Ex no: **Angular JS - Navigation Menu**

Date:

Aim:

Program to design a web page with navigation menus using Angular JS.

Procedure:

1. Using Angular's directives to set and read the active variable.
2. When it changes, it causes the HTML that uses it to be updated automatically.
3. In Angular's terminology, this variable is called a model. It is available to all directives in the current scope, and can be accessed in your controllers (more on that in the next example).
4. JavaScript templates are with the {{var}} syntax, the framework sees such a string, it replaces it with the contents of the variable.
5. This operation is repeated every time var is changed.

index.html

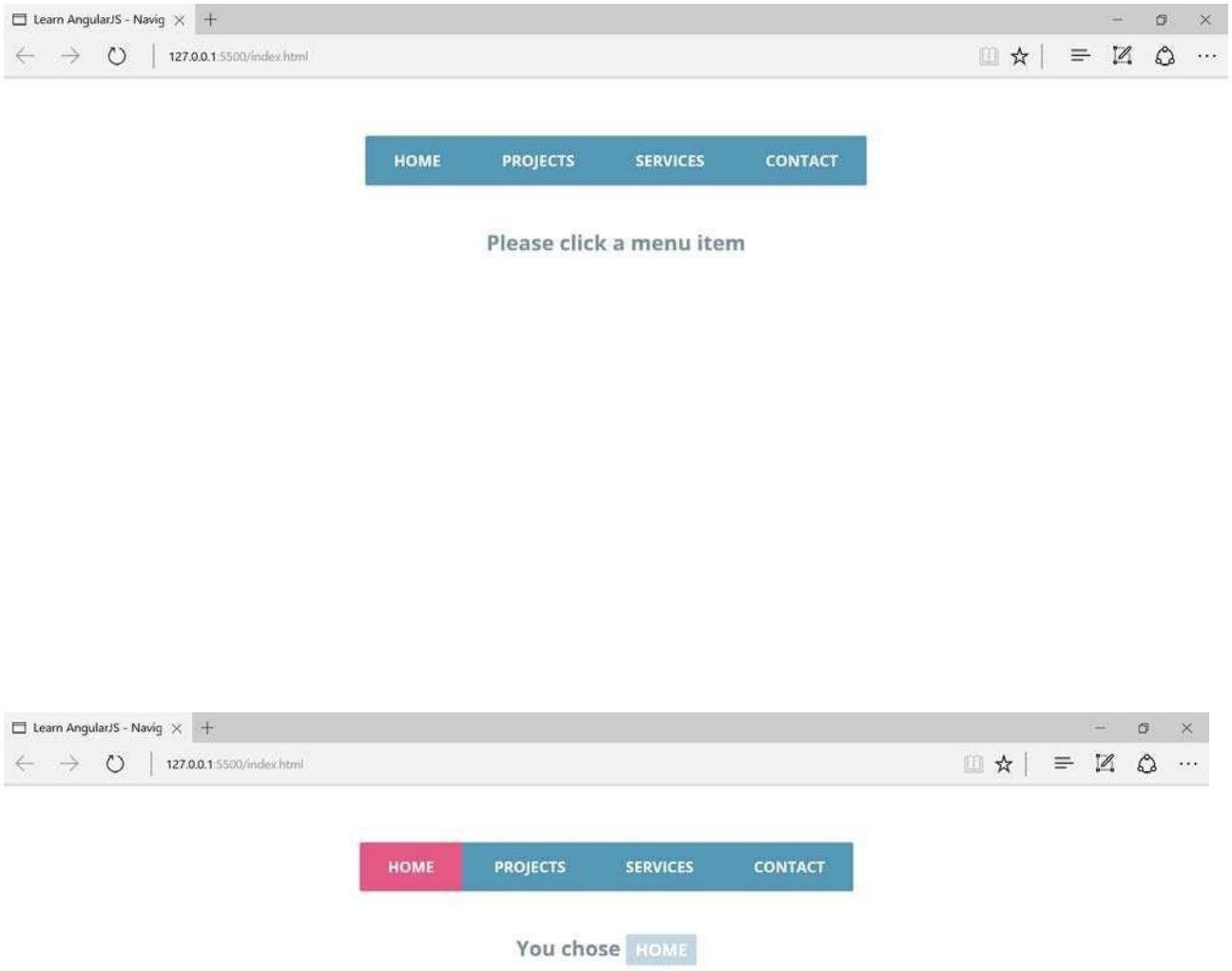
```
<!DOCTYPE html>
<html>

    <head>
        <meta charset="utf-8"/>
        <title>Learn AngularJS - Navigation Menu</title>

        <link href="http://fonts.googleapis.com/css?family=Open+Sans:400,
            700" rel="stylesheet" />

        <!-- The main CSS file -->
        <link href="style.css" rel="stylesheet" />

        <!--[if lt IE 9]>
      <script src="http://html5shiv.googlecode.com/svn/trunk/html5.js">
      </script>
        <![endif]-->
    </head>

    <!-- The ng-app directive tells angular that the code below should
      be evaluated -->

    <body ng-app>

        <!--The navigation menu will get the value of the "active" variable
      as a class. The $event.preventDefault() stops the page from jumping when
      a link is clicked.      -->

        <nav class="{{active}}" ng-click="$event.preventDefault()">

      <!-- When a link in the menu is clicked, we set the active variable -->

      <a href="#" class="home" ng-click="active='home'">Home</a>
      <a href="#" class="projects" ng-click="active='projects'">Projects</a>
      <a href="#" class="services" ng-click="active='services'">Services</a>
      <a href="#" class="contact" ng-click="active='contact'">Contact</a>
      </nav>

      <!-- ng-show will show an element if the value in the quotes is truthful,
            while ng-hide does the opposite. Because the active variable is
            not set initially, this will cause the first paragraph to be
            visible. -->

        <p ng-hide="active">Please click a menu item</p>
        <p ng-show="active">You chose <b>{{active}}</b></p>

        <!-- Include AngularJS from Google's CDN -->
<script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.0.7/angular.min.js">
</script>
    </body>
</html>
```

style.css

```
/*-------------------------------
    Simple reset
--------------------------------*/

*{
    margin:0;
    padding:0;
}

/*-------------------------------
    General Styles
--------------------------------*/

body{
    font:15px/1.3 'Open Sans', sans-serif;
    color: #5e5b64;
    text-align:center;
}

a, a:visited {
    outline:none;
    color:#389dc1;
}

a:hover{
    text-decoration:none;
}

section, footer, header, aside, nav{
    display: block;
}

/*-------------------------------
    The menu
--------------------------------*/

nav{
    display:inline-block;
    margin:60px auto 45px;
    background-color:#5597b4;
    box-shadow:0 1px 1px #ccc;
    border-radius:2px;
}
```

```css
nav a{
    display:inline-block;
    padding: 18px 30px;
    color:#fff !important;
    font-weight:bold;
    font-size:16px;
    text-decoration:none !important;
    line-height:1;
    text-transform: uppercase;
    background-color:transparent;

    -webkit-transition:background-color 0.25s;
    -moz-transition:background-color 0.25s;
    transition:background-color 0.25s;
}

nav a:first-child{
    border-radius:2px 0 0 2px;
}

nav a:last-child{
    border-radius:0 2px 2px 0;
}

nav.home .home,
nav.projects .projects,
nav.services .services,
nav.contact .contact{
    background-color:#e35885;
}

p{
    font-size:22px;
    font-weight:bold;
    color:#7d9098;
}

p b{
    color:#ffffff;
    display:inline-block;
    padding:5px 10px;
    background-color:#c4d7e0;
    border-radius:2px;
    text-transform:uppercase;
    font-size:18px;
}
```

Output :



Result:

Thus the program is executed successfully.

