

WAD LAB MANUAL

TABLE OF CONTENTS

S.NO	NAME
1.	Implement Basic Html Tags
2.	Implement Table Tags
	i. Implement Frames
3.	Design a Form in Html
	i. Validation of Form Using JavaScript
4.	Implement Various Types of CSS
5.	Display Various Forms of XML Document
	i. Raw XML ii. XML with CSS iii .XML using XSLT
6.	Using JQuery Implement : i. Selecting Element, Getting Values, Setting Values.
	ii. Events
7.	Using Angular JS Implement : i. Input Validation
	ii. Backend Building
8.	Case Study on i)Mean Stack ii) Smack Stack

WEB APPLICATION DEVELOPMENT:

LAB MANUAL

1) HTML:

HTML stands for Hyper Text Markup Language, which is the most widely used language on Web to develop web pages. HTML was created by Berners-Lee in late 1991 but "HTML 2.0" was the first standard HTML specification which was published in 1995. HTML 4.01 was a major version of HTML and it was published in late 1999. Though HTML 4.01 version is widely used but currently we are having HTML-5 version which is an extension to HTML 4.01, and this version was published in 2012.

HTML stands for Hypertext Markup Language, and it is the most widely used language to write Web Pages.

- **Hypertext** refers to the way in which Web pages (HTML documents) are linked together. Thus, the link available on a webpage is called Hypertext.
- As its name suggests, HTML is a **Markup Language** which means you use HTML to simply "mark-up" a text document with tags that tell a Web browser how to structure it to display.

Originally, HTML was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers.

Now, HTML is being widely used to format web pages with the help of different tags available in HTML language.

1) BASIC HTML TAGS :

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.

Paragraph Tag:

The `<p>` tag offers a way to structure your text into different paragraphs. Each paragraph of text should go in between an opening `<p>` and a closing `</p>` tag.

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. If you omit this space, older browsers will have trouble rendering the line break, while if you miss the forward slash character and just use `
` it is not valid in XHTML.

Centering Content:

You can use `<center>` tag to put any content in the center of the page or any table cell.

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.

Source Code:

```
<!DOCTYPE html>
<html>
<head>
<title> Basic Tags - DCET</title>
</head>
<body>
<center><h1>Welcome to DCET</h1></center>
<p>Welcome to Deccan College of Engineering and Technology. Our
official website is <a
href="http://deccancollege.ac.in">deccancollege.ac.in</a></p>
<p>The college is affiliated to Osmania University, Hyderabad.</p>
<h2>We have many departments in Engineering</h2>
<ol>
<li>IT</li>
<li>CSE</li>
<li>ECE</li>
<li>EEE</li>
<li>IE</li>
<li>PE</li>
<li>Civil</li>
<li>Mechanical</li>
</ol>

<h3>Here's a Picture of our college</h3>



<footer>Copyrights 2019. All rights reserved.</footer>

</body>

</html>
```

2) Table Tags :

The HTML tables allow web authors to arrange data like text, images, links, other tables, etc. into rows and columns of cells.

The HTML tables are created using the **<table>** tag in which the **<tr>** tag is used to create table rows and **<td>** tag is used to create data cells. The elements under **<td>** are regular and left aligned by default.

Table Heading:

Table heading can be defined using **<th>** tag. This tag will be put to replace **<td>** tag, which is used to represent actual data cell. Normally you will put your top row as table heading as shown below, otherwise you can use **<th>** element in any row. Headings, which are defined in **<th>** tag are centered and bold by default.

Cellpadding and Cellspacing Attributes:

There are two attributes called cellpadding and cellspacing which you will use to adjust the white space in your table cells. The cellspacing attribute defines space between table cells, while cellpadding represents the distance between cell borders and the content within a cell.

Colspan and Rowspan Attributes:

You will use colspan attribute if you want to merge two or more columns into a single column. Similar way you will use rowspan if you want to merge two or more rows.

Tables Backgrounds:

You can set table background using one of the following two ways –

- bgcolor attribute – You can set background color for whole table or just for one cell.
- background attribute – You can set background image for whole table or just for one cell.

Table Height and Width:

You can set a table width and height using **width** and **height** attributes. You can specify table width or height in terms of pixels or in terms of percentage of available screen area.

Table Caption:

The caption tag will serve as a title or explanation for the table and it shows up at the top of the table. This tag is deprecated in newer version of HTML/XHTML.

Table Header, Body, and Footer:

Tables can be divided into three portions – a header, a body, and a foot. The head and foot are rather similar to headers and footers in a word-processed document that remain the same for every page, while the body is the main content holder of the table.

The three elements for separating the head, body, and foot of a table are –

- <thead> – to create a separate table header.
- <tbody> – to indicate the main body of the table.
- <tfoot> – to create a separate table footer.

A table may contain several <tbody> elements to indicate different pages or groups of data. But it is notable that <thead> and <tfoot> tags should appear before <tbody>

Source Code:

```
<html>
<head>
  <title>Table Tags</title>
</head>
<body>
  <table border ="4" cellspacing="2" cellpadding="2"
width="40%">
<tr>
  <td>Full name</td>
  <td>Syed Azhar Farhan
</td>

</tr>
<tr>
  <td>ID No.</td>
  <td>160316737030
</td>

</tr>
<tr>
  <td>Year</td>
  <td>3
</td>

</tr>
<tr>
  <td>Semester</td>
  <td>6
</td>

</tr>
<tr>
  <td>Branch</td>
  <td>IT
</td>

</tr>

<tr>
  <td>College</td>
  <td>DCET
</td>

</tr>

</table>
</body>
```

3) Frames in HTML :

HTML frames are used to divide your browser window into multiple sections where each section can load a separate HTML document. A collection of frames in the browser window is known as a frameset. The window is divided into frames in a similar way the tables are organized: into rows and columns.

