

# 314448: SOFTWARE LABORATORY – III

TE-IT (2015 Course)

Semester - I

Teaching Scheme		Examination Scheme	
Practical :	2 Hrs. / Week	Term Work	50 Marks

**Savitribai Phule Pune University**

सावित्रीबाई फुले पुणे विद्यापीठ



**Preamble:**

A major component of the course is a Graphical User Interface development. The objective is to develop a GUI by using concepts learned from Software Engineering and Project management. At the beginning of the course, Course Teacher will form project teams with maximum 3 members. During the semester, the project team will work together through all the phases of development cycle up to design, from an initial feasibility study to designing, after designing phase students will deploy the designed system and will make a series of presentations and reports of the work.

**Prerequisites:**

1. Programming fundamentals.
2. Problem solving skills.

**Course Objectives :**

1. To understand the nature of software complexity in various application domains, disciplined way of software development and software life cycle process models.
2. To introduce principles of agile software development, the SCRUM process and agile practices.
3. To know methods of capturing, specifying, visualizing and analyzing software requirements.
4. To understand concepts and principles of software design and architecture.
5. To understand user-centric design approach.
6. To apply principles of designing for effective user interfaces.

**Course Outcomes :**

1. To identify the needs of users through requirement gathering.
2. To apply the concepts of Software Engineering process models for project development.
3. To apply the concepts of HCI for user-friendly project development.
4. To deploy website on live webserver and access through URL.
5. To understand, explore and apply various web technologies.
6. To develop team building for efficient project development.

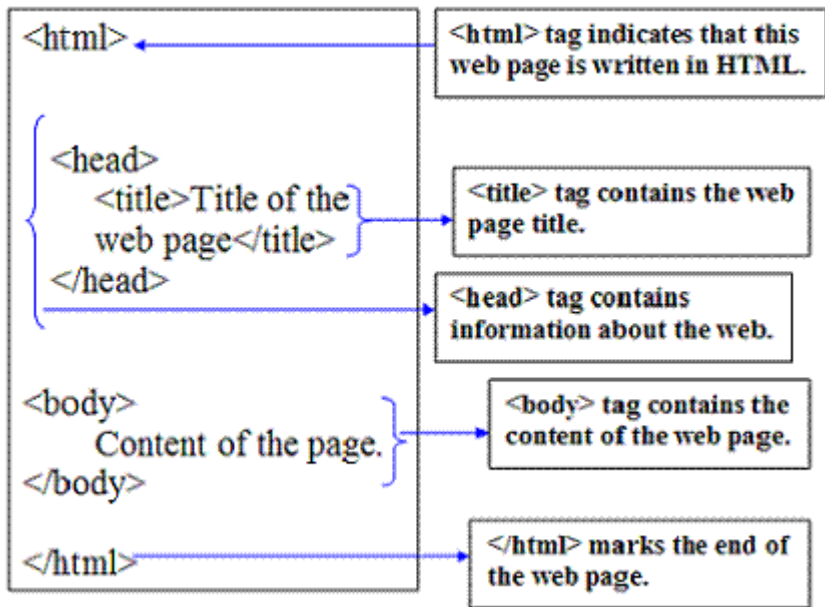
## INDEX

Sr. No	Title	Page No.
1	Using HTML5 layout tags develop informative page with sections which include various images, links to other pages for navigation, make use of all possible formatting (for example font, color etc.).	4
2	Apply CSS properties Border, margins, Padding, Navigation, dropdown list to page created in first assignment.	11
3	Create form in HTML with all form elements apply form validations (e.g. Email, mobile, Pin code, Password).	16
4	Validate URL, Email, Required using functions empty, preg_match, filter_var in PHP.	21
5	Understand servlet life cycle, create login page and apply proper validations with appropriate messages using doGet()/ doPost() methods.	24
6	Develop website using any CMS tool which falls into one of the categories blog, social networking, News updates, Wikipedia, E-commerce store. Website must include home page, and at least 3 forms (with Validation), use at list HTML5, PHP, CSS/Bootstrap, JavaScript web technologies. No database support is needed. Deploy website on live webserver and access through URL.	33

## Group A :Website Design (HTML5, CSS, Bootstrap)

### Assignment No. : 1

Aim	
Using HTML5 layout tags develop informative page with sections which include various images, links to other pages for navigation, make use of all possible formatting (for example font, color etc.).	
Objective(s)	
1	It gives an introduction into the basic concepts of HTML tags.
2	It gives an understanding of various new features introduced in HTML5.
3	It gives understanding of how you can use it to build interactive web page.
<b>Note</b>	As students may not be having knowledge of HTML Instructor should first give introduction to basic HTML tags before starting the actual assignment

Theory	
<b>HTML</b> <ul style="list-style-type: none"><li>Structure of HTML document</li></ul>  <p>The diagram illustrates the structure of an HTML document. It shows a code block on the left with the following tags: <code>&lt;html&gt;</code>, <code>&lt;head&gt;</code>, <code>&lt;title&gt;Title of the web page&lt;/title&gt;</code>, <code>&lt;/head&gt;</code>, <code>&lt;body&gt;</code>, <code>Content of the page.</code>, <code>&lt;/body&gt;</code>, and <code>&lt;/html&gt;</code>. On the right, five boxes provide descriptions for these tags, with arrows pointing from the corresponding tags in the code block to the descriptions: <code>&lt;html&gt;</code> tag indicates that this web page is written in HTML.; <code>&lt;title&gt;</code> tag contains the web page title.; <code>&lt;head&gt;</code> tag contains information about the web.; <code>&lt;body&gt;</code> tag contains the content of the web page.; <code>&lt;/html&gt;</code> marks the end of the web page.</p>	

- **Few Basic HTML Tags**

Tag	Description
<html> ... </html>	Declares the Web page to be written in HTML
<head> ... </head>	Delimits the page's head
<title> ... </title>	Defines the title (not displayed on the page)
<body> ... </body>	Delimits the page's body
<h <i>n</i> > ... </h <i>n</i> >	Delimits a level <i>n</i> heading
<b> ... </b>	Set ... in boldface
<i> ... </i>	Set ... in italics
<center> ... </center>	Center ... on the page horizontally
<ul> ... </ul>	Brackets an unordered (bulleted) list
<ol> ... </ol>	Brackets a numbered list
<li> ... </li>	Brackets an item in an ordered or numbered list
 	Forces a line break here
<p>	Starts a paragraph
<hr>	Inserts a horizontal rule
	Displays an image here
<a href="..."> ... </a>	Defines a hyperlink

## HTML5

- *HTML5* is the latest and most enhanced version of HTML. HTML is not a programming language, but rather a mark up language. It is the fifth and current version of the HTML standard. It was published in October 2014 by the World Wide Web Consortium (W3C) to improve the language with support for the latest multimedia. Many new syntactic features are included.
- A web page being rendered in the browser consists of many things - logo, informative text, pictures, hyperlinks, navigational structure and more. HTML5 offers a set of markup elements that allow you to create a structured layout for web pages.

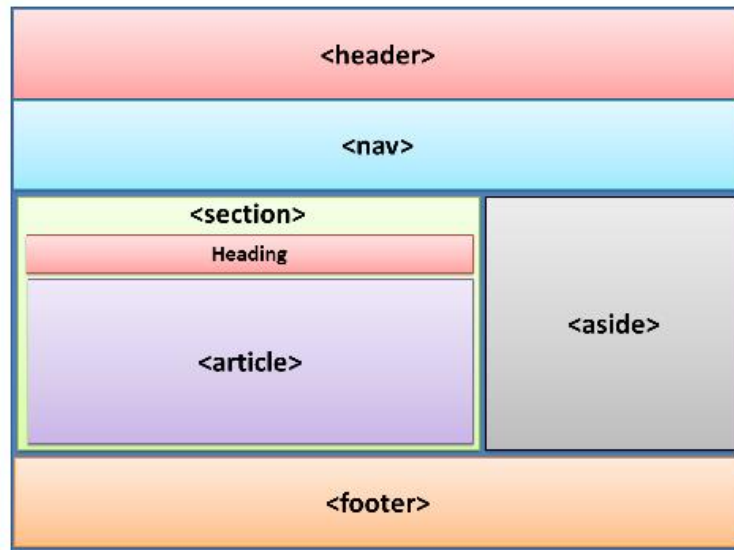
## Layouts

- Layout is the compilation of text and graphics on a page. Many websites have multiple columns in their layout - they are formatted like a magazine or newspaper. Many websites achieved this HTML layout using tables, frames or using DIV tag.
- With frames, you can display more than one HTML document in the same browser window.

Each HTML document is called a frame, and each frame is independent of the others.