EX No: **Angular JS – Inline Editor**

Date:

Aim:

Program to design a web page with inline editor using Angular JS.

Procedure:

1. Clicking a paragraph will show a tooltip with a text field.
2. Use a controller that will initialize the models and declare two methods for toggling the visibility of the tooltip.
3. Controllers are regular JavaScript functions which are executed automatically by Angular, and which are associated with your page using the ng-controller directive.
4. When the controller function is executed, it gets the special $scope object as a parameter.
5. Adding properties or functions to it makes them available to the view.
6. Using the ng-model binding on the text field tells Angular to update that variable when the value of the field changes (this in turn re-renders the paragraph with the value).

index.html

```
<!DOCTYPE html>
<html>

    <head>
        <meta charset="utf-8"/>
        <title>Learn AngularJS - Inline Editor</title>

        <link href="http://fonts.googleapis.com/css?family=Open+Sans:400,700"
      rel="stylesheet" />

        <!-- The main CSS file -->
        <link href="style.css" rel="stylesheet" />

        <!--[if lt IE 9]>
      <script src="http://html5shiv.googlecode.com/svn/trunk/html5.js">
      </script>
        <![endif]-->
    </head>

    <!-- Notice the controller directive -->
    <body ng-app ng-controller="InlineEditorController">

        <!-- When this element is clicked, hide the tooltip -->
        <div id="main" ng-click="hideTooltip()">

            <!-- This is the tooltip. It is shown only when the showtooltip
            variable is truthful -->
            <div class="tooltip" ng-click="$event.stopPropagation()"
            ng-show="showtooltip">

                <!-- ng-model binds the contents of the text field with the
                  "value" model. Any changes to the text field will
                  automatically update the value, and all other bindings on
                  the page that depend on it. -->

                <input type="text" ng-model="value" />
            </div>

            <!-- Call a method defined in the InlineEditorController that
            toggles the showtooltip varaible -->
            <p ng-click="toggleTooltip($event)">{{value}}</p>

        </div>

        <!-- Include AngularJS from Google's CDN -->
<script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.0.7/angular.min.js">
</script>
        <script src="script.js"></script>
    </body>
</html>
```

script.js

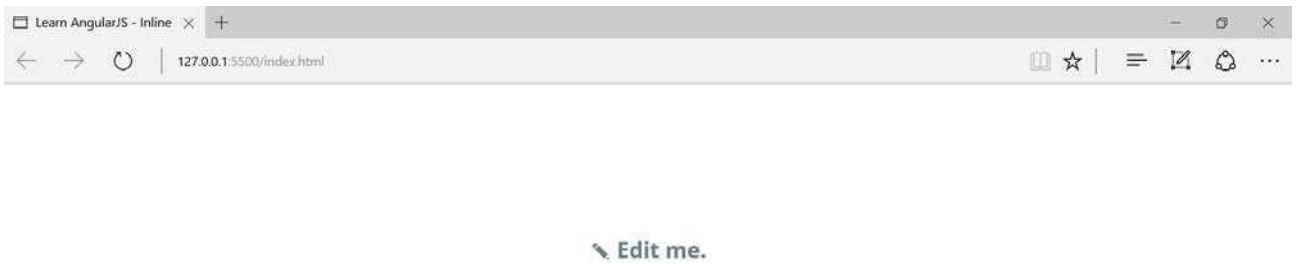
```
// The controller is a regular JavaScript function. It is called
// once when AngularJS runs into the ng-controller declaration.

function InlineEditorController($scope){

    // $scope is a special object that makes
    // its properties available to the view as
    // variables. Here we set some default values:

    $scope.showtooltip = false;
    $scope.value = 'Edit me.';

    // Some helper functions that will be
    // available in the angular declarations

    $scope.hideTooltip = function(){

        // When a model is changed, the view will be automatically
        // updated by by AngularJS. In this case it will hide the tooltip.

        $scope.showtooltip = false;
    }

    $scope.toggleTooltip = function(e){
        e.stopPropagation();
        $scope.showtooltip = !$scope.showtooltip;
    }
}
```

style.css

```
/*-------------------------------
    Simple reset
--------------------------------*/

*{
    margin:0;
    padding:0;
}

/*-------------------------------
    General Styles
--------------------------------*/

body{
    font:15px/1.3 'Open Sans', sans-serif;
    color: #5e5b64;
    text-align:center;
}

a, a:visited {
    outline:none;
    color:#389dc1;
}

a:hover{
    text-decoration:none;
}

section, footer, header, aside, nav{
    display: block;
}

/*-------------------------------
    The edit tooltip
--------------------------------*/

.tooltip{
    background-color:#5c9bb7;

    background-image:-webkit-linear-gradient(top, #5c9bb7, #5392ad);
    background-image:-moz-linear-gradient(top, #5c9bb7, #5392ad);
    background-image:linear-gradient(top, #5c9bb7, #5392ad);

    box-shadow: 0 1px 1px #ccc;
    border-radius:3px;
    width: 290px;
    padding: 10px;

    position: absolute;
    left:50%;
    margin-left:-150px;
    top: 80px;
}
```

```css
.tooltip:after{
    content:'';
    position:absolute;
    border:6px solid #5190ac;
    border-color:#5190ac transparent transparent;
    width:0;
    height:0;
    bottom:-12px;
    left:50%;
    margin-left:-6px;
}

.tooltip input{
    border: none;
    width: 100%;
    line-height: 34px;
    border-radius: 3px;
    box-shadow: 0 2px 6px #bbb inset;
    text-align: center;
    font-size: 16px;
    font-family: inherit;
    color: #8d9395;
    font-weight: bold;
    outline: none;
}

p{
    font-size:22px;
    font-weight:bold;
    color:#6d8088;
    height: 30px;
    cursor:default;
}

p b{
    color:#ffffff;
    display:inline-block;
    padding:5px 10px;
    background-color:#c4d7e0;
    border-radius:2px;
    text-transform:uppercase;
    font-size:18px;
}
p:before{
    content:'✎';
    display:inline-block;
    margin-right:5px;
    font-weight:normal;
    vertical-align: text-bottom;
}

#main{
    height:300px;
    position:relative;
    padding-top: 150px;
}
```

Output:


Learn AngularJS - Inline
127.0.0.1:5500/index.html
Edit me.


