is displayed.

• ajaxSend()

Whenever an Ajax request is about to be sent, jQuery triggers the ajaxSend event. Any and all handlers that have been registered with the .ajaxSend() method are executed at this time.

set up a basic Ajax load request:

```
<div class="trigger">Trigger</div>
<div class="result"></div>
<div class= "log"></div>
```

Attach the event handler to the document:

```
$( document ).ajaxSend(function() {
$( ".log" ).text( "Triggered ajaxSend handler." );
});
```

Now, make an Ajax request using any jQuery method:

```
$( ".trigger" ).click(function() {
  $( ".result" ).load( "ajax/test.html" );
});
```

When the user clicks the element with class trigger and the Ajax request is about to begin, the log message is displayed.

• ajaxError()

Whenever an Ajax request completes with an error, jQuery triggers the ajaxError event. Any and all handlers that have been registered with the .ajaxError() method are executed at this time. To observe this method in action, set up a basic Ajax load request.


```
<button class="trigger">Trigger</button>
<div class="result"></div>
<div class="log"></div>
```

Attach the event handler to the document:

```
$( document ).ajaxError(function() {
$( ".log" ).text( "Triggered ajaxError handler." );
});
```

Now, make an Ajax request using any jQuery method:

```
$( "button.trigger" ).on( "click", function() {
$( "div.result" ).load( "ajax/missing.html" );
});
```

When the user clicks the button and the Ajax request fails, because the requested file is missing, the log message is displayed.

• jaxStart()

Whenever an Ajax request is about to be sent, jQuery checks whether there are any other outstanding Ajax requests. If none are in progress, jQuery triggers the ajaxStart event. Any and all handlers that have been registered with the .ajaxStart() method are executed at this time.

To observe this method in action, set up a basic Ajax load request:

```
<div class= "trigger">Trigger</div>
<div class= "result"></div>
<div class= "log"></div>
```

Attach the event handler to any element:

```
$( document ).ajaxStart(function() {
$( ".log" ).text( "Triggered ajaxStart handler." );
});
```

Now, make an Ajax request using any jQuery method:

```
$( ".trigger" ).click(function() {
$( ".result" ).load( "ajax/test.html" );
});
```

When the user clicks the element with class trigger and the Ajax request is sent, the log message is displayed.

• ajaxStop()

Whenever an Ajax request completes, jQuery checks whether there are any other outstanding Ajax requests. If none remain, jQuery triggers the ajaxStop event. Any and all handlers that have been registered with the .ajaxStop() method are executed at this time. The ajaxStop event is also triggered if the last outstanding Ajax request is cancelled by returning false within the beforeSend callback function.

To observe this method in action, set up a basic Ajax load request:

```
<div class="trigger">Trigger</div>
<div class="result"></div>
<div class="log"></div>
```

Attach the event handler to the document:

```
$( document ).ajaxStop(function() {
$( ".log" ).text( "Triggered ajaxStop handler." );
});
```

Now, make an Ajax request using any jQuery method:

```
$( ".trigger" ).click(function() {
```

```
$( ".result" ).load( "ajax/test.html" );
});
```

When the user clicks the element with class trigger and the Ajax request completes, the log message is displayed.

Q. 5. (b) What is intelligent web? (Provide Example) **(4)**

Ans. An intelligent Web is one capable of MAKING SENSE in an equivalent way to how humans do. It is not the iteration after the Semantic Web. There would need to follow the Sentiment Web, the Sentient-Sensory Web, the Synergistic Web The intelligent web, also often referred to as Web 3.0, involves the idea that World Wide Web pages, sites and applications will continue to be imbued with artificial intelligence. This contrasts Web 3.0 from Web 2.0 – today's system of highly networked but not very artificially intelligent web apparatus. Examples: Web mining, Knowledge mining,

Q. 5. (c) What is a widget and what are the benefits of using it in web applications? **(4)**

Ans. A widget is a specialized application that provides users with a quick view of specific information from the parent application. In addition, the widget can allow the user to access certain features without launching the parent application. Web widgets are implemented using the Web programming technologies, such as HTML, CSS, and JavaScript; but only a subset of Tizen Web APIs.