Disadvantages of Frames:

There are few drawbacks with using frames, so it's never recommended to use frames in your web pages –

- Some smaller devices cannot cope with frames often because their screen is not big enough to be divided up.
- Sometimes your page will be displayed differently on different computers due to different screen resolution.
- The browser's back button might not work as the user hopes.
- There are still few browsers that do not support frame technology.

Creating Frames:

To use frames on a page we use `<frameset>` tag instead of `<body>` tag. The `<frameset>` tag defines how to divide the window into frames. The rows attribute of `<frameset>` tag defines horizontal frames and cols attribute defines vertical frames. Each frame is indicated by `<frame>` tag and it defines which HTML document shall open into the frame.

The `<frameset>` Tag Attributes

Following are important attributes of the `<frameset>` tag –

Sr.No	Attribute & Description
1	cols Specifies how many columns are contained in the frameset and the size of each column. You can specify the width of each column in one of the four ways – Absolute values in pixels. For example, to create three vertical frames, use <code>cols = "100, 500, 100"</code> . A percentage of the browser window. For example, to create three vertical frames, use <code>cols = "10%, 80%, 10%"</code> . Using a wildcard symbol. For example, to create three vertical frames, use <code>cols = "10%, *, 10%"</code> . In this case wildcard takes remainder of the window. As relative widths of the browser window. For example, to create three vertical frames, use <code>cols = "3*, 2*, 1*"</code> . This is an alternative to percentages. You can use relative widths of the browser window. Here the window is divided into sixths: the first column takes up half of the window, the second takes one third, and the third takes one sixth.
2	rows This attribute works just like the cols attribute and takes the same values, but it is used to specify the rows in the frameset. For example, to create two horizontal frames, use <code>rows = "10%, 90%"</code> . You can specify the height of each row in the same way as

	explained above for columns.
3	border This attribute specifies the width of the border of each frame in pixels. For example, border = "5". A value of zero means no border.
4	frameborder This attribute specifies whether a three-dimensional border should be displayed between frames. This attribute takes value either 1 (yes) or 0 (no). For example frameborder = "0" specifies no border.
5	framespacing This attribute specifies the amount of space between frames in a frameset. This can take any integer value. For example framespacing = "10" means there should be 10 pixels spacing between each frames.

Source Code:

Frame A:

```
<!DOCTYPE html>
<html>
<head>
    <title>frame_a.html</title>
</head>
<body>
<h1>This is frame a.</h1>
having 25% of the total division
</body>
</html>
```

Frame B:

```
<!DOCTYPE html>
<html>
<head>
    <title>frame_b.html</title>
</head>
<body>
<h1>This is frame b.</h1>
having 50% of the total division
</body>
</html>
```

Frame C:

```
<!DOCTYPE html>
<html>
<head>
    <title>frame_c.html</title>
</head>
<body>
<h1>This is frame c.</h1>
having 25% of the total division
```



```
</body>
</html>
Frame :
<html>
<frameset cols="25%,*,25%">
  <frame src="frame_a.html">
  <frame src="frame_b.html">
  <frame src="frame_c.html">
</frameset>
</html>
```

4) Form Validation Using Java Script :

JavaScript is a lightweight, interpreted programming language. It is designed for creating network-centric applications. It is complimentary to and integrated with Java. JavaScript is very easy to implement because it is integrated with HTML. It is open and cross-platform.

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

JavaScript was first known as LiveScript, but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first appearance in Netscape 2.0 in 1995 with the name LiveScript. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers.

The ECMA-262 Specification defined a standard version of the core JavaScript language.

- JavaScript is a lightweight, interpreted programming language.
- Designed for creating network-centric applications.
- Complementary to and integrated with Java.
- Complementary to and integrated with HTML.
- Open and cross-platform
- Client-Side JavaScript
- Client-side JavaScript is the most common form of the language. The script should be included in or referenced by an HTML document for the code to be interpreted by the browser.

It means that a web page need not be a static HTML, but can include programs that interact with the user, control the browser, and dynamically create HTML content.

The JavaScript client-side mechanism provides many advantages over traditional CGI server-side scripts. For example, you might use JavaScript to check if the user has entered a valid e-mail address in a form field.

The JavaScript code is executed when the user submits the form, and only if all the entries are valid, they would be submitted to the Web Server.

JavaScript can be used to trap user-initiated events such as button clicks, link navigation, and other actions that the user initiates explicitly or implicitly.

Advantages of JavaScript –

- Less server interaction – You can validate user input before sending the page off to the server. This saves server traffic, which means less load on your server.
- Immediate feedback to the visitors – They don't have to wait for a page reload to see if they have forgotten to enter something.
- Increased interactivity – You can create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.
- Richer interfaces – You can use JavaScript to include such items as drag-and-drop components and sliders to give a Rich Interface to your site visitors.

Limitations of JavaScript –

We cannot treat JavaScript as a full-fledged programming language. It lacks the following important features –

- Client-side JavaScript does not allow the reading or writing of files. This has been kept for security reason.
- JavaScript cannot be used for networking applications because there is no such support available.
- JavaScript doesn't have any multi-threading or multiprocessor capabilities.

- Once again, JavaScript is a lightweight, interpreted programming language that allows you to build interactivity into otherwise static HTML pages.