Learn AngularJS - Inline
127.0.0.1:5500/index.html
Edit me.
Edit me.

Result:

Thus the program is executed successfully.

Ex No: **Angular JS – Order Form**

Date:

Aim:

Program to design a web page with order form using Angular JS.

Procedure:

1. Code an order form with a total price updated in real time, using another one of Angular's useful features - filters.

2. Filters let modify models and can be chained together using the pipe character |.

3. Use the currency filter, to turn a number into a properly formatted price, complete with a dollar sign and cents. You can easily make your own filters.

4. The ng-repeat binding (docs) is another useful feature of the framework. It lets loop through an array of items and generate markup for them. It is intelligently updated when an item is changed or deleted.

index.html

```
<!DOCTYPE html>
<html>

    <head>
        <meta charset="utf-8"/>
        <title>Learn AngularJS - Order Form</title>

        <link href="http://fonts.googleapis.com/css?family=Cookie|Open+Sans:
      400,700" rel="stylesheet" />

        <!-- The main CSS file -->
        <link href="style.css" rel="stylesheet" />

        <!--[if lt IE 9]>
      <script src="http://html5shiv.googlecode.com/svn/trunk/html5.js">
      </script>
        <![endif]-->
    </head>

    <!-- Declare a new AngularJS app and associate the controller -->
    <body ng-app ng-controller="OrderFormController">

        <form>

            <h1>Services</h1>

            <ul>
                <!-- Loop through the services array, assign a click handler,
                  and set or remove the "active" css class if needed -->
                <li ng-repeat="service in services"
                  ng-click="toggleActive(service)"
                  ng-class="{active:service.active}">
                    <!-- Notice the use of the currency filter, it will
                        format the price -->
                    {{service.name}} <span>{{service.price | currency}}</span>
                </li>
            </ul>

            <div class="total">
                <!-- Calculate the total price of all chosen services.
                  Format it as currency. -->
                Total: <span>{{total() | currency}}</span>
            </div>

        </form>

        <!-- Include AngularJS from Google's CDN -->
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.0.7/angular.min.
js">
</script>
        <script src="script.js"></script>
    </body>
</html>
```

script.js

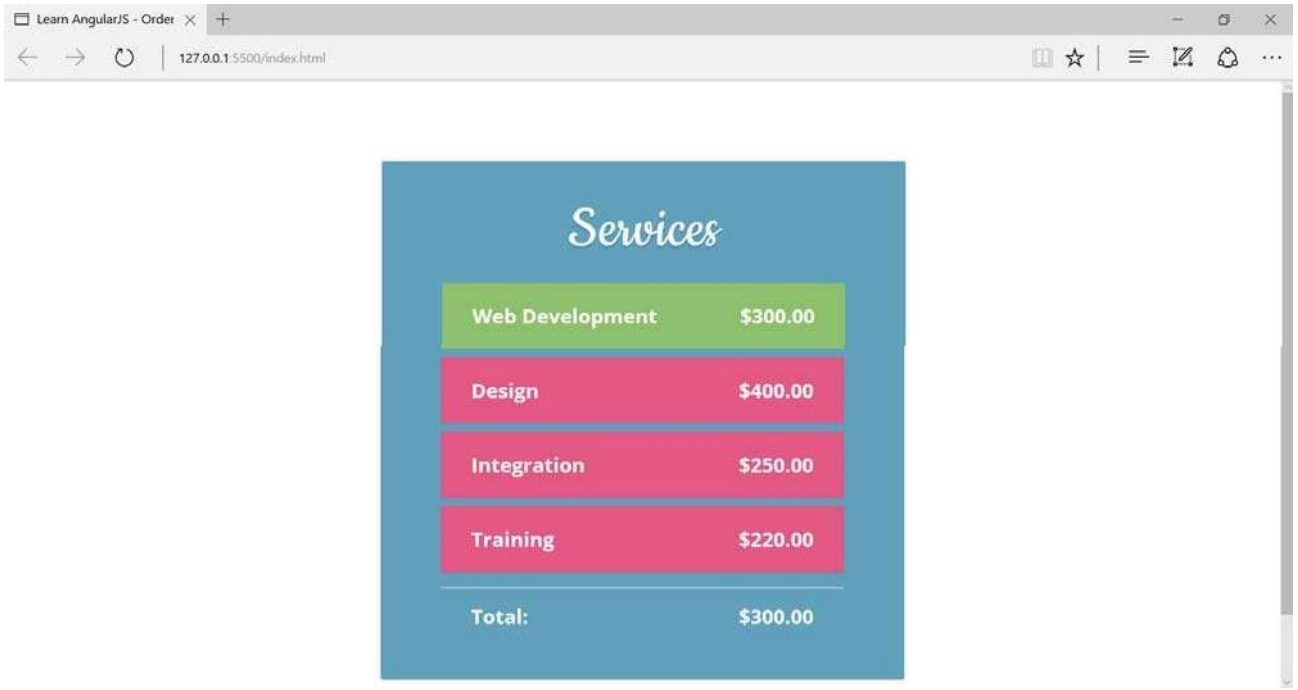
```
function OrderFormController($scope){

    // Define the model properties. The view will loop
    // through the services array and genreate a li
    // element for every one of its items.

    $scope.services = [
        {
            name: 'Web Development',
            price: 300,
            active:true
        },{
            name: 'Design',
            price: 400,
            active:false
        },{
            name: 'Integration',
            price: 250,
            active:false
        },{
            name: 'Training',
            price: 220,
            active:false
        }
    ];

    $scope.toggleActive = function(s){
        s.active = !s.active;
    };

    // Helper method for calculating the total price

    $scope.total = function(){

        var total = 0;

        // Use the angular forEach helper method to
        // loop through the services array:

        angular.forEach($scope.services, function(s){
            if (s.active){
                total+= s.price;
            }
        });

        return total;
    };
}
```