Applications:

1. Web widgets are pieces of code that you can embed right on to your Web page, or personal publishing space such as Blogger or WordPress.

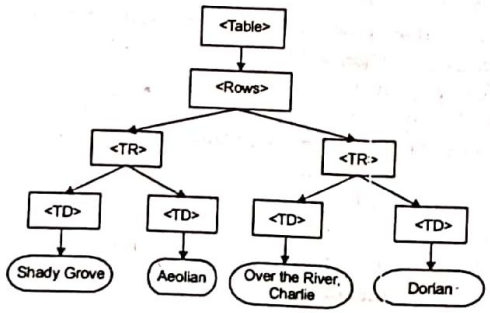
2. Web widgets work like a mini-application that you use to provide information to visitors on websites. They include things like search widgets, eBay trackers, news headlines, Twitter feeds, Facebook friend lists, games, clocks and other miniature "live" apps.

3. Web widgets are easy to use and require you only to copy and paste a snippet of code to display the widget, which is hosted on the developer's server. Widget directories, such as Widgetbox enable you to search for a specific type of widget, customize it for your own use, then copy and then paste the code to your own pages.

Q. 6. Explain following keywords with examples:

(a) Document object model **(4)**

Ans. The Document Object Model (DOM) is a programming API for HTML and XML documents. It defines the logical structure of documents and the way a document is accessed and manipulated. In the DOM specification, the term "document" is used in the broad sense - increasingly, XML is being used as a way of representing many different kinds of information that may be stored in diverse systems. With the Document Object Model, programmers can create and build documents, navigate their structure, and add, modify, or delete elements and content. The Document Object Model represents this table like this.



In the Document Object Model, documents have a logical structure which is very much like a tree. The Document Object Model does not specify that documents be *implemented* as a tree or a grove, nor does it specify how the relationships among objects be implemented in any way. One important property of DOM structure models is *structural isomorphism*: if any two Document Object Model implementations are used to create a representation of the same document, they will create the same structure model, with precisely the same objects and relationships. The name "Document Object Model" was chosen because it is an "object model" is used in the traditional object oriented design sense: documents are modeled using objects, and the model encompasses not only the structure of a document, but also the behavior of a document and the objects of which it is composed. The Document Object Model identifies:

- the interfaces and objects used to represent and manipulate a document
- the semantics of these interfaces and objects - including both behavior and attributes
- the relationships and collaborations among these interfaces and objects

Q. 6. (b) Block elements and inline elements **(4)**

Ans. HTML is made up of various elements that act as the building blocks of web pages. For the purpose of styling, elements are divided into two categories: *block-level* elements and *inline* elements.

An inline element does not cause a line break (start on a new line) and does not take up the full width of a page, only the space bounded by its opening and closing tag. It is usually used within other HTML elements. The "inline" category roughly corresponds to the category of phrasing content. Examples of inline elements are: anchor <a> tag, emphasis tag, image <img> tag

A block-level element always starts on a new line and takes up the full width of a page, from left to right. A block-level element can take up one line or multiple lines and has a line break before and after the element. Generally, block-level elements may contain inline elements and (sometimes) other block-level elements. Inherent in this structural distinction is the idea that block elements create "larger" structures than inline elements. Examples of the block-level tag are: Heading tags <h1> to <h6>, List (Ordered, Unordered, Description and List Item) tags <ol> , <ul> ,<dl> , <li>, Pre-formatted text tag <pre>, Blockquote tag <blockquote>

Q. 6. (c) Anonymous functions **(4.5)**

Ans. An anonymous function is a function that was declared without any named identifier to refer to it. As such, an anonymous function is usually not accessible after its initial creation.

```
var anon = function() {
 alert('I am anonymous');
}
anon();
```

The use of anonymous functions is as arguments to other functions. Another common use is as a closure. They are used as an **argument to other functions:**

```
setTimeout(function() {
 alert('hello');
}, 1000);
```

The anonymous function is passed to setTimeout, which will execute the function in 1000 milliseconds. They can be used as closure also..

```
(function() {
 alert('foo');
})();
```

Q. 7. (a) Describe steps involved in online purchasing using SET Protocol. **(6.5)**

Ans. Steps involved in online Purchasing using SET Protocol

1. Buyer indicates to merchant that she is interested in making a credit card purchase.

2. The merchant's system sends the customer an invoice and a unique transaction identifier.

3. The merchant's system sends the customer the merchant's certificate which includes the merchant's public key. The merchant's system also sends the certificate of its bank, which includes the bank's public key. Both of these certificates are encrypted with the private key of the certifying authority.