- HTML<div>tag defines a division or a section in an HTML document. It is used to group block-elements to format them with CSS. It is very often used together with CSS, to layout a web page. It acts as a container unit that encapsulates other page elements and divides the HTML document into sections.
- HTML5 offers new semantic elements that define the different parts of a web page:

**A sample page layout is shown in figure**



**HTML5 supports below tags for layouts.**

Tag	Description
header	Defines a header for a document or a section
nav	Defines a container for navigation links
section	Defines a section in a document
article	Defines an independent self-contained article
aside	Defines content aside from the content (like a sidebar)
footer	Defines a footer for a document or a section

details	Defines additional details
summary	Defines a heading for the details element

- **Header Element**

The `<header>` element to replace the `div` tag that has largely filled the role so far. It represents the header of the whole page or a section of it

**Example:**

```
<header>
<h1>This is pageheading</h1>
</header>
```

- **Section Element**

**The `<section>` element** define what the different parts of a page's content are and how those various parts are related. The `<section>` element is a thematic grouping of content whereas the `<div>` doesn't have any such restriction.

A section can also have its own `<header>` elements as well as a `<footer>` element. The section element is analogous to a section of a printed book that contains chapters or a section of a newspaper that contains news items.

**Example:**

```
<section>
<h1>This is a section heading</h1>
<p>
Hello world! Hello world! Hello world!
Hello world! Hello world! Hello world!
Hello world! Hello world! Hello world!
</p>
</section>
```

- **Nav Element**

The `<nav>` element is for "major navigation blocks"\*. It can go in the header or article tags or it can be on its own. *The `<nav>` element represents a section with navigation links.* The `<nav>` section can contain links to the other pages from the website or to other parts of the same web page. It is recommended that you use `<nav>` only for the main navigational structures and not for minor set of hyperlinks.

**Example:**

```
<nav>
<ul>
<li><a href="#">Home</a></li>
<li><a href="#">About Us</a></li>
<li><a href="#">Contact Us</a></li>
</ul>
</nav>
```

- **Article Element**

The <article> element contains content unique to each page. It represents an independent item section of content such as a blog post, forum post or a comment.

**Example:**

```
<article>
<p> Some contents here </p>
</article>
```

- **Aside Element**

The <aside> element is intended to house content that is related to the surrounding content but at the same time is a standalone piece of content in itself. If you take out the <aside> from the page it shouldn't change or alter the meaning or clarity of the main page content. Think of it as a sidebar that gives some extra, related yet standalone information about the topic being discussed. Some examples of <aside> include - extra information, related links and contextual advertisements.

**Example:**

```
<aside>
<figure>

<figcaption>Figure caption goes here</figcaption>
</figure>
<p>
    Hello world! Hello world! Hello world!
    Hello world! Hello world! Hello world!
</p>
</aside>
```

- **Footer Element**

The <footer> element represents the footer of the whole page or a <section> element. It is intended to contain footer information such as copyright notice.



**Example:**

**<footer>**

**<hr />**

**Copyright (C) 2013. All rights reserved.**

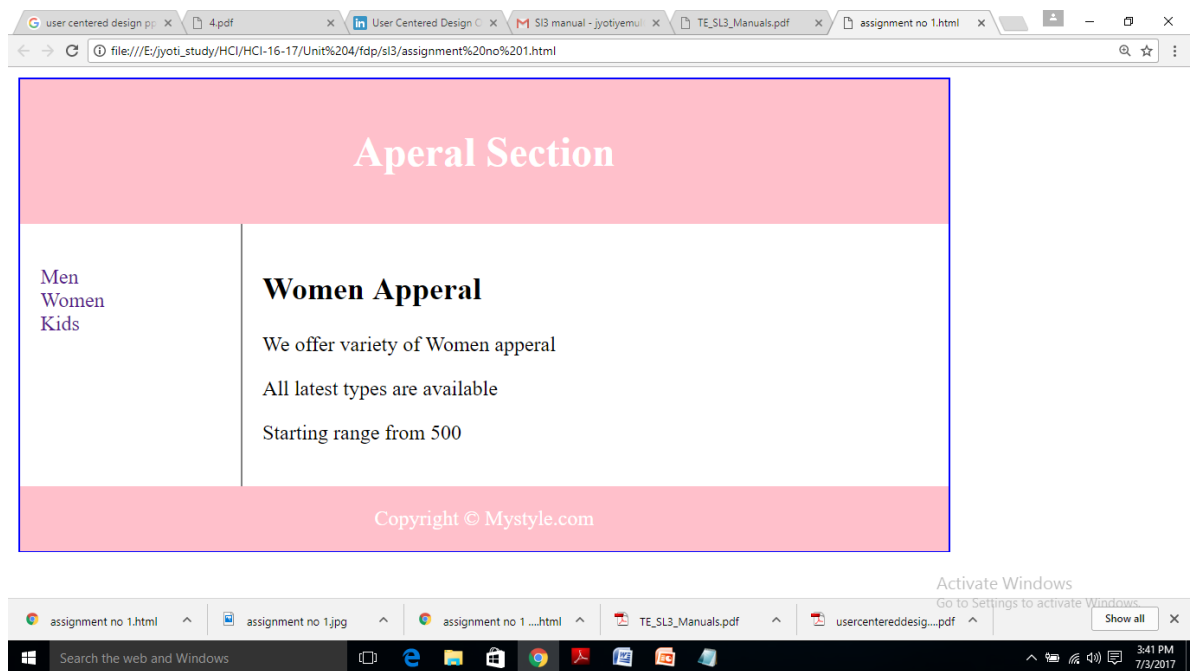
**</footer>**

The header, nav and footer tend to stay the same on every page. Once they are set you can copy and paste them into new pages then add the article.

## Conclusion

**HTML tags and HTML5 Layout tags have been studied and used for creation of interactive web page.**

- **Expected sample Output**



### **Lab. Based FAQ**

1. What are HTML elements?
2. What are HTML5 Layout tags?
3. What are various HTML formatting tags?

## Assignment No. : 2

### **Aim**

Apply CSS properties Border, margins, Padding, Navigation, dropdown list to page created in first assignment.

### **Objective(s)**

- |          |   |
|----------|---|
| <b>1</b> | It gives an introduction into the basic concepts of CSS.                        |
| <b>2</b> | It gives an understanding of how you can use CSS to build interactive web page. |

### **Theory**

#### **CSS**

- Cascading Style Sheets (CSS) form the presentation layer of the user interface.
  - Structure (XHTML)
  - Behavior (Client-Side Scripting)
  - **Presentation (CSS)**

#### **Types of CSS**

1. External style sheet
2. Embedded styles
3. Inline styles

- **Inline Style**

- Inline styles
- Add styles to each tag within the HTML file
- Use it when you need to format just a single section in a web page
- **Example**  
`<h1 style="color:red; font-family: sans-serif">IU</h1>`

- **Internal Stylesheet Embedded**

- A style is applied to the entire HTML file
- Use it when you need to modify all instances of particular element (e.g., h1) in a web page
- **Example**  
`<style>`  
`h1 {color:red; font-family:sans-serif}`  
`</style>`

- **External Stylesheet**

- External style sheets
- An external style sheet is a text file containing the style definition (declaration)
- Use it when you need to control the style for an entire web site
- Example
- `h1, h2, h3, h4, h5, h6 {color:red; font-family:sans-serif}`
- Save this in a new document using a .css extension

- **CSS Syntax**

A CSS rule set consists of a selector and a declaration block:

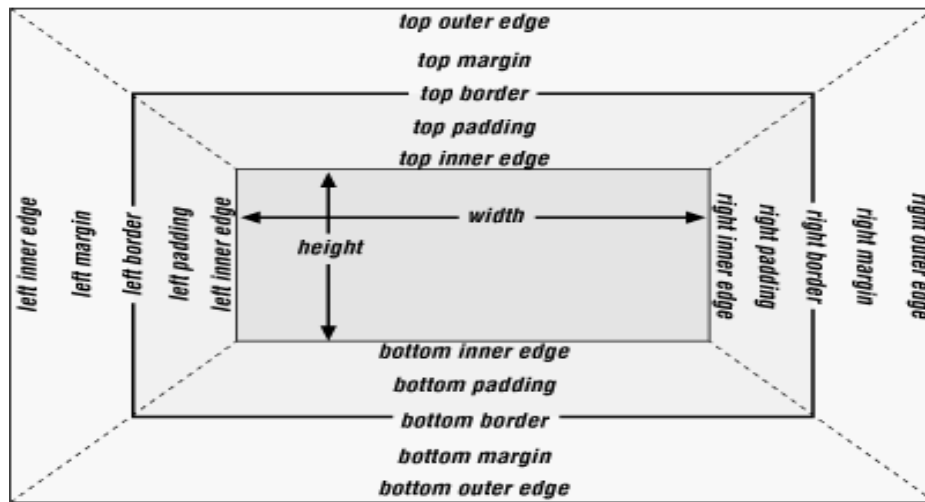


The **selector** points to the HTML element you want to style. The **declaration** block contains one or more declarations separated by semicolons. Each declaration includes a property name and a value, separated by a colon.