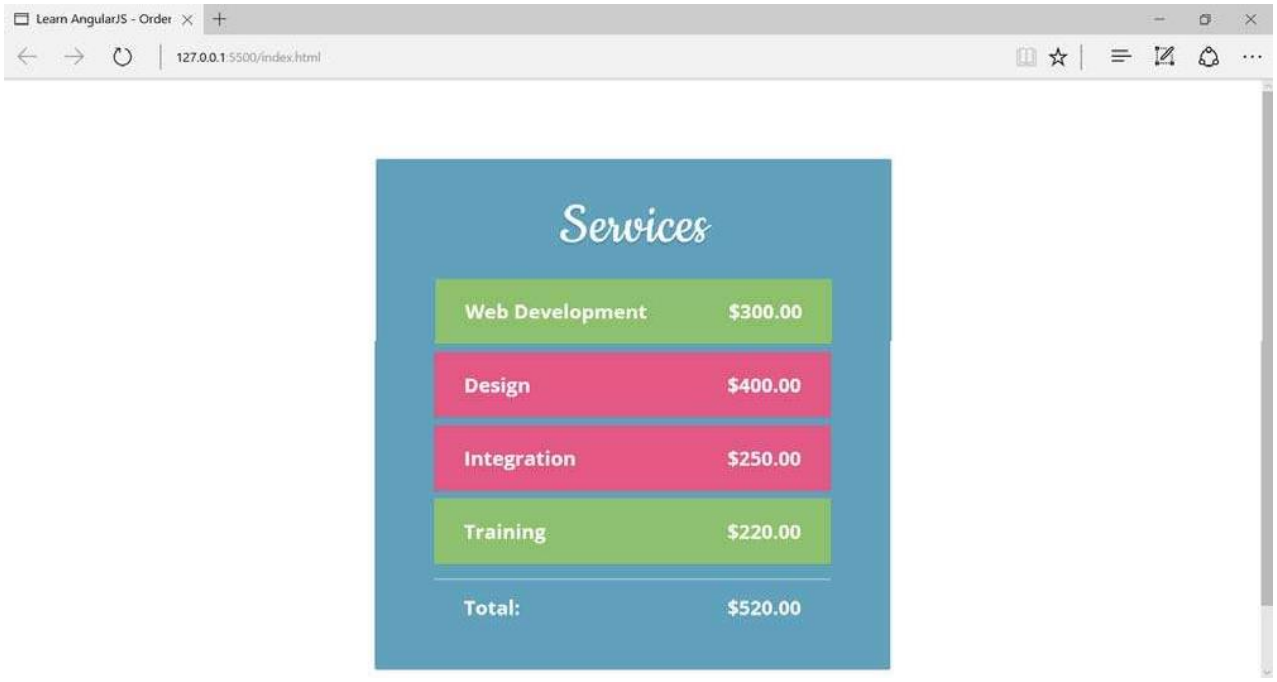
style.css

```
/*-------------------------------
    Simple reset
--------------------------------*/

*{
    margin:0;
    padding:0;
}

/*-------------------------------
    General Styles
--------------------------------*/

body{
    font:15px/1.3 'Open Sans', sans-serif;
    color: #5e5b64;
    text-align:center;
}

a, a:visited {
    outline:none;
    color:#389dc1;
}

a:hover{
    text-decoration:none;
}

section, footer, header, aside, nav{
    display: block;
}

/*-------------------------------
    The order form
--------------------------------*/

form{
    background-color: #61a1bc;
    border-radius: 2px;
    box-shadow: 0 1px 1px #ccc;
    width: 400px;
    padding: 35px 60px;
    margin: 80px auto;
}

form h1{
    color:#fff;
    font-size:64px;
    font-family:'Cookie', cursive;
    font-weight: normal;
    line-height:1;
    text-shadow:0 3px 0 rgba(0,0,0,0.1);
}
```

```css
form ul{
    list-style:none;
    color:#fff;
    font-size:20px;
    font-weight:bold;
    text-align: left;
    margin:20px 0 15px;
}

form ul li{
    padding:20px 30px;
    background-color:#e35885;
    margin-bottom:8px;
    box-shadow:0 1px 1px rgba(0,0,0,0.1);
    cursor:pointer;
}

form ul li span{
    float:right;
}

form ul li.active{
    background-color:#8ec16d;
}

div.total{
    border-top:1px solid rgba(255,255,255,0.5);
    padding:15px 30px;
    font-size:20px;
    font-weight:bold;
    text-align: left;
    color:#fff;
}

div.total span{
    float:right;
}
```

Output :





Result:

Thus the program is executed successfully.

Ex No: **Angular JS – Instant Search**

Date:

Aim:

Program to design a web page with instant search using Angular JS.

Procedure:

1. To filter a list of items by typing into a text field.

2. First have to turn the application into a module.

3. Modules are a way of organizing JavaScript applications into self-contained components that can be combined in new and interesting ways.

4. Angular relies on this technique for code isolation and requires that your application follows it before you can create a filter.

5. There are only two things that you need to do to turn your app into a module:

    1. Use the angular.module("name",[]) function call in your JS. This will instantiate and return a new module;
    2. Pass the name of the module as the value of the ng-app directive.

6. Creating a filter then is as simple as calling the filter() method on the module object returned by angular.module("name", []).

7. Filters follow the Angular.js philosophy - every piece of code that you write should be self-contained, testable and reusable.

8. Use this filter in all your views and even combine it with others through chaining.

index.html

```
<!DOCTYPE html>
<html>

    <head>
        <meta charset="utf-8"/>
        <title>Learn AngularJS - Instant Search</title>

        <link href="http://fonts.googleapis.com/css?family=Cookie|Open+Sans:
      400,700" rel="stylesheet" />

        <!-- The main CSS file -->
        <link href="style.css" rel="stylesheet" />

        <!--[if lt IE 9]>
      <script src="http://html5shiv.googlecode.com/svn/trunk/html5.js">
      </script>
        <![endif]-->
    </head>

    <!-- Initialize a new AngularJS app and associate it with a module named
      "instantSearch"-->
    <body ng-app="instantSearch" ng-controller="InstantSearchController">

        <div class="bar">
            <!-- Create a binding between the searchString model and the
            text field -->
            <input type="text" ng-model="searchString"
            placeholder="Enter your search terms" />
        </div>

        <ul>
            <!-- Render a li element for every entry in the items array. Notice
                 the custom search filter "searchFor". It takes the value of
                 the searchString model as an argument. -->
            <li ng-repeat="i in items | searchFor:searchString">
                <a href="{{i.url}}"><img ng-src="{{i.image}}" /></a>
                <p>{{i.title}}</p>
            </li>
        </ul>

        <!-- Include AngularJS from Google's CDN -->
      <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.0.7/
      angular.min.js"></script>
        <script src="script.js"></script>
    </body>
</html>
```