4. The customer uses the certifying authority's public key to decrypt the two certificates. The customer now has the merchant's public key and the bank's public key.

5. The customer generates two packages of information: the *order information* (**OI**) package and the *purchase instructions* (**PI**) package. The **OI**, destined for the merchant, contains the transaction identifier, brand of card being used; it does not include the customer's card number. The **PI**, destined for the merchant's bank, contains the transaction identifier, the card number, purchase amount agreed to the buyer, and a description of the order. The **OI** is encrypted with the merchant's public key; the **PI** is encrypted with bank's public key. The customer sends the **OI** and the **PI** to the merchant.

6. The merchant generates an authorization request for the card payment request, which includes the transaction identifier.

7. The merchant sends to its bank a message encrypted with the bank's public key. This message includes the authorization request, the **PI** package sent from the buyer, and the merchant's certificate.

8. The merchant's bank receives the message and unravels it. The bank checks for tampering. It also makes sure that the transaction identifier in the authorization request matches the one in the customer's **PI** package.

9. The merchant's bank then sends a request for payment authorization to the customer's credit card bank through traditional bankcard channels — just as the merchant's bank would request authorization for any normal credit card transaction.

10. Once the customer's bank authorizes the payment, the merchant's bank sends a response to the merchant, which is (of course) encrypted. The response includes the transaction identifier.

If the transaction was approved, the merchant sends its own response message to the customer. This message informs the customer that the payment was accepted and that the goods will be delivered. The customer will have software to handle all of its SET tasks.

Q. 7. (b) What are the application vulnerabilities and their defence? **(6)**

Ans. Application vulnerability is a system flaw or weakness in an application that could be exploited to compromise the security of the application. Once an attacker has found a flaw, or application vulnerability, and determined how to access it, the attacker has the potential to exploit the application vulnerability to facilitate a cyber crime. These

crimes target the confidentiality, integrity, or availability (known as the "CIA triad") of resources possessed by an application, its creators, and its users. Attackers typically rely on specific tools or methods to perform application vulnerability discovery and compromise.

SQL injection: SQL injection is a basic attack used to either gain unauthorized access to a database or to retrieve information directly from the database. The principles behind a SQL injection are simple and these types of attacks are easy to execute and master. SQL injection is a code injection technique that exploits a security vulnerability occurring in the database layer of an application. The vulnerability is present when user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed.

Detection Mechanisms for Injection Attacks:

Compare user input with possible known attack patterns

Reconstruct all user supplied input into a known safe input

Mechanisms that learn normal database accesses

Defensive programming

Fault injection/Fuzzying testing

Input Validation

Dynamic Checks and Static Source Code Analysis

Filter variables before constructing queries to be executed by server or database

Validate outgoing data

Restrict the access level of the web application to the minimum required

Cross-Site Scripting (XSS): Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted web sites. XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user. Flaws that allow these attacks to succeed are quite widespread and occur anywhere a web application uses input from a user within the output it generates without validating or encoding it. An attacker can use XSS to send a malicious script to an unsuspecting user. The end user's browser has no way to know that the script should not be trusted, and will execute the script. Because it thinks the script came from a trusted source, the malicious script can access any cookies, session tokens, or other sensitive information retained by the browser and used with that site. These scripts can even rewrite the content of the HTML page

Cross-Site Request Forgery (XSRF): CSRF is an attack which forces an end user to execute unwanted actions on a web application in which he/she is currently authenticated. With a little help of social engineering (like sending a link via email/chat), an attacker may trick the users of a web application into executing actions of the attacker's choosing. A successful CSRF exploit can compromise end user data and operation in case of normal user. If the targeted end user is the administrator account,this can compromise the entire web application. Cross-Site Request Forgery (CSRF) is an attack that tricks the victim into loading a page that contains a malicious request. It is malicious in the sense that it inherits the identity and privileges of the victim to perform an undesired function on the victim's behalf, like change the victim's e-mail address, home address, or password, or purchase something. CSRF attacks generally target functions that cause a state change on the server but can also be used to access sensitive data.