Source Code:

```
<html>
<head>
  <title>Form Validation</title>
  <script type="text/javascript">
    function validate(){

      var fname=document.getElementById("name");

      var username=document.getElementById("username");
      var password=document.getElementById("password1");
      var repassword=document.getElementById("password2");
      var address=document.getElementById("address");
      var age=document.getElementById("age");
      var gender=document.getElementById("gender");

      if(fname.value==" "||username.value==" "||address.value==" "||
      age.value==" "|| password.value==" " || repassword.value==""){
        alert("fill all the details");
        return false;
      }
      else{
        return true;
      }
    }
  </script>
</head>
<body>
  <form onsubmit="return validate()" action="xyz.html"
  method="POST">
  <center><h3>Registration form</h3>
    <table border ="4" bordercolor="#7FFFD4" cellpadding="2"
    cellspacing="2" width="40%">
    <tr>
      <td>Full name</td>
      <td>:</td>
      <td><input type="text" placeholder="full name"
      id="name">
      </td>
    </tr>
    <tr>
      <td>User name</td>
      <td>:</td>
      <td><input type="text" placeholder="user name"
      id="username">
      </td>
    </tr>
    <tr>
      <td>Password</td>
      <td>:</td>
```

```

        <td><input type="password" placeholder="Password"
id="password1">
        </td>
</tr>
<tr>
        <td>Re Password</td>
        <td>:</td>
        <td><input type="Password" placeholder="password"
id="password2">
        </td>
</tr>
<tr>
        <td>Address</td>
        <td>:</td>
        <td><input type="textarea" placeholder="address"
id="address">
        </td>
</tr>
<tr>
        <td>Age</td>
        <td>:</td>
        <td><input type="number" label="age" id="age">
        </td>
</tr>
<tr>
        <td>Gender</td>
        <td>:</td>
        <td style="width: 8%"><input type="radio"
label="male" name="Gender" value="1">male
        </td>
        <td><input type="radio" label="female" name="Gender"
value="0">female
        </td>
</tr>
</table>
<br><br>
<input type="submit" name="submit">
</center>
</form>
</body>

```

5) Cascading Style Sheets (CSS):

CSS is used to control the style of a web document in a simple and easy way.

CSS is the acronym for "Cascading Style Sheet". This tutorial covers both the versions CSS1, CSS2 and CSS3, and gives a complete understanding of CSS, starting from its basics to advanced concepts.

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects.

CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

Advantages of CSS:

- CSS saves time – You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.
- Pages load faster – If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times.
- Easy maintenance – To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
- Superior styles to HTML – CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.
- Multiple Device Compatibility – Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.
- Global web standards – Now HTML attributes are being deprecated and it is being recommended to use CSS. So its a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.

Inline CSS:

Inline styles look and operate much like CSS, with a few differences. Inline styles directly affect the tag they are written in, without the use of selectors.

Here's a basic HTML page using inline styles:

```
<!DOCTYPE html>
<html>
<head>
  <title>demo</title>
</head>
<body style="background-color: grey"><center>
  <h1 style="background-color: white"> Implementation of inline
css</h1></center>
<p style="background-color: aqua; width: 70%;"> A paragraph
with<br> <br>background-color : aqua<br><br>color : black<br><br>
having width : 70%</p>
```

```


<p style="background-color: aqua; "> An image with <br><br> opacity
: 0.5 <br><br> width : 25% and <br><br> height : 25%</p>
</body>
</html>

```

Inpage CSS :

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.

```

<!DOCTYPE html>
<html>
<head>
    <style type="text/css">
        h1{
            background-color: white;
        }
        body{
            background-color: grey;
        }
        .a{
            background-color: aqua;
            width: 70%;
        }
        #p1{
            background-color: aqua;
            width: 30%;
        }
        img{
            height: 25%;
            width: 25%;
            opacity: 0.5;
        }
    </style>
    <title>inPage</title>
</head>
<body><center>
    <h1> Implementation of inPage css</h1></center>
    <p class="a"> A paragraph with<br> <br>background-color :
aqua<br><br> color : black <br><br> having width : 70%</p>

    
    <p id="p1"> An image with <br><br> opacity : 0.7<br><br> width : 25%
and <br><br> height : 25%</p>
</body>
</html>

```

External CSS:

When using an external stylesheet you must reference the stylesheet in the HTML page that is using it. You would add the code below to your HTML document to reference a stylesheet in the same location as the HTML page called "style.css". You can upload the "style.css" page can be located anywhere in your files. You can name your stylesheet whatever you like and link to as many as you like. You can simply link to it in your head section and every edit you make to the "style.css" sheet will be globally changed through out the site. Below is what the code looks like.

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

```
<!DOCTYPE html>
<html>
<head>
    <link rel="stylesheet" type="text/css" href="mycss.css">
    <title>external</title>
</head>
<body><center>
    <h1> Implementation of External css</h1></center>
<p class="a"> A paragraph with<br> <br>background-color :
aqua<br><br> color : black <br><br> having width : 70%</p>

<p id="p1"> An image with <br><br> opacity : 0.5<br><br> width : 25%
and <br><br> height : 25%</p>
</body>
</html>
```

```
h1{
    background-color: white;
}
body{
    background-color: grey;
}
.a{
background-color: aqua;
width: 70%;
}
#p1{
background-color: aqua;
width: 30%;
}
img{
    height: 25%;
    width: 25%;
    opacity: 0.5;
}
```

6) XML :

XML stands for Extensible Markup Language. It is a text-based markup language derived from Standard Generalized Markup Language (SGML).