script.js

```
// Define a new module for our app
var app = angular.module("instantSearch", []);

// Create the instant search filter

app.filter('searchFor', function(){

    // All filters must return a function. The first parameter
    // is the data that is to be filtered, and the second is an
    // argument that may be passed with a colon (searchFor:searchString)

    return function(arr, searchString){

        if(!searchString){
            return arr;
        }

        var result = [];

        searchString = searchString.toLowerCase();

        // Using the forEach helper method to loop through the array
        angular.forEach(arr, function(item){

            if(item.title.toLowerCase().indexOf(searchString) !== -1){
                result.push(item);
            }

        });

        return result;
    };

});

// The controller

function InstantSearchController($scope){

    // The data model. These items would normally be requested via AJAX,
    // but are hardcoded here for simplicity. See the next example for
    // tips on using AJAX.

    $scope.items = [
        {
            url: 'http://tutorialzine.com/2013/07/50-must-have-plugins-for-
extending-twitter-bootstrap/',
            title: '50 Must-have plugins for extending Twitter Bootstrap',
            image: 'http://cdn.tutorialzine.com/wp-
content/uploads/2013/07/featured_4-100x100.jpg'
        },
```

```
        {
            url: 'http://tutorialzine.com/2013/08/simple-registration-system-
php-mysql/',
            title: 'Making a Super Simple Registration System With PHP and MySQ
L',
            image: 'http://cdn.tutorialzine.com/wp-
content/uploads/2013/08/simple_registration_system-100x100.jpg'
        },
        {
            url: 'http://tutorialzine.com/2013/08/slideout-footer-css/',
            title: 'Create a slide-out footer with this neat z-index trick',
            image: 'http://cdn.tutorialzine.com/wp-
content/uploads/2013/08/slide-out-footer-100x100.jpg'
        },
        {
            url: 'http://tutorialzine.com/2013/06/digital-clock/',
            title: 'How to Make a Digital Clock with jQuery and CSS3',
            image: 'http://cdn.tutorialzine.com/wp-
content/uploads/2013/06/digital_clock-100x100.jpg'
        },
        {
            url: 'http://tutorialzine.com/2013/05/diagonal-fade-gallery/',
            title: 'Smooth Diagonal Fade Gallery with CSS3 Transitions',
            image: 'http://cdn.tutorialzine.com/wp-
content/uploads/2013/05/featured-100x100.jpg'
        },
        {
            url: 'http://tutorialzine.com/2013/05/mini-ajax-file-upload-form/',
            title: 'Mini AJAX File Upload Form',
            image: 'http://cdn.tutorialzine.com/wp-
content/uploads/2013/05/ajax-file-upload-form-100x100.jpg'
        },
        {
            url:    'http://tutorialzine.com/2013/04/services-chooser-backbone-
js/',
            title: 'Your First Backbone.js App – Service Chooser',
            image: 'http://cdn.tutorialzine.com/wp-
content/uploads/2013/04/service_chooser_form-100x100.jpg'
        }
    ];

}
```