- **The CSS Box Model**

- 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 HTML elements, and it consists of: margins, borders, padding, and the actual content.
- It allows us to place a border around elements and space elements in relation to other elements.

The image below illustrates the box model.



- **CSS Border Properties**

Property	Description
<b>border</b>	<b>all the individual border properties in one property. This is called a shorthand property.</b>
<b>Border-style</b>	<b>none  dotted   dashed   solid   double   groove   ridge   inset   outset</b>
<b>border-width</b>	<b>is used to set the width of the border. The width is set in pixels, or by using one of the three pre-defined values: thin, medium, or thick.</b>
<b>border-color</b>	<b>is used to set the color of the border</b>

- **Margin**

The margin clears an area around an element (outside the border). The margin does not have a background color, and is completely transparent. The top, right, bottom, and left margin can be changed independently using separate properties. A shorthand margin property can also be used, to change all margins at once. It is also possible to use negative values, to overlap content.

Value	Description
auto	The browser calculates a margin
<i>length</i>	Specifies a margin in px, pt, cm, etc. Default value is 0px
%	Specifies a margin in percent of the width of the containing element
inherit	Specifies that the margin should be inherited from the parent element

#### • Padding

The padding clears an area around the content (inside the border) of an element. The padding is affected by the background color of the element. The top, right, bottom, and left padding can be changed independently using separate properties. A shorthand padding property can also be used, to change all paddings at once.

#### • Navigation

There are two ways to create a horizontal navigation bar. Using **inline** or **floating** list items. The display property of css is used for navigation bar. The value block will create a vertical navigation bar and inline will create a horizontal navigation bar.

#### Example:

**Display: inline | block**

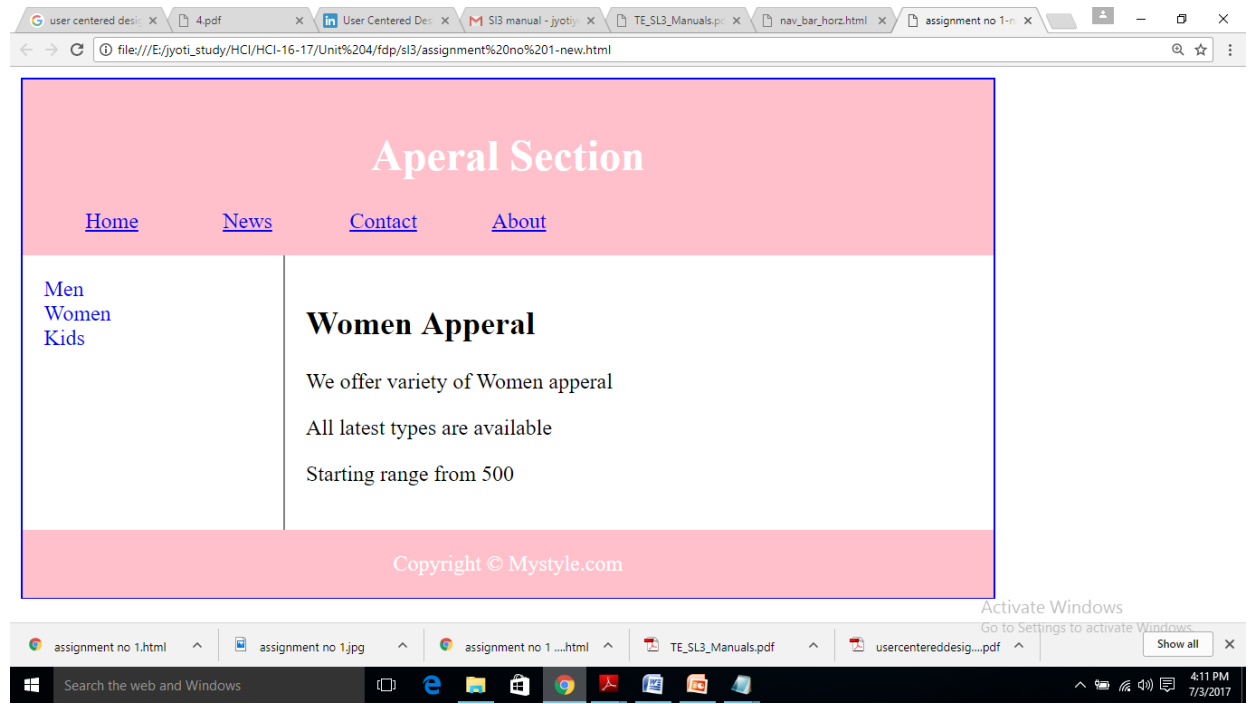
Drop Down list can be created by using HTML onordered lists tags <ul> and list items <li>. One can set below CSS properties for lists.

Property	Description
list-style	Sets all the properties for a list in one declaration
list-style-image	Specifies an image as the list-item marker Eg. url("image.gif");
list-style-position	Specifies if the list-item markers should appear inside or outside the content flow Inside   outside   initial
list-style-type	Specifies the type of list-item marker Circle   square   decimal   lower-alpha   lower-greek   lower-latin   lower-roman   upper-alpha   upper-latin   upper-roman

#### Conclusion

Various CSS properties have been studied and used for creation of interactive web page.

- **Expected sample Output**



### Lab. Based FAQ

- 1.What is CSS?
2. What are the types of writing CSS?
3. What are various CSS Properties?
4. What is CSS Box model?

## Group B : Website GUI Validation (JavaScript, PHP)

### Assignment No. : 3

#### Aim

Create form in HTML with all form elements apply form validations (e.g. Email, mobile, Pin code, Password).

#### Objective(s)

- |   |   |
|---|---|
| 1 | It gives an introduction into the basic concepts of Javascript.           |
| 2 | It gives understanding of how you can use Javascript for form validation. |

#### Theory

##### • Introduction to HTML Forms:

- Forms are user interfaces for data input
- Main application: to provide user input for
  - programs and databases located on a web server
  - local (client-side) scripts associated with the form
- Server-based programs may return data to the client as a web page
- Client-side scripts can read input data
  - To validate the data, prior to sending to server
  - To use in local processing which may output web page content that is displayed on the client

##### Input types:

- text
- checkbox
- radio (buttons)
- select (options)
- textarea
- password
- button
- submit
- reset
- hidden
- file
- image

**Tell us what you think**

Name

Address

How did you hear about this web site?

A friend told me ☐

Via a search engine ☐

Followed a link (URL) ☐

How do you rate this site?

Please write your comments:

Do you want to receive any further information:

Thank you



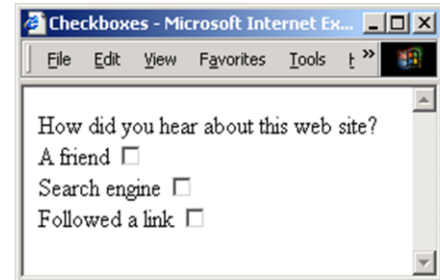
- The input element: `type="text"`
- The **type** attribute specifies the type of user input
- The **name** attribute gives an identifier to the input data
- The **size** attribute specifies the length of the input field
- The **value** attribute specifies an initial value for the text (optional)



```
<form method="POST" action="comments.pl">  
  <h2>Tell us what you think</h2>  
  Name <input name="name" type="text" size="20"><br>  
  Address <input name="address" type="text" size="30">  
</form>
```

## The `input` element: `type="checkbox"`

- The **name** attribute is used to define a set of checkboxes
- The **value** attribute identifies the individual checkbox
- If the **checked** attribute is set the box is initially checked



```
How did you hear about this web site?<br>
A friend
<input type="checkbox" name="howdid" value="friend"><br>
Search engine
<input type="checkbox" name="howdid" value="engine"><br>
<!-- etc -->
```

21

### Conclusion

Various Javascript functionalities have been studied and used for interactive web page design and form validations.