XML tags identify the data and are used to store and organize the data, rather than specifying how to display it like HTML tags, which are used to display the data. XML is not going to replace HTML in the near future, but it introduces new possibilities by adopting many successful features of HTML.

There are three important characteristics of XML that make it useful in a variety of systems and solutions –

- XML is extensible – XML allows you to create your own self-descriptive tags, or language, that suits your application.
- XML carries the data, does not present it – XML allows you to store the data irrespective of how it will be presented.
- XML is a public standard – XML was developed by an organization called the World Wide Web Consortium (W3C) and is available as an open standard.

XML Usage

A short list of XML usage says it all –

- XML can work behind the scene to simplify the creation of HTML documents for large web sites.
- XML can be used to exchange the information between organizations and systems.
- XML can be used for offloading and reloading of databases.
- XML can be used to store and arrange the data, which can customize your data handling needs.
- XML can easily be merged with style sheets to create almost any desired output.
- Virtually, any type of data can be expressed as an XML document.

What is Markup?

XML is a markup language that defines set of rules for encoding documents in a format that is both human-readable and machine-readable. So what exactly is a markup language? Markup is information added to a document that enhances its meaning in certain ways, in that it identifies the parts and how they relate to each other. More specifically, a markup language is a set of symbols that can be placed in the text of a document to demarcate and label the parts of that document.

Following example shows how XML markup looks, when embedded in a piece of text –

```
<message>
  <text>Hello, world!</text>
</message>
```

This snippet includes the markup symbols, or the tags such as <message>...</message> and <text>...</text>. The tags <message> and </message> mark the start and the end of the XML code fragment. The tags <text> and </text> surround the text Hello, world!.

Is XML a Programming Language?

A programming language consists of grammar rules and its own vocabulary which is used to create computer programs. These programs instruct the computer to perform specific tasks. XML does not qualify to be a programming language as it does not perform any computation or algorithms. It is usually stored in a simple text file and is processed by special software that is capable of interpreting XML.

Raw-XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<DCET>
  <DETAILS>
    <NAME>Syed Azhar Farhan</NAME>
    <ID>160316737030</ID>
    <BRANCH>IT</BRANCH>
    <YEAR>3rd</YEAR>
    <SEMESTER>6th</SEMESTER>
    <ADDRESS>Hyderabad</ADDRESS>
  </DETAILS>
  <DETAILS>
    <NAME>Farhan Ahmed Jaweed</NAME>
    <ID>160316737051</ID>
    <BRANCH>IT</BRANCH>
    <YEAR>3rd</YEAR>
    <SEMESTER>6th</SEMESTER>
    <ADDRESS>Hyderabad</ADDRESS>
  </DETAILS>
  <DETAILS>
    <NAME>Mohd Sami Uddin</NAME>
    <ID>160316737054</ID>
    <BRANCH>IT</BRANCH>
    <YEAR>3rd</YEAR>
    <SEMESTER>6th</SEMESTER>
    <ADDRESS>Hyderabad</ADDRESS>
  </DETAILS>
</DCET>
```

XML with CSS:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/css" href="mycss.css"?>
<DCET>
  <DETAILS>
    <NAME>Syed Azhar Farhan</NAME>
    <ID>160316737030</ID>
    <BRANCH>IT</BRANCH>
    <YEAR>3rd</YEAR>
    <SEMESTER>6th</SEMESTER>
    <ADDRESS>Hyderabad</ADDRESS>
  </DETAILS>
  <DETAILS>
    <NAME>Farhan Ahmed Jaweed</NAME>
    <ID>160316737051</ID>
    <BRANCH>IT</BRANCH>
    <YEAR>3rd</YEAR>
    <SEMESTER>6th</SEMESTER>
    <ADDRESS>Hyderabad</ADDRESS>
  </DETAILS>
  <DETAILS>
    <NAME>Mohd Sami Uddin</NAME>
```

```
<ID>160316737054</ID>
<BRANCH>IT</BRANCH>
<YEAR>3rd</YEAR>
<SEMESTER>6th</SEMESTER>
<ADDRESS>Hyderabad</ADDRESS>
</DETAILS>
</DCET>
```