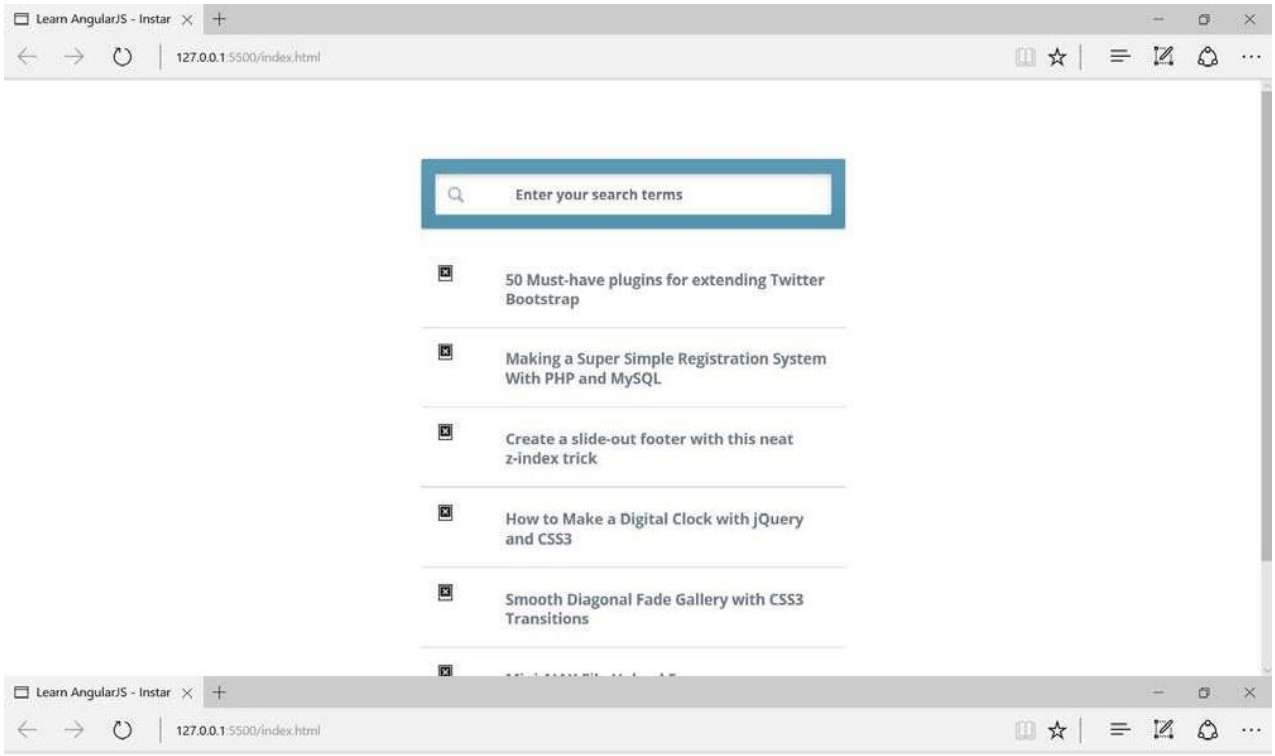
style.css

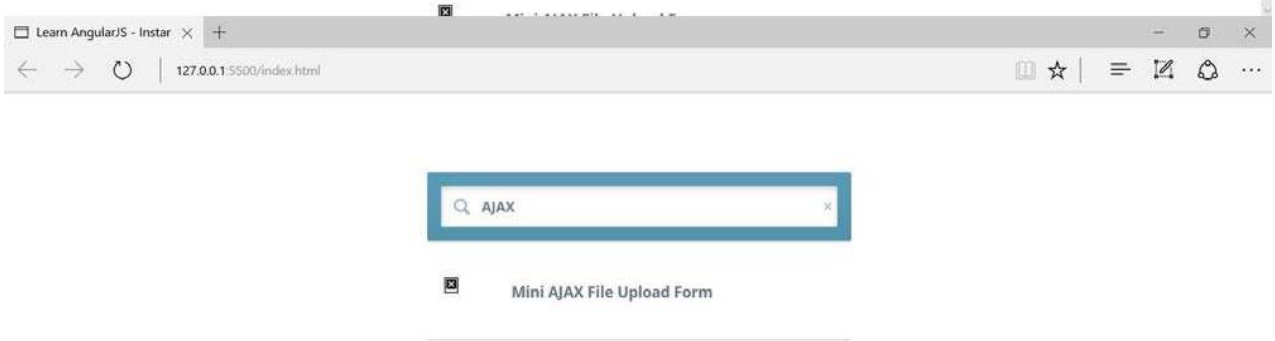
```
/*-------------------------------
    Simple reset
--------------------------------*/

*{
    margin:0;
    padding:0;
}

/*-------------------------------
    General Styles
--------------------------------*/

body{
    font:15px/1.3 'Open Sans', sans-serif;
    color: #5e5b64;
    text-align:center;
}

a, a:visited {
    outline:none;
    color:#389dc1;
}

a:hover{
    text-decoration:none;
}

section, footer, header, aside, nav{
    display: block;
}

/*-------------------------------
    The search input
--------------------------------*/

.bar{
    background-color:#5c9bb7;

    background-image:-webkit-linear-gradient(top, #5c9bb7, #5392ad);
    background-image:-moz-linear-gradient(top, #5c9bb7, #5392ad);
    background-image:linear-gradient(top, #5c9bb7, #5392ad);

    box-shadow: 0 1px 1px #ccc;
    border-radius: 2px;
    width: 400px;
    padding: 14px;
    margin: 80px auto 20px;
    position:relative;
}

.bar input{
    background:#fff no-repeat 13px 13px;
```

```
    background-
image:url(
AGXRFWHRTb2Z0d2FyZQBBZG9iZSBJbWFnZVJlYWR5ccllPAAAAyBpVFh0WE1MOmNvbS5hZG9iZS54bX
AAAAAAADw/eHBhY2tldCBiZWdpbj0i77u/IiBpZD0iVzVNME1wQ2VoaUh6cmVTek5UY3prYzlkIj8+I
Dx4OnhtcG1ldGEgeG1sbnM6eD0iYWRvYmU6bnM6bWV0YS8iIHg6eG1wdGs9IkFkb2JlIFhNUCBDb3Jl
IDUuMC1jMDYwIDYxLjEzNDc3NywgMjAxMC8wMi8xMi0xNzozMjowMCAgICAgICAgICAgIj4gPHJkZjpSREY
geG1sbnM6cmRmPSJodHRwOi8vd3d3LnczLm9yZy8xOTk5LzAyLzIyLXJkZi1zeW50YXgtbnMjIj4gPH
JkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9IiIgeG1sbnM6eG1wPSJodHRwOi8vbnMuYWRvYmUuY29tL
3hhcC8xLjAvIiB4bWxuczp4bXBNTT0iaHR0cDovL25zLmFkb2JlLmNvbS94YXAvMS4wL21tLyIgeG1s
bnM6c3RSZWY9Imh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC9zVHlwZS9SZXNvdXJjZVJlZiMiIHh
tcDpDcmVhdG9yVG9vbD0iQWRvYmUgUGhvdG9zaG9wIENTNSBXaW5kb3dzIiB4bXBNTTpJbnN0YW5jZU
lEPSJ4bXAuaWlkOkU5NEY0RTlFMTA4NzExRTM5RTEzQkFBQzMyRjkyQzVBIiB4bXBNTTpEb2N1bWVud
ElEPSJ4bXAuZGlkOkU5NEY0RTlGMTA4NzExRTM5RTEzQkFBQzMyRjkyQzVBIj4gPHhtcE1NOkRlcml2
ZWRGcm9tIHN0UmVmOmluc3RhbmNlSUQ9InhtcC5paWQ6RTk0RjRFOUMxMDg3MTFFMzlFMTNCQUFDMzJ
GOTJDNUEiIHN0UmVmOmRvY3VtZW50SUQ9InhtcC5kaWQ6RTk0RjRFOUQxMDg3MTFFMzlFMTNCQUFDMz
JGOTJDNUEiLz4gPC9yZGY6RGVzY3JpcHRpb24+IDwvcmRmOlJERj4gPC94OnhtcG1ldGE+IDw/eHBhY
2tldCBlbmQ9InIiPz4DjA/RAAABK0lEQVR42pTSQUdEURjG8dOY0TqmPkGmRcqYD9CmzZAWJRHVRIa0
iFYtM6uofYaiEW2SRJtEi9YxIklp07ZkWswu0v/wnByve7vm5ee8M+85zz1jbt9Os+WiGkYdYxjCOx5
wgFeXUHmtBSzpcCGa+5BJTCjEP+0nKWAT8xqe4ArPGEEVC1hHEbs2oBwdXkM7mj/JLZrad437sCGHOf
UtcziutuYu2v8XUFF/4f6vMK/YgAH1HxkBYV60AR31gxkBYd6xAeF3VzMCwvzOBpypX8V4yuFRzX2d2
gD/l5yjH4fYQEnzkj4fae5rJulF2sMXVrAsaTWttRFu4Osb+1jEDT71/ZveyhouTch2fINQL9hKefKj
uYFfuznXWzXMTabyrvfyIV3M4vhXgAEAUMs7K0J9UJAAAAAASUVORK5CYII=);

    border: none;
    width: 100%;
    line-height: 19px;
    padding: 11px 0;

    border-radius: 2px;
    box-shadow: 0 2px 8px #c4c4c4 inset;
    text-align: left;
    font-size: 14px;
    font-family: inherit;
    color: #738289;
    font-weight: bold;
    outline: none;
    text-indent: 40px;
}

ul{
    list-style: none;
    width: 428px;
    margin: 0 auto;
    text-align: left;
}

ul li{
    border-bottom: 1px solid #ddd;
    padding: 10px;
    overflow: hidden;
}
```

```
ul li img{
    width:60px;
    height:60px;
    float:left;
    border:none;
}

ul li p{
    margin-left: 75px;
    font-weight: bold;
    padding-top: 12px;
    color:#6e7a7f;
```

Output:





Result:

Thus the program is executed successfully.

Ex No: **Angular JS - Switchable Grid**

Date:

Aim:

Program to design a web page with Switchable grid using Angular JS.

Procedure:

1. Write a service that communicates with Instagram's API and returns an array with the most popular photos at the moment.

2. Include one additional Angular.js file in the page:

<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.0.7/angular-resource.min.js">
</script>

3. This includes the ngResource module for easily working with AJAX APIs (the module is exposed as the $resource variable in the code).

4. This file is automatically included in the editor.

index.html

```
<!DOCTYPE html>
<html>

    <head>
        <meta charset="utf-8"/>
        <title>Learn AngularJS - Switchable Grid</title>

        <link href="http://fonts.googleapis.com/css?family=Cookie|Open+Sans:
      400,700" rel="stylesheet" />

        <!-- The main CSS file -->
        <link href="style.css" rel="stylesheet" />

        <!--[if lt IE 9]>
      <script src="http://html5shiv.googlecode.com/svn/trunk/html5.js">
      </script>
        <![endif]-->
    </head>

    <body ng-app="switchableGrid" ng-controller="SwitchableGridController">

        <div class="bar">

            <!-- These two buttons switch the layout varaible,
                 which causes the correct UL to be shown. -->

            <a href="#" class="list-icon" ng-class =
            "{active: layout == 'list'}" ng-click="layout = 'list'"></a>
            <a href="#" class="grid-icon" ng-class =
            "{active: layout == 'grid'}" ng-click="layout = 'grid'"></a>
        </div>

        <!-- We have two layouts. We choose which one to show depending on
      the "layout" binding -->

        <ul ng-show="layout == 'grid'" class="grid">
            <!-- A view with big photos and no text -->
            <li ng-repeat="p in pics">
                <a href="{{p.link}}" target="_blank">
                  <img ng-src="{{p.images.low_resolution.url}}" /></a>
            </li>
        </ul>

        <ul ng-show="layout == 'list'" class="list">
            <!-- A compact view smaller photos and titles -->
            <li ng-repeat="p in pics">
                <a href="{{p.link}}" target="_blank">
                  <img ng-src="{{p.images.thumbnail.url}}" /></a>
                <p>{{p.caption.text}}</p>
            </li>
        </ul>

        <!-- Include AngularJS from Google's CDN and the resource module -->
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.0.7/angular.min.
js"></script>
```

```
<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.0.7/angular-
resource.min.js"></script>
        <script src="script.js"></script>
    </body>
</html>
```