- **Expected sample Output**

The screenshot shows a web browser window with multiple tabs. The active tab displays a registration form titled "Registration Form". The form contains two text input fields labeled "First Name:" and "Last Name:", and a "Submit" button. The browser's address bar shows the file path: file:///E:/jyoti\_study/TEIT14-15-sem-I/WebTechnology/WTLab/javascript/programs/form.html. Below the browser window, the Windows taskbar is visible, showing several open applications and the system clock indicating 4:16 PM on 7/3/2017.

user centered desi x 4.pdf x User Centered Des x S13 manual - jyoti x TE\_SL3\_Manuals.p x Form Validation x Form Validation x

file:///E:/jyoti\_study/TEIT14-15-sem-I/WebTechnology/WTLab/javascript/programs/form.html

## Registration Form

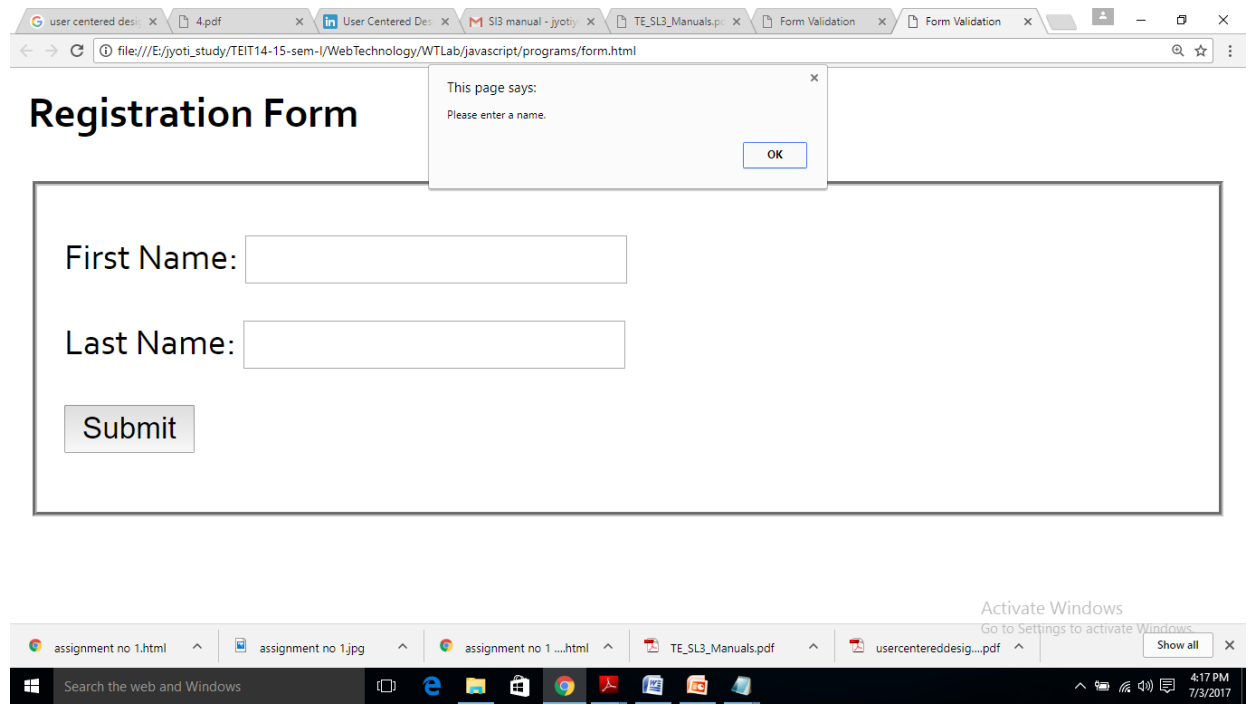
First Name:

Last Name:

assignment no 1.html assignment no 1.jpg assignment no 1 ...html TE\_SL3\_Manuals.pdf usercentereddesig...pdf

Search the web and Windows

4:16 PM 7/3/2017



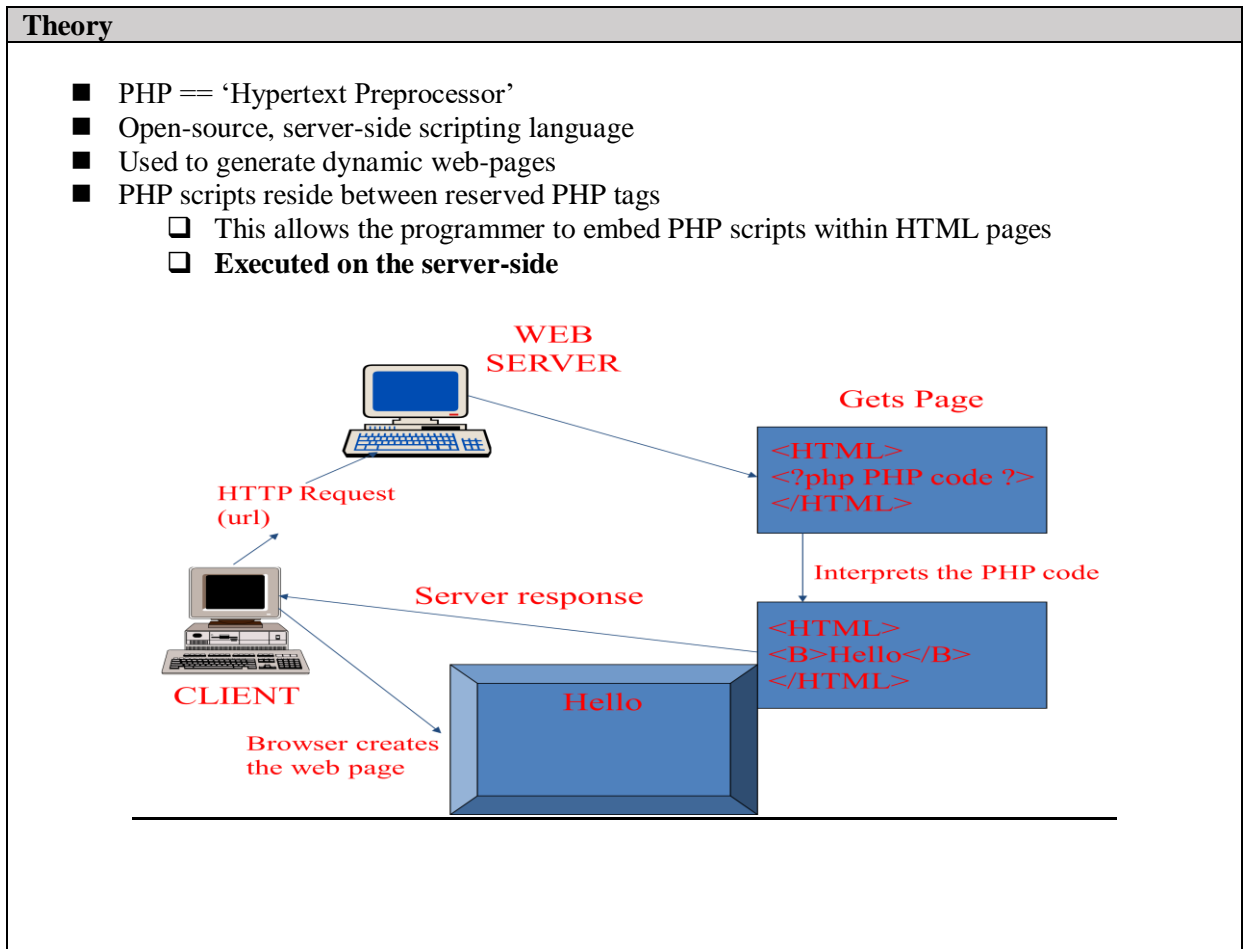
### Lab. Based FAQ

1. What is Javascript?
2. What is client-side and server-side validation?
3. What is Document Object Model?
4. How to embed javascript in a web page?
5. What are event handlers in JavaScript?

## Group B : Website GUI Validation (JavaScript, PHP)

### Assignment No. : 4

Aim	
Validate URL, Email, Required using functions empty, preg_match, filter_var in PHP.	
Objective(s)	
1	It gives an introduction into the basic concepts of PHP.
2	It gives understanding of how you can use PHP for form validation.
3	It gives of understanding of various web servers and server-side validation
Note	<b>Instructor should teach basics of PHP before teaching the assignment</b> <b>This assignment can be taken in easyphp(available free on <a href="http://www.easyphp.org">http://www.easyphp.org</a>) or by using Xamp/Wamp</b>



Instructor should teach following concepts:

1. **What does PHP code look like?**
2. **Comments in PHP**
3. **Variables in PHP**
4. **Echo**
5. **PHP - Forms**

- In PHP to check whether an email address is valid is to use PHP's filter\_var() function. To validate email-address in php use the FILTER\_VALIDATE\_EMAIL which is a PHP email validation filter.
- This PHP filters is used to validate and filter data coming from insecure sources, like user input. As of PHP 5.2.0, the filter functions are enabled by default. There is no installation needed to use these functions.filter\_var — Filters a variable with a specified filter.

- **Example:**

```
<?php
$email_address = "me@example.com";
if (filter_var($email_address, FILTER_VALIDATE_EMAIL))
{
    // The email address is valid
} else
{
    // The email address is not valid
}
?>
```

- The preg\_match() function searches a string for pattern, returning true if the pattern exists, and false otherwise.