XML using XSLT:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="xslt.xsl"?>
<DCET>
  <DETAILS>
    <NAME>Syed Azhar Farhan</NAME>
    <ID>160316737030</ID>
    <BRANCH>IT</BRANCH>
    <YEAR>3rd</YEAR>
    <SEMESTER>6th</SEMESTER>
    <ADDRESS>Hyderabad</ADDRESS>
  </DETAILS>
  <DETAILS>
    <NAME>Farhan Ahmed Jaweed</NAME>
    <ID>160316737051</ID>
    <BRANCH>IT</BRANCH>
    <YEAR>3rd</YEAR>
    <SEMESTER>6th</SEMESTER>
    <ADDRESS>Hyderabad</ADDRESS>
  </DETAILS>
  <DETAILS>
    <NAME>Mohd Sami Uddin</NAME>
    <ID>160316737054</ID>
    <BRANCH>IT</BRANCH>
    <YEAR>3rd</YEAR>
    <SEMESTER>6th</SEMESTER>
    <ADDRESS>Hyderabad</ADDRESS>
  </DETAILS>
</DCET>
```

Xslt.xsl

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
<body>
  <h2>DCET</h2>
  <table border="1">
    <tr bgcolor="#9acd32">
      <th style="text-align:left">Name</th>
      <th style="text-align:left">ID</th>
    </tr>
    <xsl:for-each select="DCET/DETAILS">
```

```
<tr>
  <td><xsl:value-of select="NAME"/></td>
  <td><xsl:value-of select="ID"/></td>
</tr>
</xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

7) jQuery:

jQuery is a fast and concise JavaScript library created by John Resig in 2006. jQuery simplifies HTML document traversing, event handling, animating, and Ajax interactions for Rapid Web Development. jQuery is a fast and concise JavaScript Library created by John Resig in 2006 with a nice motto: Write less, do more. jQuery simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development. jQuery is a JavaScript toolkit designed to simplify various tasks by writing less code. Here is the list of important core features supported by jQuery –

- DOM manipulation – The jQuery made it easy to select DOM elements, negotiate them and modifying their content by using cross-browser open source selector engine called Sizzle.
- Event handling – The jQuery offers an elegant way to capture a wide variety of events, such as a user clicking on a link, without the need to clutter the HTML code itself with event handlers.
- AJAX Support – The jQuery helps you a lot to develop a responsive and featurerich site using AJAX technology.
- Animations – The jQuery comes with plenty of built-in animation effects which you can use in your websites.
- Lightweight – The jQuery is very lightweight library - about 19KB in size (Minified and gzipped).
- Cross Browser Support – The jQuery has cross-browser support, and works well in IE 6.0+, FF 2.0+, Safari 3.0+, Chrome and Opera 9.0+
- Latest Technology – The jQuery supports CSS3 selectors and basic XPath syntax.

How to use jQuery?

There are two ways to use jQuery.

- Local Installation – You can download jQuery library on your local machine and include it in your HTML code.
- CDN Based Version – You can include jQuery library into your HTML code directly from Content Delivery Network (CDN).

Local Installation

Go to the <https://jquery.com/download/> to download the latest version available.

Now put downloaded jquery-2.1.3.min.js file in a directory of your website, e.g. /jquery.

Event1.html

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("h1").hide();
    });
});
</script>

<style type="text/css">
    .butn{
        padding: 10px;
        background: aqua;
        border: solid black;
    }
</style>
</head>
<body>
    <center>
<h1>Click the button to hide this text</h1>

<button class="butn">Click on this</button>
</center>

</body>
</html>
```

Event2.html

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").mouseenter(function(){
        $("h1").hide();
    });
    $("button").mouseleave(function(){
        $("h1").show();
    });
});
</script>
```

```
<style type="text/css">
    .butn{
        margin-top: 200px;
        padding: 10px;
        background: aqua;
        border: solid black;
    }
</style>
</head>
<body>
    <center>
<button class="butn">Hover me</button>
<h1>Hover the button to hide this text</h1>
</center>

</body>
</html>
```

Event3.html

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("#f").fadeOut(1000);
    $("#s").fadeOut(1000);
    $("#s").fadeIn(1000);
});
</script>
</head>
<body>

<p id="f">This will vanish</p>
<p id="s">This will fade and come </p>

</body>
</html>
```

Getting Values :

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script>
$(document).ready(function() {
    $("#btn1").click(function() {
        $("#test1").text($("#test").val());
    });
});
</script>
</head>
<body>
<input type="text" id="test"></input>
<br><br>
<button id="btn1">Show Text</button>

<h1>The entered value is :</h1>
<h2 id="test1"></h2>
</body>
</html>
```

Setting Values:

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script>
$(document).ready(function() {
    $("#btn1").click(function() {
        $("#test1").text("Hi!");
    });
    $("#btn2").click(function() {
        $("#test2").html("<b>Changed</b>");
    });
    $("#btn3").click(function() {
        $("#test3").val("Goodbye");
    });
});
</script>
</head>
<body>
<button id="btn1">Hello World!</button>
<p id="test1">Magic1</p>
<button id="btn2">This will change</button>
<p id="test2">Magic2</p>
```

```
<button id="btn3">Bye</button>
<p>Input field: <input type="text" id="test3" value="Magic3"></p>

</body>
</html>
```


8) Angular JS :

AngularJS is an open-source web application framework. It was originally developed in 2009 by Misko Hevery and Adam Abrons. It is now maintained by Google. Its latest version is 1.2.21.

Definition of AngularJS as put by its official documentation is as follows –

AngularJS is a structural framework for dynamic web applications. It lets you use HTML as your template language and lets you extend HTML's syntax to express your application components clearly and succinctly. Its data binding and dependency injection eliminate much of the code you currently have to write. And it all happens within the browser, making it an ideal partner with any server technology.

