

## 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" cellspacing="2" cellpadding="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;
}
```



## 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 –

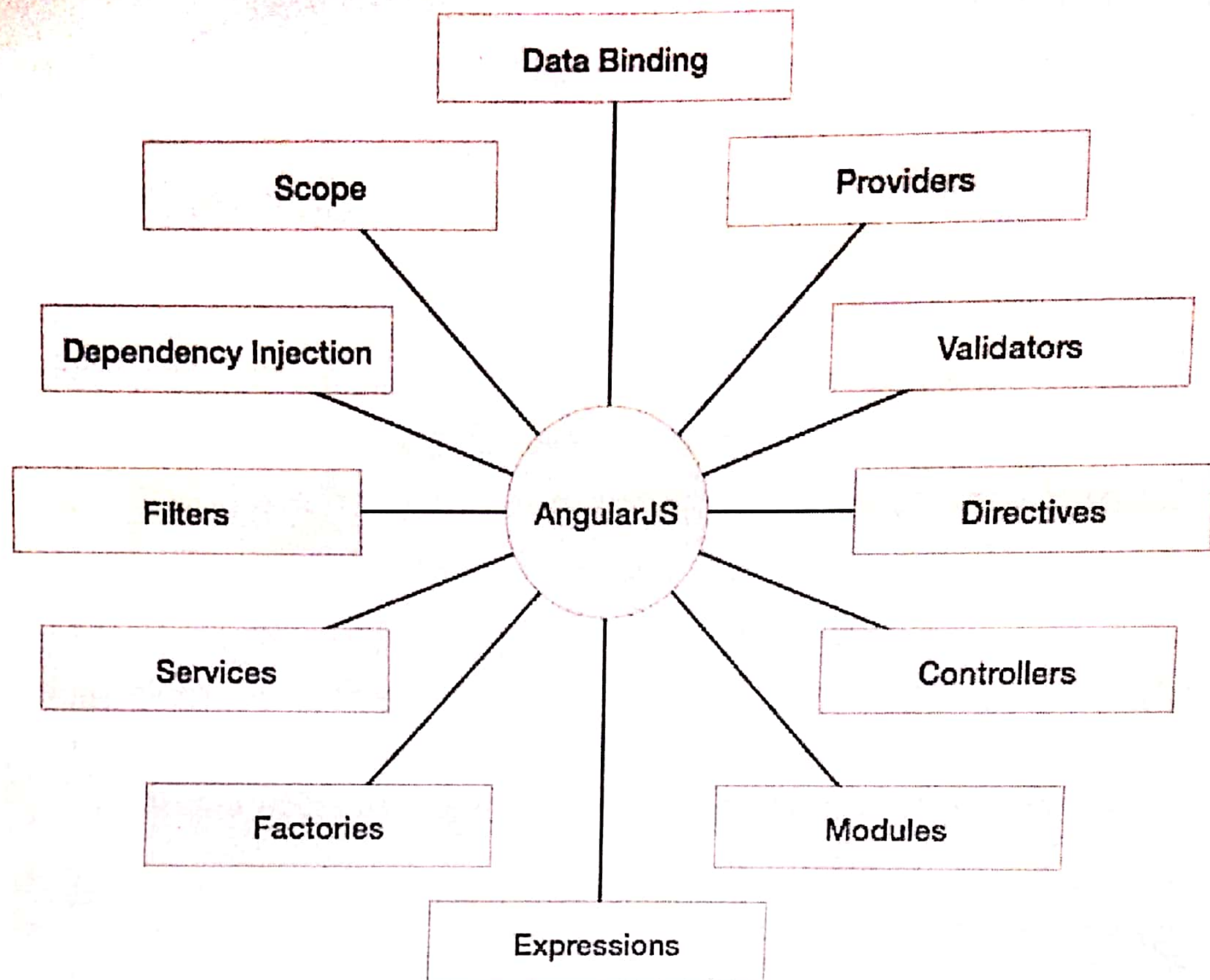
- AngularJS is a efficient framework that can create Rich Internet Applications (RIA).
- AngularJS provides developers an options to write client side applications using JavaScript in a clean Model View Controller (MVC) way.
- Applications written in AngularJS are cross-browser compliant. AngularJS automatically 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.min.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.min.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>
```