- **Example:**

```
$name = test_input($_POST["name"]);
if (!preg_match("/^[a-zA-Z ]*$/", $name)) {
    $nameErr = "Only letters and white space allowed";
}
```

The code below shows a way to check if a URL address syntax is valid (this regular expression also allows dashes in the URL). If the URL address syntax is not valid, then store an error message:

```

$website=test_input($_POST['website']);
if (!preg_match("/^b(?:(:?https?|ftp):\\|www\\.)([-a-z0-9+&@#/%?~_!:,;]*[-a-z0-9+&@#/%?~_!:/i",$website))
{
    $websiteErr = "Invalid URL";
}

```

It accepts a regular expression followed by a string to match against the expression, and then followed by an array that is filled with results of the match

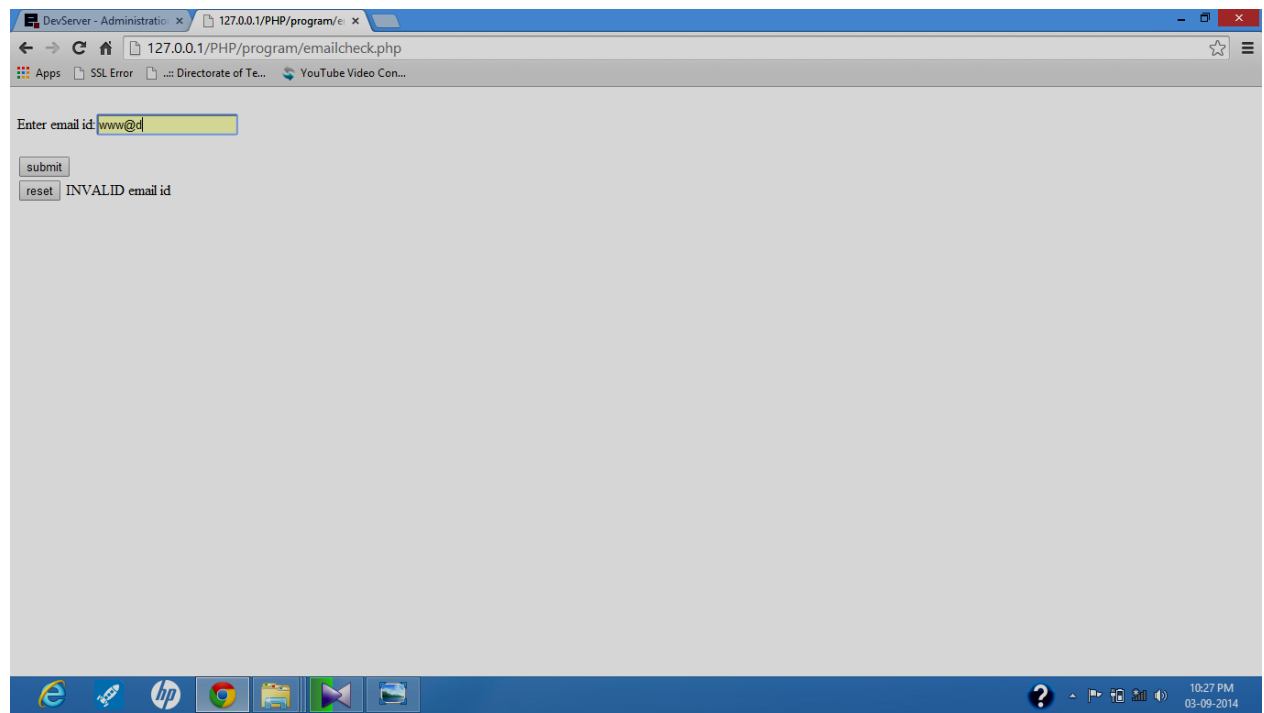
The empty() determines whether a variable is empty.

Syntax: **boolempty** (\$var )

## Conclusion

The various features of PHP is studied and used PHP for form validation.

## • Expected sample Output



### **Lab. Based FAQ**

1. What is PHP? What are the benefits of using php?
2. How form fields validation is done in PHP?
3. What is difference between client-side and server-side validation?
4. What are the various functions used for form validation?



## Group C : Website Working (Java Servlet)

### Assignment No. : 5

#### Aim

Understand servlet life cycle, create login page and apply proper validations with appropriate messages using doGet()/ doPost() methods.

#### Objective(s)

- |   |   |
|---|---|
| 1 | It gives an introduction into the basic concepts of Java Servlets.            |
| 2 | It gives understanding of how you can use Servlets to add business logic.     |
| 3 | It gives of understanding of various web servers and server-side programming. |

#### Theory

##### Before Servlets :

## CGI Scripts

- CGI stands for “Common Gateway Interface”

Client sends a request to server

Server starts a CGI script

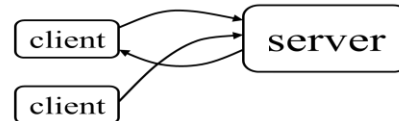
Script computes a result for server and quits

Server returns response to client

Another client sends a request

Server starts the CGI script again

Etc.



2

# Servlets

- A **servlet** is like an applet, but on the server side

Client sends a request to server

Server starts a servlet

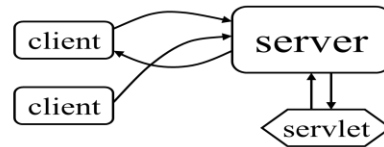
Servlet computes a result for server and *does not quit*

Server returns response to client

Another client sends a request

Server calls the servlet again

Etc.



3

- A **servlet** is any class that implements the `javax.servlet.Servlet` interface
- It is a simple java class, which is dynamically loaded on a web server and enhances the functionality of a web server.

- In practice, most servlets extend the `javax.servlet.http.HttpServlet` class

- Some servlets extend `javax.servlet.GenericServlet` instead

## Types of Servlet

- **Generic Servlet**

- `javax.servlet` (package)
- extends `javax.servlet.Servlet`
- service method

### • Http Servlet

- javax.servlet.http (package)
- extends javax.servlet.HttpServlet
- doGet(), doPost()....

## Life cycle of 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 designed to be called only once. It is called when the servlet is first created, and not called again for each user request. So, it is used for one-time initializations, just as with the init method of applets.
- The servlet is normally created when a user first invokes a URL corresponding to the servlet, but you can also specify that the servlet be loaded when the server is first started.
- When a user invokes a servlet, a single instance of each servlet gets created, with each user request resulting in a new thread that is handed off to doGet or doPost as appropriate. The init() method simply creates or loads some data that will be used throughout the life of the servlet.

### ■ The init method definition looks like this:

```
public void init() throws ServletException
{
    // Initialization code...
}
```

### ■ 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.
- Each time the server receives a request for a servlet, the server spawns a new thread and calls service. The service() method checks the HTTP request type (GET, POST, PUT, DELETE, etc.) and calls doGet, doPost, doPut, doDelete, etc. methods as appropriate.

■ **Service() method looks like this:**

```
public void service(ServletRequest request, ServletResponse
response) throws ServletException, IOException
{

}
```

- The service () method is called by the container and service method invokes **doGe, doPost, doPut, doDelete**, etc. methods as appropriate.
- So you have nothing to do with service() method but you override either doGet() or doPost() depending on what type of request you receive from the client.
- The doGet() and doPost() are most frequently used methods within each service request. Here is the signature of these two methods.

■ **The doGet() Method**

A GET request results from a normal request for a URL or from an HTML form that has no METHOD specified and it should be handled by doGet() method.

```
public void doGet(HttpServletRequest request, HttpServletResponse
response)
throws ServletException, IOException
{
// Servlet code
}
```

■ **The doPost() Method**

A POST request results from an HTML form that specifically lists POST as the METHOD and it should be handled by doPost() method.

```
public void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException
{
// Servlet code
}
```

## ■ The destroy() method :

The 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.

After the destroy() method is called, the servlet object is marked for garbage collection. The

**Destroy method definition looks like this:**

```
public void destroy()
{
    // Finalization code...
}
```

## Sample Code for Hello World:

Following is the sample source code structure of a servlet example

```
// Import required java libraries
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
// Extend HttpServlet class
public class HelloWorld extends HttpServlet
{
    private String message;
    public void init() throws ServletException
    {
        // Do required initialization
        message = "Hello World";
    }
    public void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException
    {
        // Set response content type
        response.setContentType("text/html");
        // Actual logic goes here.
        PrintWriter out = response.getWriter();
        out.println("<h1>" + message + "</h1>");
    }
    public void destroy()
    {
        // do nothing.
    }
}
```

## Conclusion

The servlets features have been studied and used for creating dynamic and interactive web-sites.