General Features

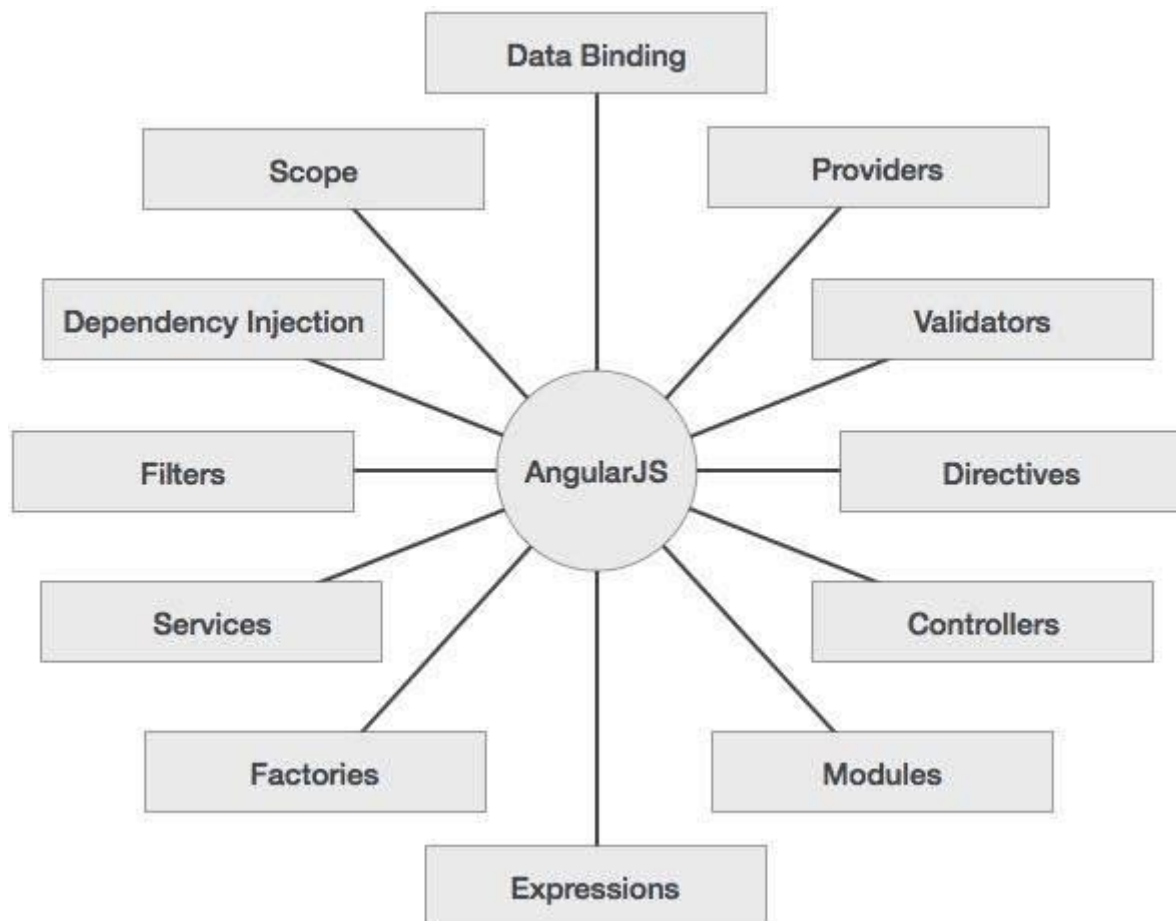
The general features of AngularJS are as follows –

- AngularJS is a efficient framework that can create Rich Internet Applications (RIA).
- AngularJS provides developers an options to write client side applications using JavaScript in a clean Model View Controller (MVC) way.
- Applications written in AngularJS are cross-browser compliant. AngularJS automatically handles JavaScript code suitable for each browser.
- AngularJS is open source, completely free, and used by thousands of developers around the world. It is licensed under the Apache license version 2.0.
- Overall, AngularJS is a framework to build large scale, high-performance, and easy to-maintain web applications.

Core Features:

The core features of AngularJS are as follows –

- Data-binding – It is the automatic synchronization of data between model and view components.
- Scope – These are objects that refer to the model. They act as a glue between controller and view.
- Controller – These are JavaScript functions bound to a particular scope.
- Services – AngularJS comes with several built-in services such as \$http to make a XMLHttpRequests. These are singleton objects which are instantiated only once in app.
- Filters – These select a subset of items from an array and returns a new array.
- Directives – Directives are markers on DOM elements such as elements, attributes, css, and more. These can be used to create custom HTML tags that serve as new, custom widgets. AngularJS has built-in directives such as ngBind, ngModel, etc.
- Templates – These are the rendered view with information from the controller and model. These can be a single file (such as index.html) or multiple views in one page using partials.
- Routing – It is concept of switching views.
- Deep Linking – Deep linking allows to encode the state of application in the URL so that it can be bookmarked. The application can then be restored from the URL to the same state.
- Dependency Injection – AngularJS has a built-in dependency injection subsystem that helps the developer to create, understand, and test the applications easily.



Advantages of AngularJS:

The advantages of AngularJS are –

- It provides the capability to create Single Page Application in a very clean and maintainable way.
- It provides data binding capability to HTML. Thus, it gives user a rich and responsive experience.
- AngularJS code is unit testable.
- AngularJS uses dependency injection and make use of separation of concerns.
- AngularJS provides reusable components.
- With AngularJS, the developers can achieve more functionality with short code.
- In AngularJS, views are pure html pages, and controllers written in JavaScript do the business processing.
- On the top of everything, AngularJS applications can run on all major browsers and smart phones, including Android and iOS based phones/tablets.

Back-end Building using AngularJS :

```
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.m
in.js"></script>
<script>
var app=angular.module("myapp",[ ])
app.controller("myctrl",function($scope){
$scope.s=[{
rollno:"1",
```

```
name:"syed",
dept:"it",
address:"hyd"
},{
rollno:"2",
name:"mohd",
dept:"cse",
address:"chennai"
},{
rollno:"3",
name:"uddin",
dept:"mech",
address:"bangalore"
},{
rollno:"4",
name:"ahmed",
dept:"ece",
address:"mumbai"
},{
rollno:"5",
name:"ali",
dept:"civil",
address:"delhi"
}}
})
</script>
</head>
<body ng-app="myapp">
<div ng-controller="myctrl">
<table border="4">
<tr>
<td>rollno</td>
<td>name</td>
<td>dept</td>
<td>address</td>
</tr>
<tr ng-repeat="student in s">
<td>{{student.rollno}}</td>
<td>{{student.name}}</td>
<td>{{student.dept}}</td>
<td>{{student.address}}</td>
</tr>
</table>
</body>
</html>
```