script.js

```
// Define a new module. This time we declare a dependency on
// the ngResource module, so we can work with the Instagram API

var app = angular.module("switchableGrid", ['ngResource']);

// Create and register the new "instagram" service
app.factory('instagram', function($resource){

    return {
        fetchPopular: function(callback){

// The ngResource module gives us the $resource service. It makes working with
// AJAX easy. Here I am using a client_id of a test app. Replace it with yours.

var api = $resource('https://api.instagram.com/v1/media/popular?client_id=:clie
nt_id&callback=JSON_CALLBACK',{
      client_id: '642176ece1e7445e99244cec26f4de1f'
},{
// This creates an action which we've chosen to name "fetch". It issues
// an JSONP request to the URL of the resource. JSONP requires that the
// callback=JSON_CALLBACK part is added to the URL.
      fetch:{method:'JSONP'}
      });

      api.fetch(function(response){

      // Call the supplied callback function
      callback(response.data);
      });
      }
    }
});

// The controller. Notice that I've included our instagram service which we
// defined below. It will be available inside the function automatically.

function SwitchableGridController($scope, instagram){

    // Default layout of the app. Clicking the buttons in the toolbar
    // changes this value.

    $scope.layout = 'grid';

    $scope.pics = [];

    // Use the instagram service and fetch a list of the popular pics
    instagram.fetchPopular(function(data){

        // Assigning the pics array will cause the view
        // to be automatically redrawn by Angular.
        $scope.pics = data;
    });

}
```

style.css

```
/*-------------------------------
    Simple reset
--------------------------------*/

*{
    margin:0;
    padding:0;
}

/*-------------------------------
    General Styles
--------------------------------*/

body{
    font:15px/1.3 'Open Sans', sans-serif;
    color: #5e5b64;
    text-align:center;
}

a, a:visited {
    outline:none;
    color:#389dc1;
}

a:hover{
    text-decoration:none;
}

section, footer, header, aside, nav{
    display: block;
}

/*-------------------------------
    The search input
--------------------------------*/

.bar{
    background-color:#5c9bb7;

    background-image:-webkit-linear-gradient(top, #5c9bb7, #5392ad);
    background-image:-moz-linear-gradient(top, #5c9bb7, #5392ad);
    background-image:linear-gradient(top, #5c9bb7, #5392ad);

    box-shadow: 0 1px 1px #ccc;
    border-radius: 2px;
    width: 580px;
    padding: 10px;
    margin: 80px auto 25px;
    position:relative;
    text-align:right;
    line-height: 1;
}
```

```
.bar a{
    background:#4987a1 center center no-repeat;
    width:32px;
    height:32px;
    display:inline-block;
    text-decoration:none !important;
    margin-right:5px;
    border-radius:2px;
}
.bar a.active{
    background-color:#c14694;
}

.bar a.list-icon{
    background-
image:url(
AGXRFWHRTb2Z0d2FyZQBBZG9iZSBJbWFnZVJlYWR5ccllPAAAAyBpVFh0WE1MOmNvbS5hZG9iZS54bX
AAAAAAADw/eHBhY2tldCBiZWdpbj0i77u/IiBpZD0iVzVNME1wQ2VoaUh6cmVTek5UY3prYzlkIj8+I
Dx4OnhtcG1ldGEgeG1sbnM6eD0iYWRvYmU6bnM6bWV0YS8iIHg6eG1wdGs9IkFkb2JlIFhNUCBDb3Jl
IDUuMC1jMDYwIDYxLjEzNDc3NywgMjAxMC8wMi8xMi0xNzozMjowMCAgICAgICAgICAgIj4gPHJkZjpSREY
geG1sbnM6cmRmPSJodHRwOi8vd3d3LnczLm9yZy8xOTk5LzAyLzIyLXJkZi1zeW50YXgtbnMjIj4gPH
JkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9IiIgeG1sbnM6eG1wPSJodHRwOi8vbnMuYWRvYmUuY29tL
3hhcC8xLjAvIiB4bWxuczp4bXBNTT0iaHR0cDovL25zLmFkb2JlLmNvbS94YXAvMS4wL21tLyIgeG1s
bnM6c3RSZWY9Imh0dHA6Ly9ucy5jb20veGFwLzEuMC9zVHlwZS9SZXNvdXJjZVJlZiMiIHh
tcDpDcmVhdG9yVG9vbD0iQWRvYmUgUGhvdG9zaG9wIENTNSBXaW5kb3dzIiB4bXBNTTpJbnN0YW5jZU
lEPSJ4bXAuaWlkOkYzNkFCQ0ZBMTBCRTExRTM5NDk4RDFEM0E5RkQ1NEZCIiB4bXBNTTpEb2N1bWVud
ElEPSJ4bXAuZGlkOkYzNkFCQ0ZCMTBCRTExRTM5NDk4RDFEM0E5RkQ1NEZCIj4gPHhtcE1NOkRlcml2
ZWRGcm9tIHN0UmVmOmluc3RhbmNlSUQ9InhtcC5paWQ6RjM2QUJDRjgxMEJFMTFFMzk0OThEMUQzQTl
GRDU0RkIiIHN0UmVmOmRvY3VtZW50SUQ9InhtcC5kaWQ6RjM2QUJDRjkxMEJFMTFFMzk0OThEMUQzQT
lGRDU0RkIiLz4gPC9yZGY6RGVzY3JpcHRpb24+IDwvcmRmOlJERj4gPC94OnhtcG1ldGE+IDw/eHBhY
2tldCBlbmQ9InIiPz7h1bLqAAAAWUlEQVR42mL8////BwYGBn4GCACxBRlIAIxAA/4jaXoPEkMyjJ+A
/g9MDJQBRhYg8RFqMwg8RJIUINYLFDmBUi+ADQAF1n8ofk9yIAy6WPg4GgtDMRYAAgwAdLYwLAoIwPg
AAAAASUVORK5CYII=);
}

.bar a.grid-icon{
    background-
image:url(
AGXRFWHRTb2Z0d2FyZQBBZG9iZSBJbWFnZVJlYWR5ccllPAAAAyBpVFh0WE1MOmNvbS5hZG9iZS54bX
AAAAAAADw/eHBhY2tldCBiZWdpbj0i77u/IiBpZD0iVzVNME1wQ2VoaUh6cmVTek5UY3prYzlkIj8+I
Dx4OnhtcG1ldGEgeG1sbnM6eD0iYWRvYmU6bnM6bWV0YS8iIHg6eG1wdGs9IkFkb2JlIFhNUCBDb3Jl
IDUuMC1jMDYwIDYxLjEzNDc3NywgMjAxMC8wMi8xMi0xNzozMjowMCAgICAgICAgICAgIj4gPHJkZjpSREY
geG1sbnM6cmRmPSJodHRwOi8vd3d3LnczLm9yZy8xOTk5LzAyLzIyLXJkZi1zeW50YXgtbnMjIj4gPH
JkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9IiIgeG1sbnM6eG1wPSJodHRwOi8vbnMuYWRvYmUuY29tL
3hhcC8xLjAvIiB4bWxuczp4bXBNTT0iaHR0cDovL25zLmFkb2JlLmNvbS94YXAvMS4wL21tLyIgeG1s
bnM6c3RSZWY9Imh0dHA6Ly9ucy5jb20veGFwLzEuMC9zVHlwZS9SZXNvdXJjZVJlZiMiIHh
tcDpDcmVhdG9yVG9vbD0iQWRvYmUgUGhvdG9zaG9wIENTNSBXaW5kb3dzIiB4bXBNTTpJbnN0YW5jZU
lEPSJ4bXAuaWlkOkYzNkFCQ0ZBMTBCRjExRTNBMDlGRTYyOTlBNDdCN0I4IiB4bXBNTTpEb2N1bWVud
ElEPSJ4bXAuZGlkOkYzNkFCQ0ZCMTBCRjExRTNBMDlGRTYyOTlBNDdCN0I4Ij4gPHhtcE1NOkRlcml2
ZWRGcm9tIHN0UmVmOmluc3RhbmNlSUQ9InhtcC5paWQ6MERCQzJDMTIxMEJGMTFFM0EwOUZFNjI5OUE
0N0I3QjgiIHN0UmVmOmRvY3VtZW50SUQ9InhtcC5kaWQ6MERCQzJDMTMxMEJGMTFFM0EwOUZFNjI5OU
E0N0I3QjgiLz4gPC9yZGY6RGVzY3JpcHRpb24+IDwvcmRmOlJERj4gPC94OnhtcG1ldGE+IDw/eHBhY
2tldCBlbmQ9InIiPz4MjPshAAAAXklEQVR42mL4////h/8I8B6IGaCYKHFGEMnAwCDIAAHvgZgRyiZK
nImBQsACxB+hNoDAQyQ5osQZIT4gH1DsBZABH6AB8x/JaQzEig++WPiII7Rxio/GwmCIBYAAAwAwVIz
Mp1R0aQAAAABJRU5ErkJggg==);
}
```

```css
.bar input{
    background:#fff no-repeat 13px 13px;

    border: none;
    width: 100%;
    line-height: 19px;
    padding: 11px 0;

    border-radius: 2px;
    box-shadow: 0 2px 8px #c4c4c4 inset;
    text-align: left;
    font-size: 14px;
    font-family: inherit;
    color: #738289;
    font-weight: bold;
    outline: none;
    text-indent: 40px;
}

/*-------------------------------
    List layout
--------------------------------*/

ul.list{
    list-style: none;
    width: 500px;
    margin: 0 auto;
    text-align: left;
}

ul.list li{
    border-bottom: 1px solid #ddd;
    padding: 10px;
    overflow: hidden;
}

ul.list li img{
    width:120px;
    height:120px;
    float:left;
    border:none;
}

ul.list li p{
    margin-left: 135px;
    font-weight: bold;
    color:#6e7a7f;
}
```

```
/*-------------------------------
    Grid layout
--------------------------------*/

ul.grid{
    list-style: none;
    width: 570px;
    margin: 0 auto;
    text-align: left;
}

ul.grid li{
    padding: 2px;
    float:left;
}

ul.grid li img{
    width:280px;
    height:280px;
    display:block;
    border:none;
}
```