- **Expected sample Output**

Insert title here

localhost:8080/login/index.html

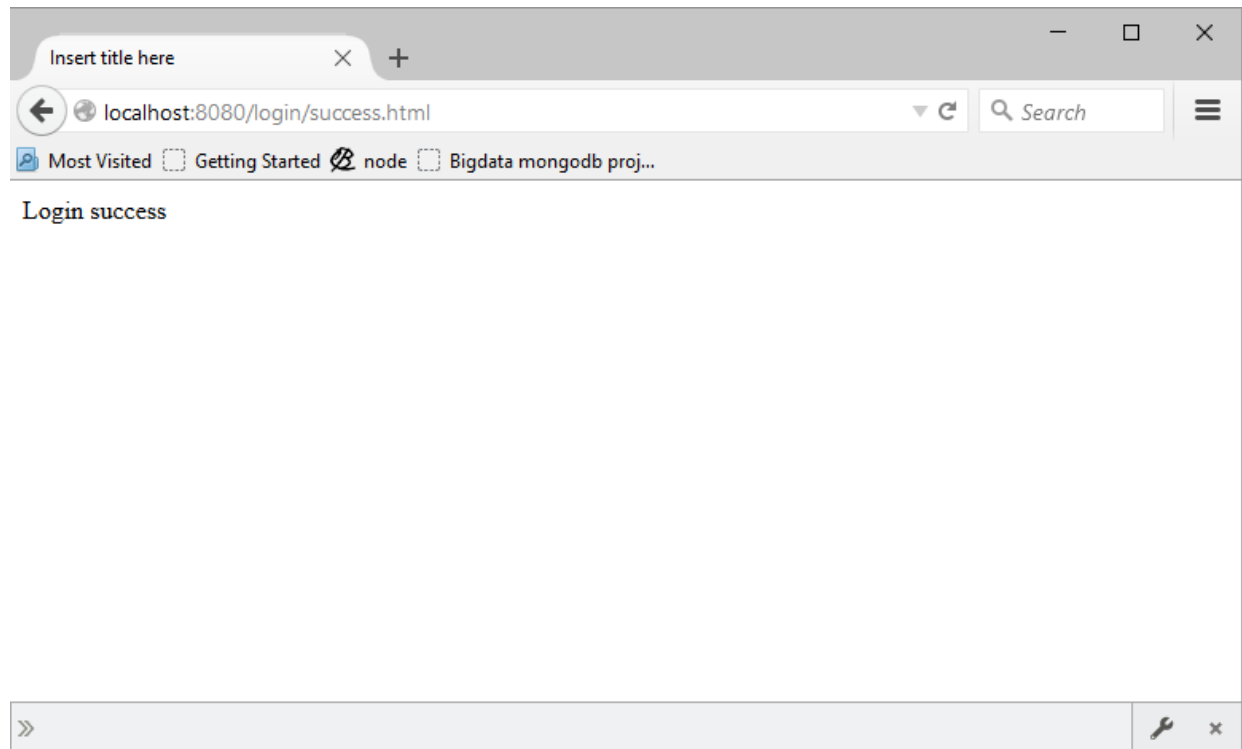
Most Visited Getting Started node Bigdata mongodb proj...

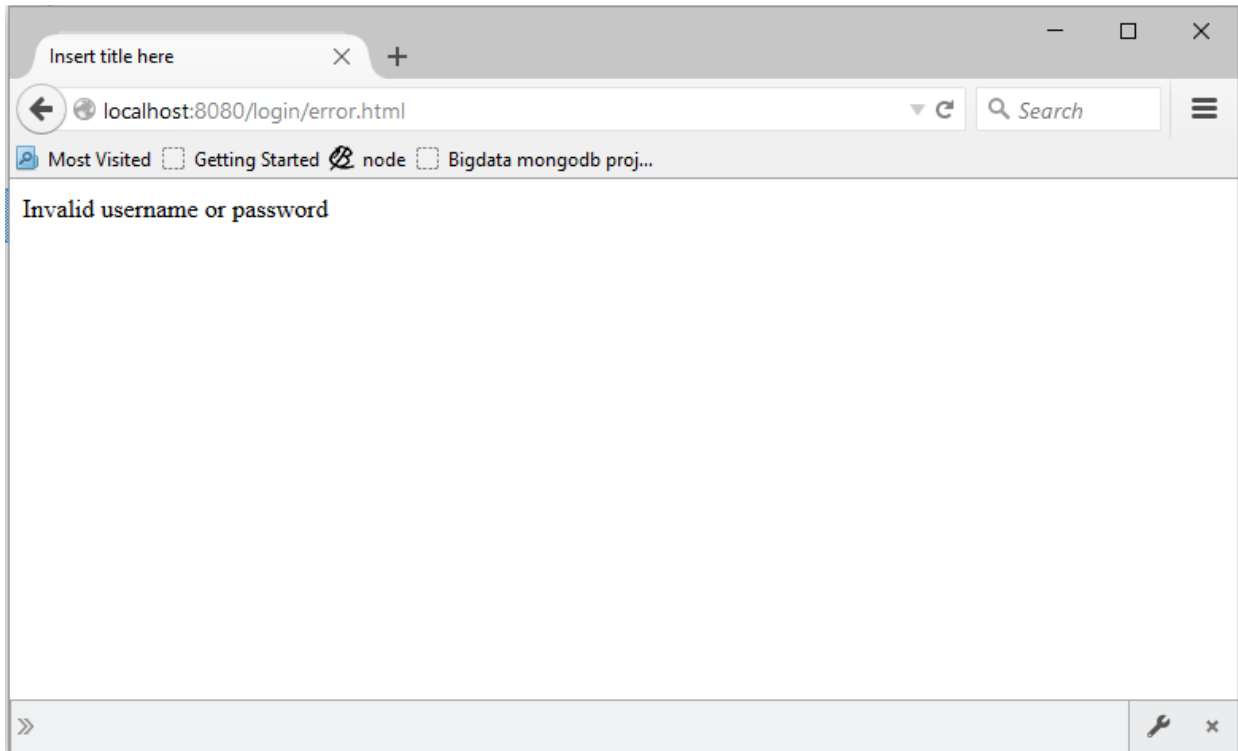
Sample login Example (try with username as "admin" and password as "admin" without quart )

Enter username : admin

Enter password : ●●●●●

Login





### Lab. Based FAQ

1. What is Servlet? What is the use of servlet?
2. Explain life-cycle of servlet?
3. What is servlet interface?
4. What is the difference between GenericServlet and HttpServlet?
5. What is the difference between the doGet () and doPost ()?



## Group D : Website Development (Mini-Project)

### Assignment No. : 6

Aim	
Develop website using any CMS tool which falls into one of the categories blog, socialnetworking, News updates, Wikipedia, E-commerce store. Website must include home page, and at least 3 forms (with Validation), use at list HTML5, PHP, CSS/Bootstrap, JavaScript web technologies. No databasesupport is needed. Deploy website on live webserver and access through URL.	

Objective(s)	
1	It gives an introduction into the basic concepts of CMS.
2	It gives understanding of how you can use various CMS tools such as Wordpress to create website.
3	It gives of understanding of how to deploy web-site on a web server and access through URL.

Theory	
<p>■ <b>Introduction to CMS</b></p> <ul style="list-style-type: none"><li>• Contents of websites may consist of text, graphics, sounds, movies and so on,</li><li>• Content Management Systems are used as a tool to publish such contents onto a website easily and efficiently, it also helps to separate web page design from content creation.</li><li>• Content Management Systems are usually developed using a combination of programming/scripting languages and database technologies, in which they work well together.</li><li>• In order to manage the process of creating and presenting information on web pages, this software consists of the following:<ul style="list-style-type: none"><li>• <b>Client software:</b> is used to present a user interface to help users add, change, or delete information on web pages, for example a web browser such as Internet Explorer or Mozilla Firefox are used to display web based user interfaces.</li><li>• <b>Database software:</b> is used as a tool to store data or information for future retrieval or manipulation, for example Microsoft SQL server, Oracle and MySQL databases.</li><li>• <b>Web programming languages and scripts:</b> these are used as a tool to communicate to the underlying database to extract, change, save, or remove data from and to the database, for example JSP, PHP, Java Servlets, Perl and so on.</li></ul></li></ul>	

- **Template page:** This is usually created by Hyper Text Mark up Language (HTML) to keep the website layout consistent in order to preserve the corporate image by keeping the same look and feel through out the system.

#### ■ Advantages of Content Management Systems

- It helps content authors to update or publish information onto the website without any need for web programming knowledge, which in turn reduces the cost of professional assistance.
- It allows content writers to concentrate on writing information without worrying about the design of the website. They can easily add, edit, and remove contents from and to the database by using simple user interfaces.
- It allows web content maintenance carried out easily and efficiently to meet business needs.
- It provides a consistent corporate image by keeping a consistent presentation and layout of web pages throughout the website by separating web page design tasks from content creation tasks.
- It helps web developers to concentrate on more important aspects of the website such web architecture, design, navigation and usability of the website without worrying about creating and updating information on the website.