Input Validation using AngularJS:

```
<!DOCTYPE html>
<html>
<script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.m
in.js"></script>
<body>

<h2>AngularJS Validation Example</h2>

<form ng-app="myApp" ng-controller="validateCtrl"
name="myForm" novalidate>

<p>Username:<br>
<input type="text" name="user" ng-model="user" required>
<span style="color:red" ng-show="myForm.user.$dirty &&
myForm.user.$invalid">
<span ng-show="myForm.user.$error.required">Username is
required.</span>
</span>
</p>

<p>Email:<br>
<input type="email" name="email" ng-model="email" required>
<span style="color:red" ng-show="myForm.email.$dirty &&
myForm.email.$invalid">
<span ng-show="myForm.email.$error.required">Email is
required.</span>
<span ng-show="myForm.email.$error.email">Invalid email
address.</span>
</span>
</p>

<p>
<input type="submit"
ng-disabled="myForm.user.$dirty && myForm.user.$invalid ||
myForm.email.$dirty && myForm.email.$invalid">
</p>

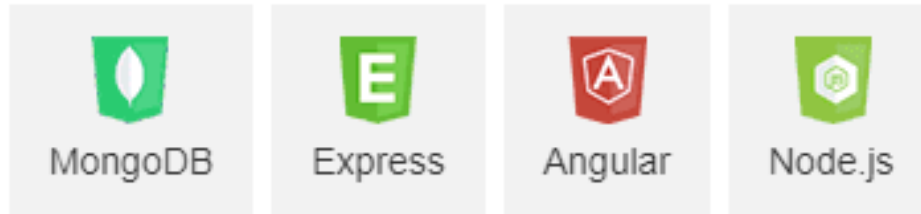
</form>

<script>
var app = angular.module('myApp', []);
app.controller('validateCtrl', function($scope) {
    $scope.user = 'Azhar Farhan';
    $scope.email = 'syedazharfarhan@gmail.com';
});
</script>

</body>
</html>
```

9) MEAN STACK

MEAN is a user-friendly full-stack JavaScript framework ideal for building dynamic websites and applications. It is a free and open-source stack designed to supply developers with a quick and organized method for creating rapid prototypes of MEAN-based web applications. One of the main benefits of the MEAN stack is that a single language, JavaScript, runs on every level of the application, making it an efficient and modern approach to web development.



[Node.js](#) is a server side JavaScript execution environment. It's a platform built on Google Chrome's V8 JavaScript runtime. It helps in building highly scalable and concurrent applications rapidly.

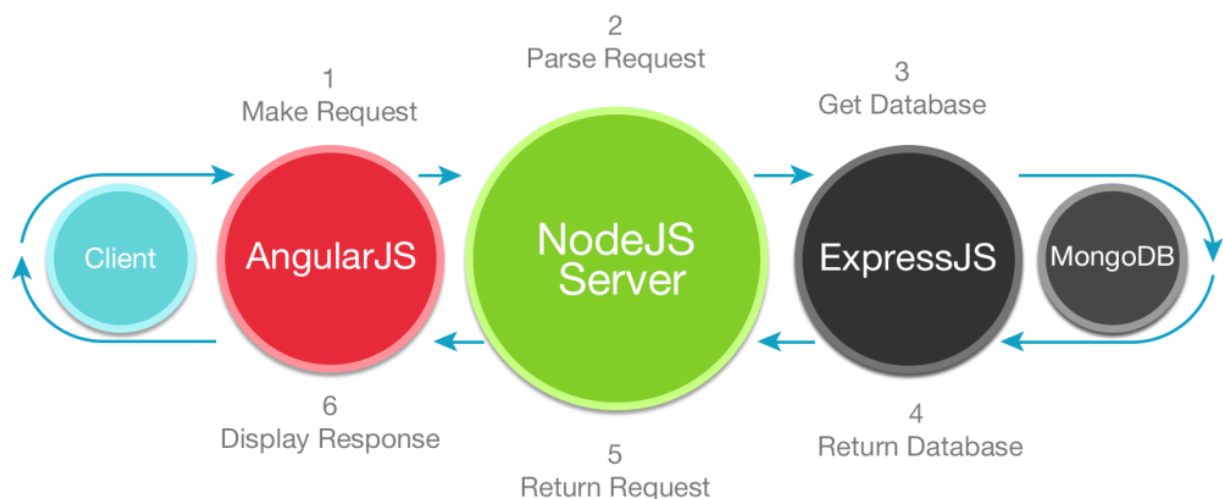
[Express](#) is lightweight framework used to build web applications in Node. It provides a number of robust features for building single and multi page web application. Express is inspired by the popular Ruby framework, Sinatra.

[MongoDB](#) is a schemaless NoSQL database system. MongoDB saves data in binary JSON format which makes it easier to pass data between client and server.

[AngularJS](#) is a JavaScript framework developed by Google. It provides some awesome features like the two-way data binding. It's a complete solution for rapid and awesome front end development.

WHY LEARN MEAN?

It's hard to accomplish much on the web without JavaScript, which is the single language that runs the entire MEAN full stack and boasts one of the most active developer communities. Because every part of MEAN programming is written in one language, it allows unique server-side and client-side execution environments. Valued for its versatility in building fast, robust and maintainable production web applications, MEAN is in high demand with numerous startups and employers.