Output:

Ex No: **Angular JS – Single Page Application**

Date:

Aim:

Program to develop an attractive web pages using Bootstrap.

Procedure:

1. Define a simple controller:
2. After created module and controller, use them in our HTML.
3. Include angular script and app.js that we built.
4. Specify module in ng-app attribute and controller in ng-controller attribute.
5. Start working on adding single page application support.
6. Make a single page application and don't want any page refreshes, use Angular's routing capabilities.
7. Include angular-route script after the main angular script.
8. Specify that the module depends on ngRoute module to be able to use it.
9. The next thing is to distinguish common HTML for every page. This HTML will be layout of the website.
10. Then specify the place where HTML of each page will be placed in our layout. There is a ng-view directive for that.
11. ng-view is an Angular directive that will include the template of the current route (for example, /blog or /about) in the main layout file.
12. Configure the routes. Use $routeProvider service from the ngRoute module.
13. For each route, specify templateUrl and controller.
14. If user will try to go to the route that does not exist, handle this by using otherwise function. In our case, we will redirect user to the "/" route:
15. Build controllers for every route (already specified their names in routeProvider).

index.html

```
<!doctype html>
<html ng-app="myApp">
  <head>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/angular.js/1.4.7/
      angular.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/angular.js/1.4.7/
      angular-route.min.js"></script>
  </head>
  <body>
    <script type="text/ng-template" id="pages/home.html">
      <h1>Home</h1>
      <h3>{{message}}</h3>
    </script>
    <script type="text/ng-template" id="pages/courses.html">
      <h1>Courses</h1>
      <h3>{{message}}</h3>
    </script>
    <script type="text/ng-template" id="pages/contactus.html">
      <h1>Contact Us</h1>
      <h3>{{message}}</h3>
    </script>

    <a href="#/">Home</a>
    <a href="#/courses">Courses</a>
    <a href="#/contactus">Contact Us</a>

    <div ng-view></div>

    <script src="app.js"></script>
  </body>
</html>
```

app.js

```
var app = angular.module('myApp', ['ngRoute']);

app.config(function($routeProvider) {
  $routeProvider

  .when('/', {
    templateUrl : 'pages/home.html',
    controller  : 'HomeController'
  })

  .when('/courses', {
    templateUrl : 'pages/courses.html',
    controller  : 'CoursesController'
  })

  .when('/contactus', {
    templateUrl : 'pages/contactus.html',
    controller  : 'ContactUsController'
  })

  .otherwise({redirectTo: '/'});
});

app.controller('HomeController', function($scope) {
  $scope.message = 'Welcome to REC';
});

app.controller('CoursesController', function($scope) {
  $scope.message = 'AERO, AUTO, BIOMED, BIOTECH, CHEMICAL, CIVIL, CSE, CSBC,
      ECE, EEE, FT, IT, MCT, MECH';
});

app.controller('ContactUsController', function($scope) {
  $scope.message = 'Rajalakshmi Nagar, Thandalam, Chennai - 602 105';
});
```

Output:

Home Courses Contact Us

Home

Welcome to REC

Home Courses Contact Us

Courses

AERO, AUTO, BIOMED, BIOTECH, CHEMICAL, CIVIL, CSE, CSBC, ECE, EEE, FT, IT, MCT, MECH

Home Courses Contact Us

Contact Us

Rajalakshmi Nagar, Thandalam, Chennai - 602 105

Result:

Thus the program is executed successfully.