Detection Mechanisms for Cross-Site Scripting Attacks

- o String Analysis
- o Bounded Model Checking
- o Fault Injection
- o Taint Propagation
- o Comparing User Input with Known Untrusted Scripts
- o Browser-Enforced Embedded Policies
- o Syntactic Structure Analysis
- o Lattice Based Approach
- o Proxy Based Approach
- o Properly escape all untrusted data based on the HTML context
- o Validate the length, characters, format, and business rules on the data before accepting the input
- o Use auto-sanitization libraries such as OWASP's AntiSamy or the Java HTML Sanitizer Project
- o Content Security Policy (CSP) to defend against XSS

Q. 8. (a) Write HTTP status codes series and their meaning. **(4)**

Ans. HTTP status codes are standard response codes given by web site servers on the internet. The codes help identify the cause of the problem when a web page or other resource does not load properly. The term HTTP status code is actually the common term for the HTTP status line that includes both the *HTTP status code* and the *HTTP reason phrase*. HTTP status codes are sometimes called browser error codes or internet error codes. All HTTP response status codes are separated into five classes. The first digit of the status code defines the class of response. The last two digits do not have any class or categorization role. There are five values for the first digit:

- 1xx (Informational): The request was received, continuing process
- 2xx (Successful): The request was successfully received, understood, and accepted
- 3xx (Redirection): Further action needs to be taken in order to complete the request
- 4xx (Client Error): The request contains bad syntax or cannot be fulfilled
- 5xx (Server Error): The server failed to fulfill an apparently valid request

1XX Informational response:

An informational response indicates that the request was received and understood. It alerts the client to wait for a final response. The message consists only of the status line and optional header fields, and is terminated by an empty line. Examples: 100 Continue, 101 Switching Protocols, 102 Processing, 103 Early Hints

2xx Success:

This class of status codes indicates the action requested by the client was received, understood and accepted. Examples: 200 OK, 201 Created, 202 Accepted, 203 Non-Authoritative Information , 204 No Content

3xx Redirection:

This class of status code indicates the client must take additional action to complete the request. Many of these status codes are used in URL redirection. A user agent may carry out the additional action with no user interaction only if the method used in the second request is GET or HEAD. A user agent may automatically redirect a request. A user agent should detect and intervene to prevent cyclical redirects. Examples:300 Multiple Choices, 301 Moved Permanently, 302 Found (Previously "Moved temporarily"),303 See Other , 304 Not Modified, 305 Use Proxy

4xx Client errors

This class of status code is intended for situations in which the error seems to have been caused by the client. Except when responding to a HEAD request, the server *should* include an entity containing an explanation of the error situation, and whether it is a temporary or permanent condition. These status codes are applicable to any request method. User agents *should* display any included entity to the user. . Examples:400 Bad Request, 401 Unauthorized, 402 Payment Required, 403 Forbidden, 404 Not Found, 405 Method Not Allowed etc.

5xx Server errors:

The server failed to fulfill a request. Response status codes beginning with the digit "5" indicate cases in which the server is aware that it has encountered an error or is otherwise incapable of performing the request. Except when responding to a HEAD request, the server *should* include an entity containing an explanation of the error situation, and indicate whether it is a temporary or permanent condition. Likewise, user agents *should* display any included entity to the user. These response codes are applicable to any request method. Examples: 500 Internal Server Error, 501 Not Implemented, 502 Bad Gateway, 503 Service Unavailable, 504 Gateway Timeout etc.