**Few CMS Development tools are wordpress, Drupal, Jhoomla.**

#### ■ Wordpress

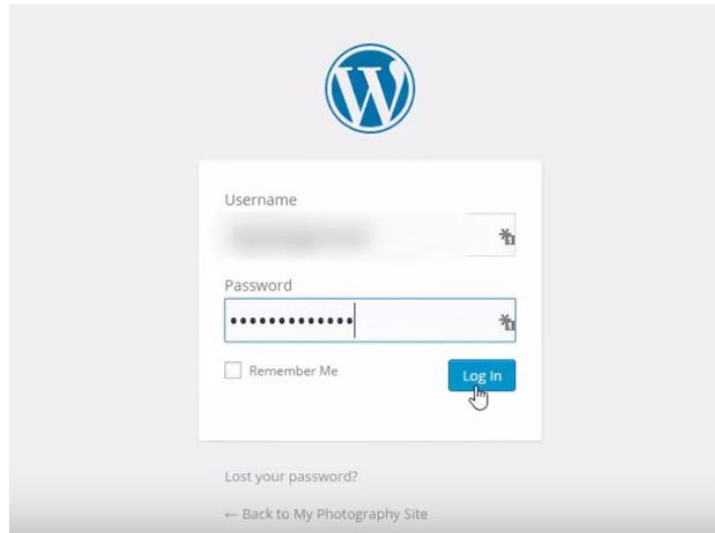
WordPress is a completely free tool used to create dynamic websites. It is most popular blogging tools on the web, making it easy for anybody to post their ideas, pictures, and audio/video. A blog is a type of website or part of a website that is maintained on a regular basis by it's owner with entries regarding commentary, reviews, opinions, and other media such as video. It will be used to create any type of websites such as Business Website, Informative Website, Review Website, Personal Website, Photo Gallery, etc.

### Example

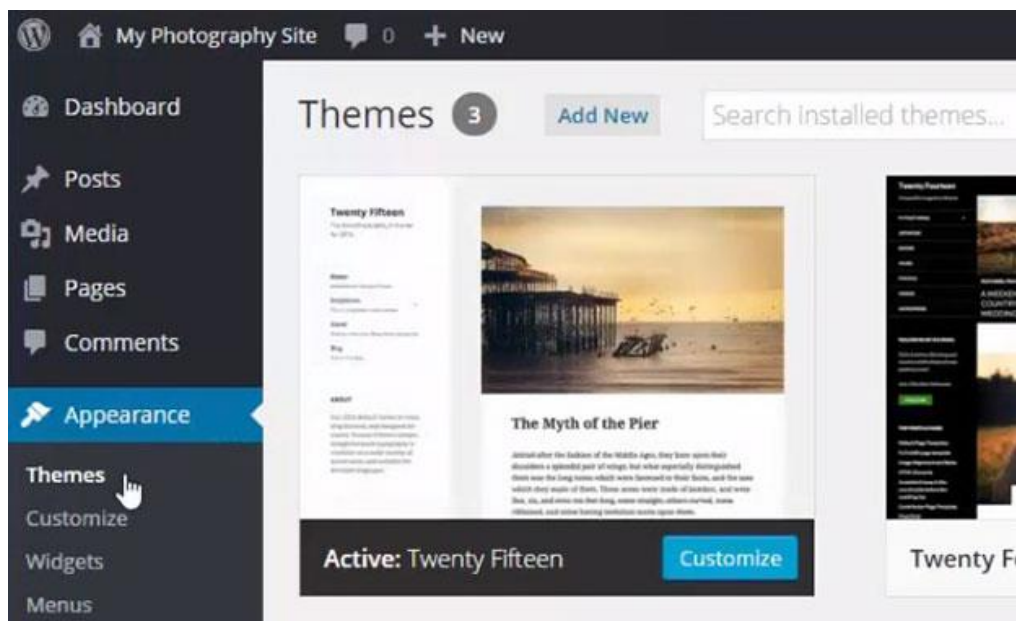
**Steps to create Photography website using Wordpress**

#### **1. Install Wordpress**

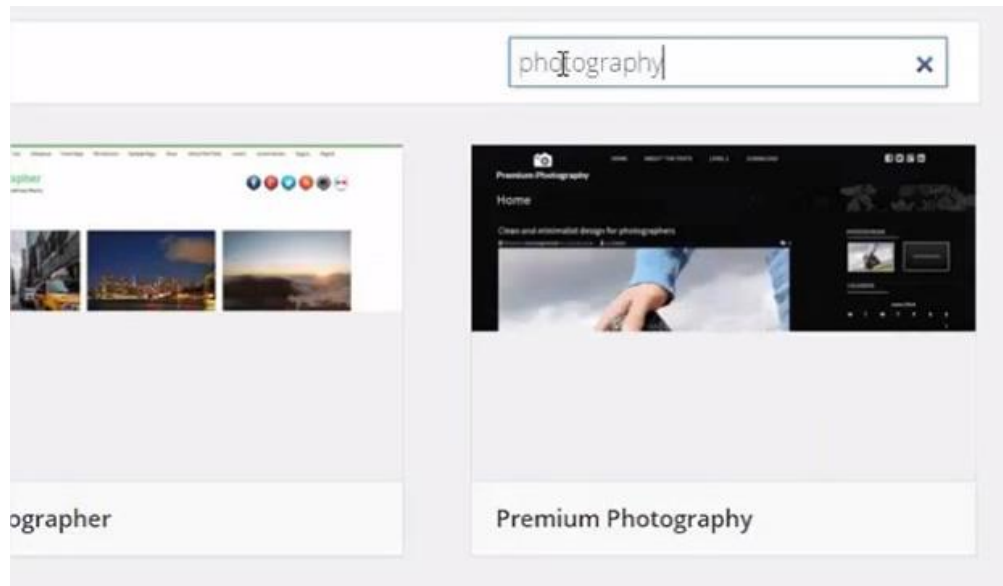
## 2. Now go ahead and login your WordPress admin



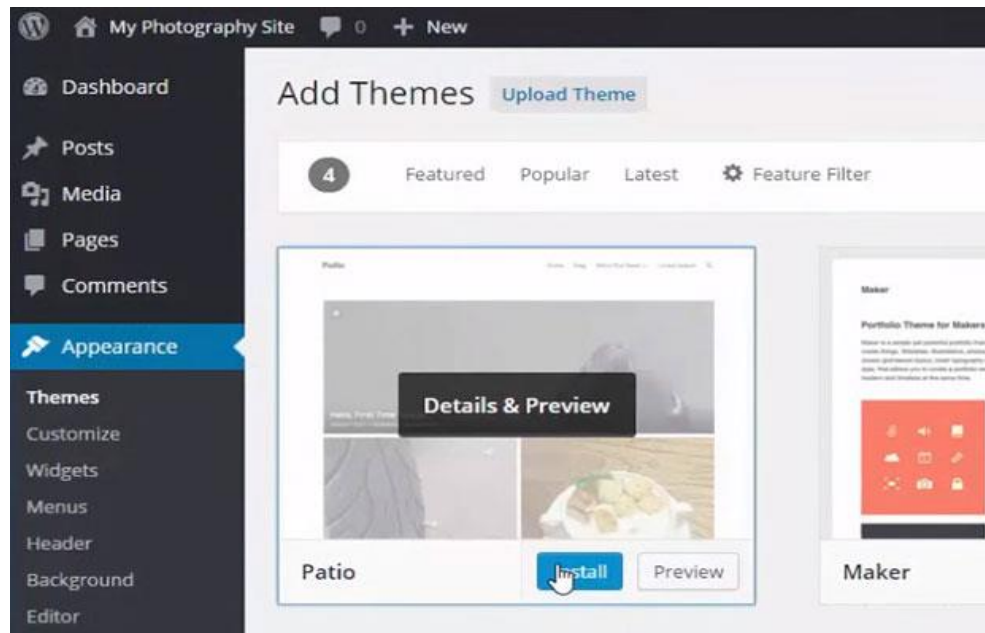
## 3. You can install a WordPress theme in your admin area by going to **Appearance » Themes** and clicking on the **Add New** button at the top.



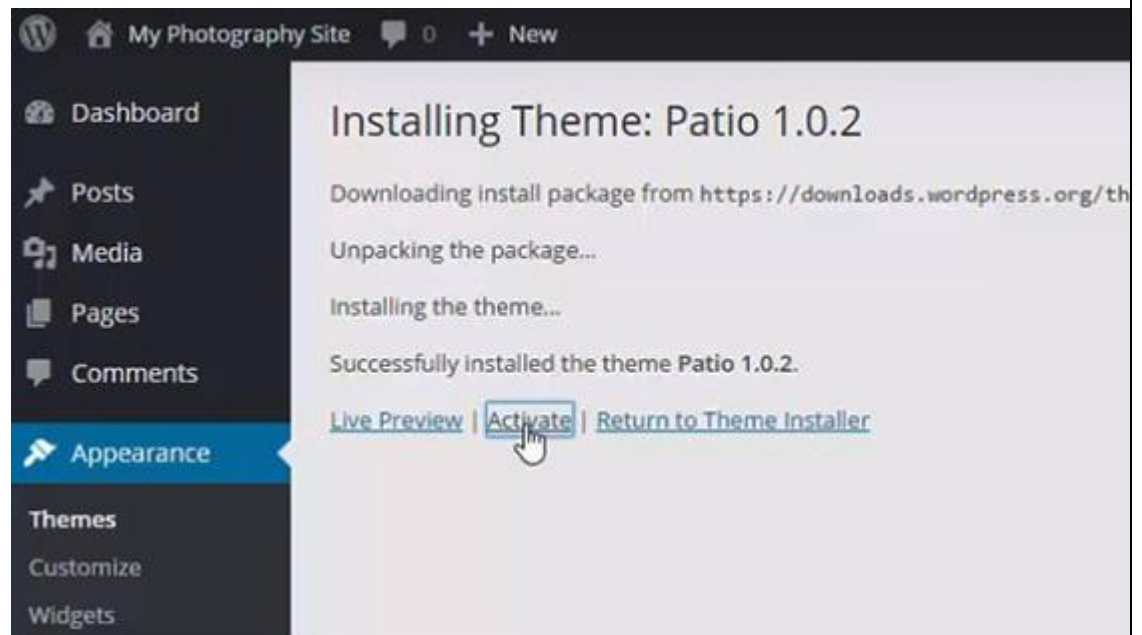
4. Write 'photography' in the search bar and hit the enter button.