FEATURES OF MEAN

EMPLOYABILITY

More and more employers are in need of engineers familiar with MEAN Stack and other JavaScript-based technologies.

SIMPLE & QUICK

Building websites and applications that revolve around one language, JavaScript, is relatively straightforward.

ADAPTABILITY

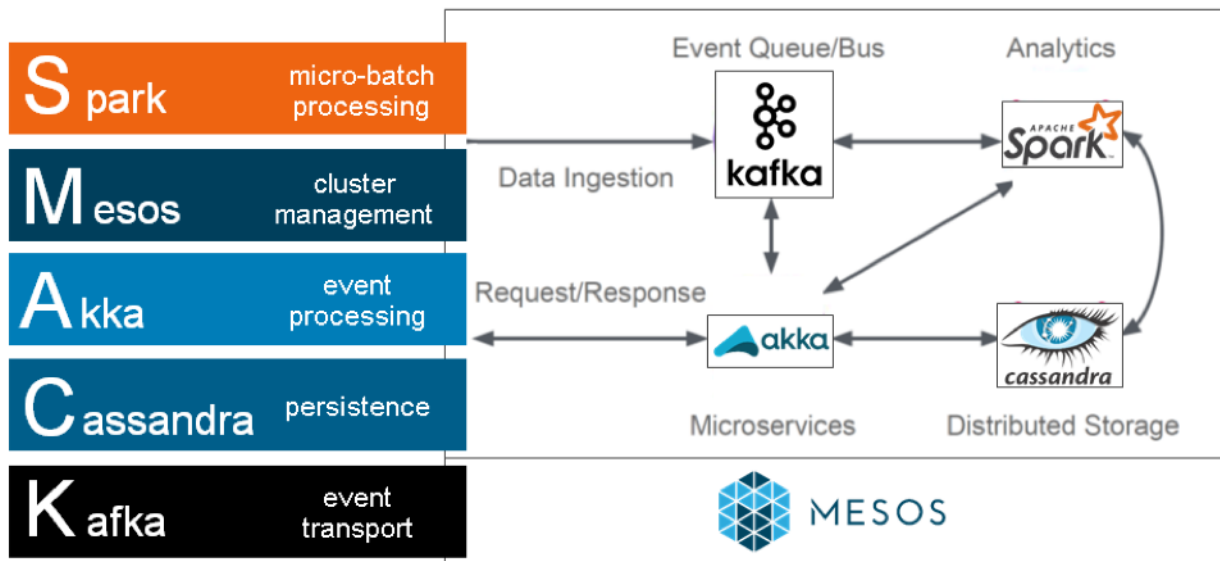
Due to the versatility of MEAN Stack's common programming language, JavaScript, it is highly adaptable for a wide range of web applications.

ACTIVE DEV COMMUNITY

MEAN Stack runs on JavaScript, the most common programming language in the world with one of the most active developer communities, making solutions to problems easily accessible.

SMACK (Spark, Mesos, Akka, and Kafka)

The SMACK Stack



APACHE SPARK :

The job of actually analyzing the data goes to Apache Spark. This fast and general-purpose big data processing engine enables you to combine SQL, streaming, and complex analytics. It also provides high-level APIs for Java, Scala, Python, and R, with an optimized general execution graphs engine.

APACHE MESOS :

Apache Mesos is SMACK's foundation. Mesos, a distributed systems kernel, abstracts CPU, memory, storage, and other computational resources away from physical or virtual machines. On Mesos, you build fault-tolerant and elastic distributed systems. Mesos runs applications within its cluster. It also provides a highly available platform. In the event of a system failure, Mesos relocates applications to different cluster nodes.

This Mesos kernel provides the SMACK applications (and other big data applications, such as Hadoop), with the APIs they need for resource management and scheduling across data center, cloud, and container platforms. While many SMACK implementations use Mesosphere's Mesos Data Center Operating System (DC/OS) distribution, SMACK works with any version of Mesos or, with some elbow grease, other distributed systems.

APACHE AKKA :

Next on the stack is Akka. Akka both brings data into a SMACK stack and sends it out to end-user applications.

The Akka toolkit aims to help developers build highly concurrent, distributed, and resilient message-driven applications for Java and Scala. It uses the actor model as its abstraction level to provide a platform to build scalable, resilient, and responsive applications.

The actor model is a conceptual model to work with concurrent computation. It defines general rules for how the system's components should behave and interact. The best-known language using this abstraction is Erlang.

With Akka, all interactions work in a distributed environment; its interactions actors use pure message-passing data in an asynchronous approach.

APACHE CASSANDRA :

In SMACK, data is kept in Apache Cassandra, a well-known distributed NoSQL database for managing large amounts of structured data across multiple servers, depended on for a lot of high-availability applications. Cassandra can handle huge quantities of data across multiple storage devices and vast numbers of concurrent users and operations per second.

APACHE KAFKA :

Apache Kafka is a distributed, partitioned, replicated commit log service. In SMACK, Kafka serves to provide messaging system functionality.

In a larger sense, Kafka decouples data pipelines and organizes data streams. With Kafka, data messages are byte arrays, which you can use to store objects in many formats, such as Apache Avro, JSON, and String. Kafka treats each set of data messages as a log—that is, an ordered set of messages. SMACK uses Kafka as a messaging system between its other programs.

Why SMACK:

- SMACK enables your company to quickly create big data analysis applications.
- Once built, those applications let you pull data speedily from your real-time data.
- And because SMACK is both flexible and makes efficient use of your server resources, you can do all the above with minimal hardware costs.

Building Data Analytics Infrastructure With SMACK Stack