Q. 8. (b) Explain JSP directives, declarations and scriptlets. **(4.5)**

Ans. JSP Scriptlet : Scriptlet tag allows to write Java code into JSP file.JSP container moves statements in _jspservice() method while generating servlet from jsp.For each request of the client, service method of the JSP gets invoked hence the code inside the Scriptlet executes for every request. A Scriptlet contains java code that is executed every time JSP is invoked.

Syntax of Scriptlet tag: <% java code %>

JSP Declaration: A declaration tag is a piece of Java code for declaring variables, methods and classes. If we declare a variable or method inside declaration tag it means that the declaration is made inside the servlet class but outside the service method. We can declare a static member, an instance variable (can declare a number or string) and methods inside the declaration tag.

Syntax of declaration tag: <%! Dec var %>

JSP directives

The **jsp directives** are messages that tells the web container how to translate a JSP page into the corresponding servlet.

Syntax of JSP Directive: <%@ directive attribute= "value" %>

There are three types of directives:

1. page directive: The page directive defines attributes that apply to an entire JSP page. The attributes of JSP page directive are import content, Type, extends ,info, buffer, language,isELIgnored etc

Syntax of JSP page directive: <%@ page attribute="value" %>

2. include directive: The include directive is used to include the contents of any resource it may be jsp file, html file or text file. The include directive includes the original content of the included resource at page translation time. The advantage of Include directive is Code Reusability

Syntax of include directive: <%@ include file="resourceName" %>

3. taglib directive: The JSP taglib directive is used to define a tag library that defines many tags. We use the TLD (Tag Library Descriptor) file to define the tags. In the custom tag section we will use this tag so it will be better to learn it in custom tag.

Syntax JSP Taglib directive: <%@ taglib uri = "uri of the tag library" prefix = "prefix of tag library" %>

Q. 8. (c) Explain Trojan horse, worm and trapdoor. (4)

Ans. Trojan Horse: A destructive program that masquerades as a benign application. Unlike viruses, Trojan horses do not replicate themselves but they can be just as destructive. One of the most insidious types of Trojan horse is a program that claims to rid your computer of viruses but instead introduces viruses onto your computer. Trojan horses are broken down in classification based on how they breach systems and the damage they cause. The seven main types of Trojan horses are:

- Remote Access Trojans
- Data Sending Trojans
- Destructive Trojans
- Proxy Trojans
- FTP Trojans
- security software disabler Trojans
- denial-of-service attack (DoS) Trojans

WORM: A computer worm is a standalone malware computer program that replicates itself in order to spread to other computers. Often, it uses a computer network to spread itself, relying on security failures on the target computer to access it. Worms almost always cause at least some harm to the network, even if only by consuming bandwidth. Many worms are designed only to spread, and do not attempt to change the systems they pass through. Any code designed to do more than spread the worm is typically referred to as the "payload". Typical malicious payloads might delete files on a host system; encrypt files in a ransomware attack, or exfiltrate data such as confidential documents or passwords. Computer worms were spread through infected storage media, such as floppy diskettes, which, when mounted on a system, would infect other storage devices connected to the victim system. USB drives are still a common vector for computer worms. Email worms spread by creating and sending outbound messages to all the addresses in a user's contacts list.

Trapdoors: A trap door is a secret entry point into a program that allows someone that is aware of the trap door to gain access without going through the usual security access procedures. Trap doors have been used legitimately for many years by programmers to debug and test programs. Trap doors become threats when they are used by unscrupulous programmers to gain unauthorized access. It is difficult to implement operating system controls for trap doors. Security measures must focus on the program development and software update activities.

Because it is most commenly appearing new browser versions to supress the starvation due to workload on browser. A trap doors in a computer system is a method of bypassing normal authentication, securing remote access to a computer, obtaining access to plaintext, and so on, while attempting to remain undetected. The trap doors may take the form of an installed program or may subvert the system through a rootkit. The threat of trap doors surfaced when multiuser and networked operating systems became widely adopted, a class of active infiltration attacks that use "trapdoor" entry points into the system to bypass security facilities and permit direct access to data.