5. We are selecting the *Patio* photography theme. It is a beautiful WordPress theme for photographers.

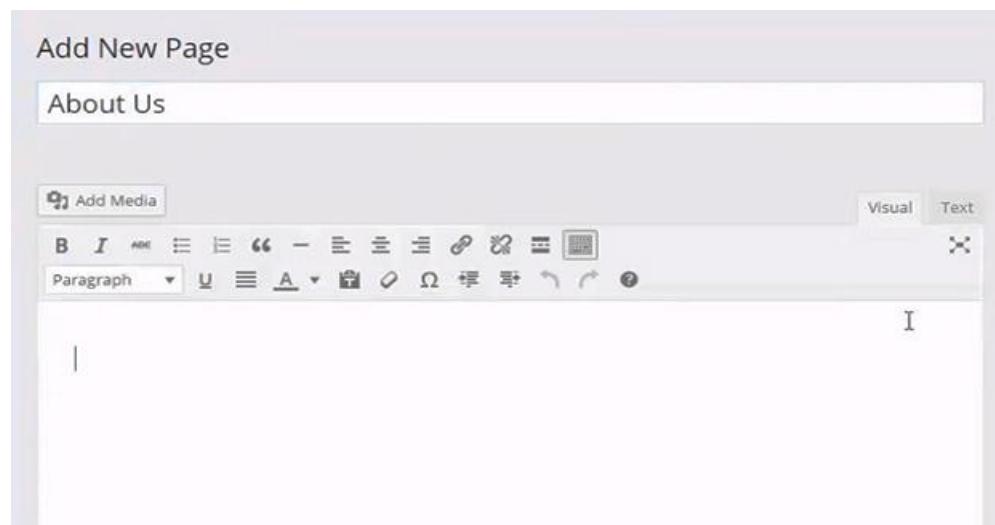


6. You need to click on the Install button from *Patio* theme and then click on the Activate link from the next screen.

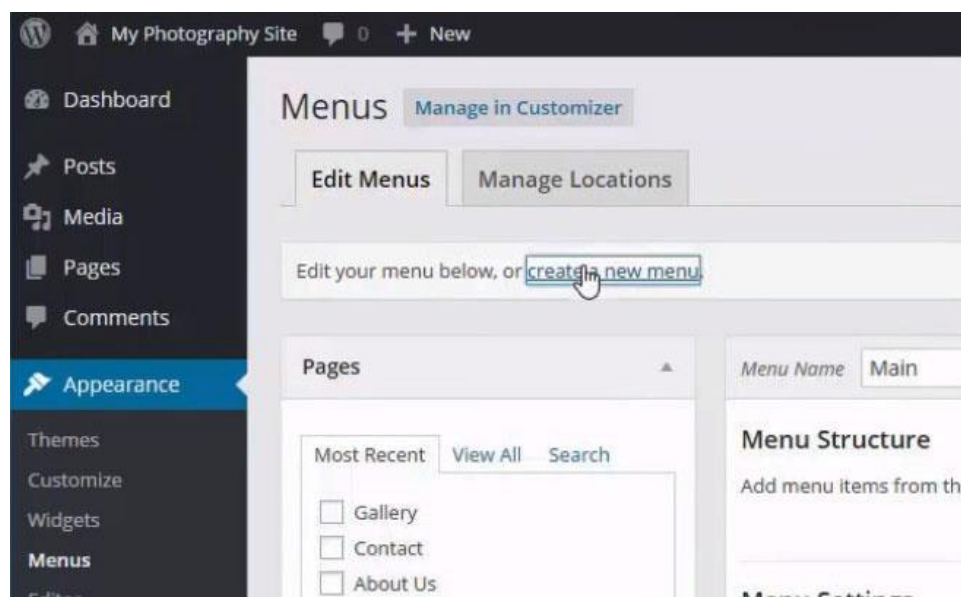


Your photography theme is now installed and activated in WordPress.

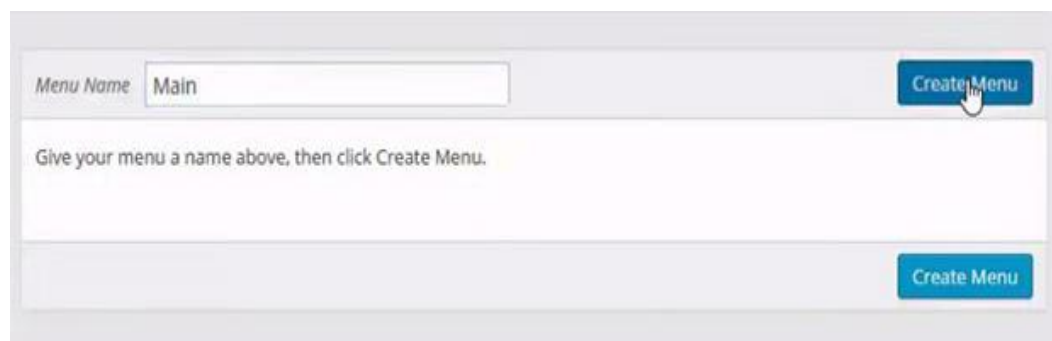
7. Add Pages and Posts in Your Website After activating the theme, you can go ahead to Pages » Add New to add your website pages one by one.



8. Once you have created a few pages, simply go to Appearance » Menus to create the menu for your website and click on the 'create a new menu' link.



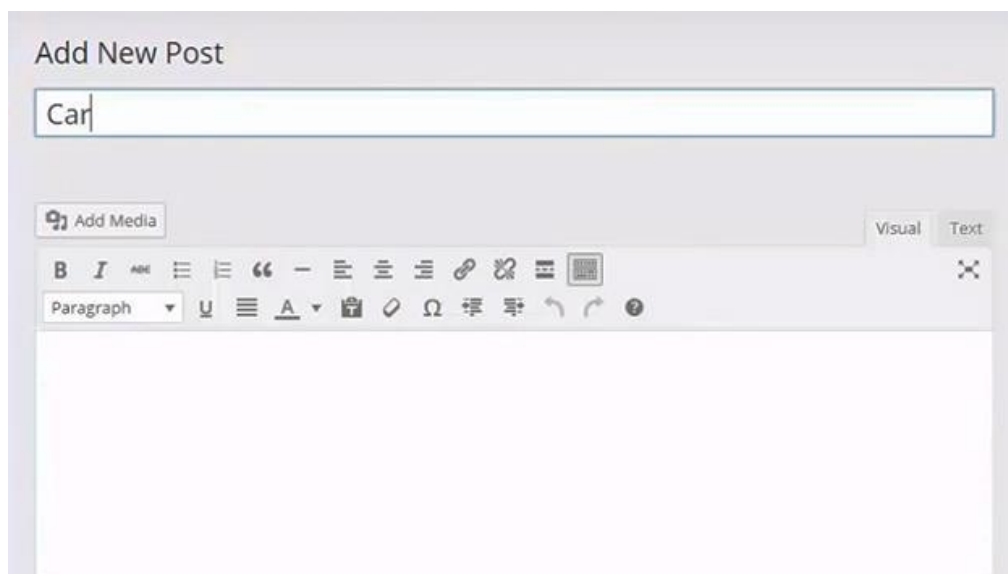
9. You will need to give a name to your menu and click on the Create Menu button.



10. You can see how the menu will appear on the homepage of your photography website in this example below:



**11. Once the pages are added, you can go to Posts » Add New to add a new blog post in WordPress.**



**12. You can edit these posts to add content and images for display on your photography website.**

**13. With all these and many more plug ins you can develop your website easily using wordpress.**

## ■ Website Deployment

When the process of website development is complete, its needs to be deployed on a web server. When development of website is completed, it is tested for its final working and ready to be hosted on a live web server. The hosting of a website on a web server involves the following steps:

- Domain registration
- Domain hosting
- parking websites
- uploading data using FTP
- email configuration.

### **Conclusion**

The various features of CMS have been studied and Wordpress have been used to develop a mini project.

### **Lab. Based FAQ**

1. What is CMS? What are its advantages?
2. What are various CMS tools?
3. How Wordpress is used for creation of interactive website?
4. What are the steps to deploy a website on web server?
5. Explain concept of Parking a website.