Class ID:10 Name: Venkatesh Kandula

CS5590 APS - Programming for Web/Cloud based Application

ASSIGNMENT 3

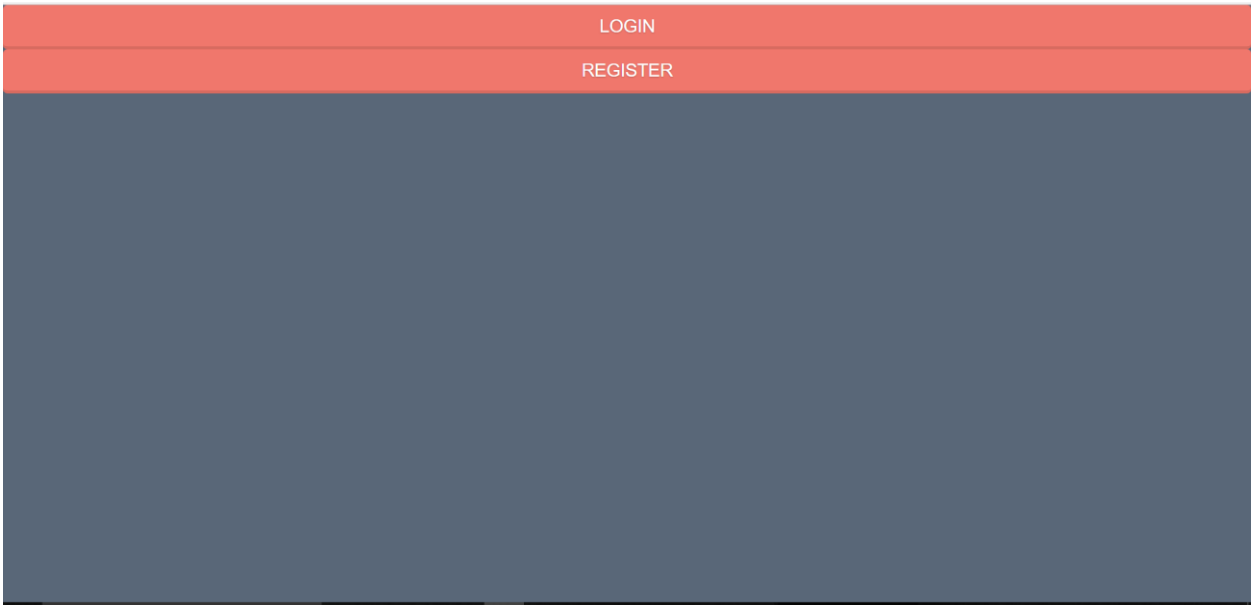
TASK1:

Opening page

HTML code:

```
<a href="#login" class="btn btn-primary btn-lg"><input type="submit" value="LOGIN" class ="log-in-button" ></a>

<a href="#register" class="btn btn-danger btn-lg"><input type="submit" value="REGISTER" class ="log-in-button"></a>
```

Image:



Login Page

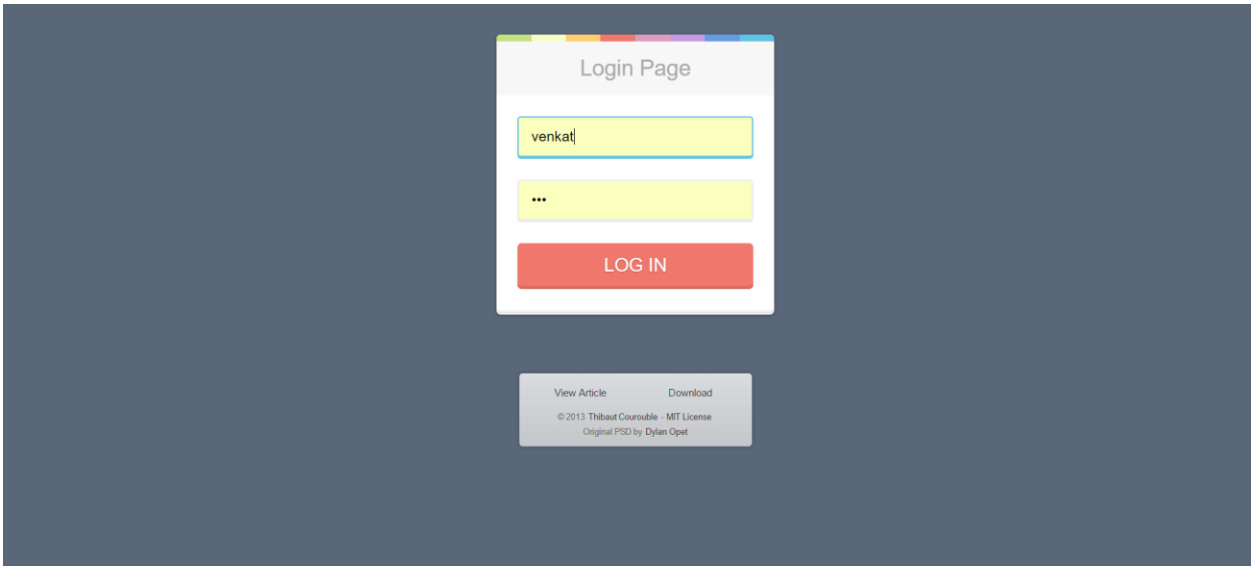
HTML code:

```
<!DOCTYPE html>
<!--[if lt IE 7]> <html class="lt-ie9 lt-ie8 lt-ie7" lang="en"> <![endif]-->
<!--[if IE 7]> <html class="lt-ie9 lt-ie8" lang="en"> <![endif]-->
<!--[if IE 8]> <html class="lt-ie9" lang="en"> <![endif]-->
<!--[if gt IE 8]><!--> <html lang="en"> <!--<![endif]-->
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
  <title>Sign Up Form</title>
  <link rel="stylesheet" href="CSS/style.css">
  <!--[if lt IE 9]><script
src="//html5shim.googlecode.com/svn/trunk/html5.js"></script><![endif]-->
  <script>
    function loginCheck()
    { if(localStorage) {
      form = document.forms["log"];
      var logname= form["user"].value;
      var logpwd    = form["pword"].value;
      var name = localStorage.getItem("regname");
      var pwd    = localStorage.getItem("regpwd");
      if (logname == name && logpwd == pwd) {
        return true;
      } else{
        alert("check login details");
        return false;
```

```
            }
        }
        }
    </script>
</head>
<body>
<form class="sign-up" method="post" action="home.html" name="log" onsubmit="return loginCheck()">
    <h1 class="sign-up-title">Login Page</h1>
    <input type="text" id="user"  class="sign-up-input" placeholder="Enter username?" autofocus>
    <input type="password" id = "pword" class="sign-up-input" placeholder="enter password">
    <input type="submit" value="LOG IN" class="log-in-button">



</form>

<div class="about">
    <p class="about-links">
        <a href="http://www.cssflow.com/snippets/sign-up-form" target="_parent">View Article</a>
        <a href="http://www.cssflow.com/snippets/sign-up-form.zip" target="_parent">Download</a>
    </p>
    <p class="about-author">
        &copy; 2013 <a href="http://thibaut.me" target="_blank">Thibaut Courouble</a> -
        <a href="http://www.cssflow.com/mit-license" target="_blank">MIT License</a><br>
        Original PSD by <a href="http://dribbble.com/shots/1037950-Sign-up-freebie" target="_blank">Dylan Opet</a>
```

```
  </p>
</div>
</body>
</html>
```

Image:



Login Page

venkat

•••

© 2013 Thibaut Courouble - MIT License
Original PSD by Dylan Opet

Registration Page

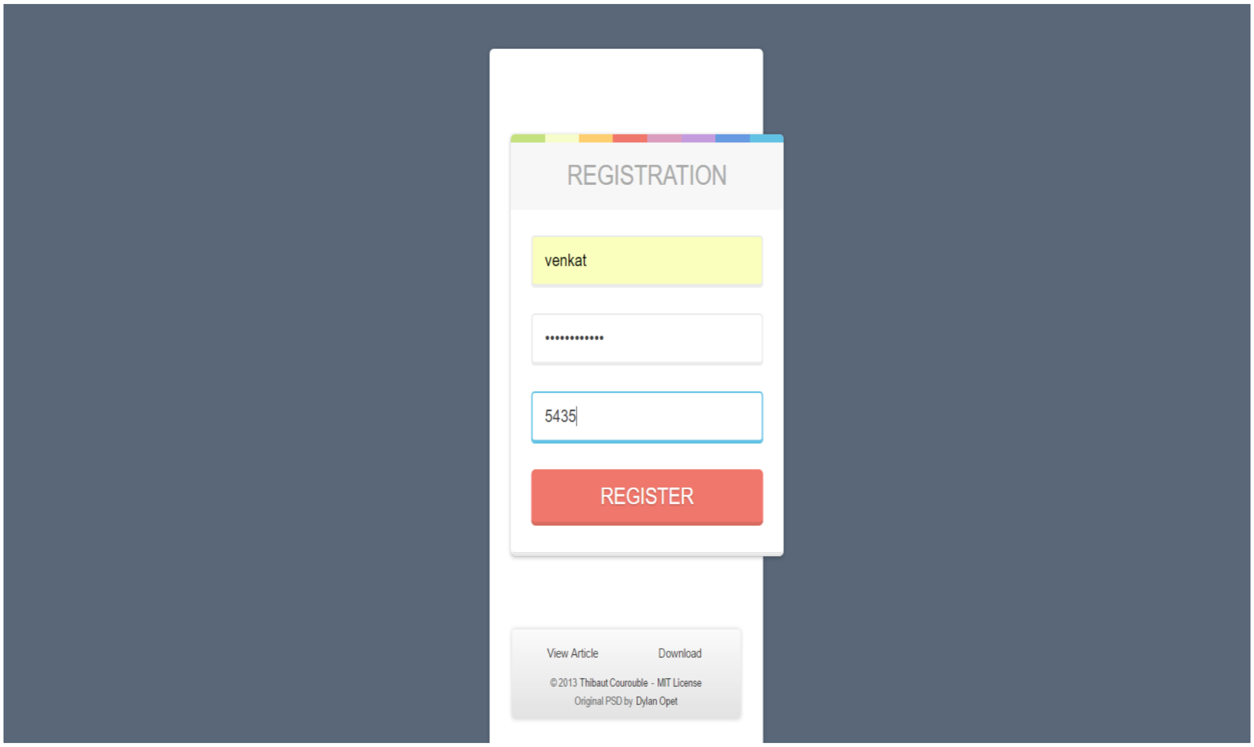
HTML code:

```
<!DOCTYPE html>
<!--[if lt IE 7]> <html class="lt-ie9 lt-ie8 lt-ie7" lang="en"> <![endif]-->
<!--[if IE 7]> <html class="lt-ie9 lt-ie8" lang="en"> <![endif]-->
<!--[if IE 8]> <html class="lt-ie9" lang="en"> <![endif]-->
<!--[if gt IE 8]><!--> <html lang="en"> <!--<![endif]-->
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
  <title>Sign Up Form</title>
  <link rel="stylesheet" href="CSS/style.css">
  <!--[if lt IE 9]><script
src="//html5shim.googlecode.com/svn/trunk/html5.js"></script><![endif]-->
  <script>
    function registration()
    {
      if (typeof(Storage) !== "undefined") {
       /* form = document.forms["reg"];
        var localpwd = form["password"].value;
        var localname = form["username"].value;
        var localaddress = form["address"].value; */

        //name validation
        localStorage.setItem('regname', localname);
        localStorage.setItem('regpwd', localpwd);
        localStorage.setItem('regaddr', localaddress);
        return true;
      }
```

```
        }
    </script>
</head>
<body>
<form class="sign-up" method="post" action="login.html" name="reg" onsubmit="return registration()">
    <h1 class="sign-up-title">REGISTRATION</h1>
    <input type="text"  name="username" class="sign-up-input" placeholder="Enter username" autofocus>
    <input type="password" name="password" class="sign-up-input" placeholder="enter password">
    <input type="text" name="address" class="sign-up-input" placeholder="enter address">

    <input type="submit" value="REGISTER" class="register-button">
</form>

<div class="about">
    <p class="about-links">
        <a href="http://www.cssflow.com/snippets/sign-up-form" target="_parent">View Article</a>
        <a href="http://www.cssflow.com/snippets/sign-up-form.zip" target="_parent">Download</a>
    </p>
    <p class="about-author">
        &copy; 2013 <a href="http://thibaut.me" target="_blank">Thibaut Courouble</a> -
        <a href="http://www.cssflow.com/mit-license" target="_blank">MIT License</a><br>
        Original PSD by <a href="http://dribbble.com/shots/1037950-Sign-up-freebie" target="_blank">Dylan Opet</a>
    </p>
```

```
</div>
</body>
</html>
```

Image:


REGISTRATION
REGISTER
View Article
Download
©2013 Thibaut Courouble - MIT License
Original PSD by Dylan Opet

venkat

••••••••••••

5435

Home Page

HTML code:

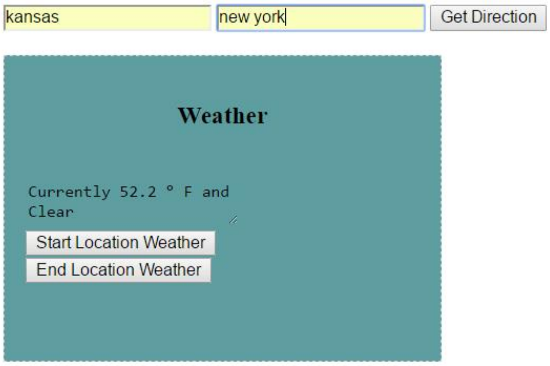
```
<!DOCTYPE html>
<html lang="" data-ng-app="Direction">
<head>
  <meta t="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Angular Demo</title>
```

```
    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.4.4/angular.min.js"
type="application/javascript"></script>

    <script
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyBIPIKVULg06gzKIYipDIj0UVdNX37ye
Qg&v=3.exp&sensor=true"></script>

    <link rel="stylesheet" href="css/main.css">


    <style>

        #map-canvas {

            width: 800px;

            height: 800px;

        }




        .weatherModule {

            background: cadetblue;

            height:200px;

            width:300px;

            float: inherit;

            border: 1px dashed #ccc;

            padding: 1em;

        }

    </style>

</head>


<body>
```

```
<div class="wrapp" ng-controller="googlemapoutput">
  <form class="wrap">
  <div class="row">
    <input type="text" placeholder="enter start location" ng-model="startlocation"
id="startlocation" autocomplete="on"/>
    <input type="text" placeholder="enter End location" ng-model="endlocation"
id="endlocation" autocomplete="on"/>
    <button id="btn" ng-click="calcRoute()">
      Get Direction
    </button>
    <br/></br>
  </div>
  </form>

  <div class="weatherModule">
    <center><label><h3>Weather</h3></label></center>
    <p class="currentConditions" ng-bind-html-unsafe="currentweather.html" ></p>
    <br>
    <div class="currentIcon" ng-bind-html-unsafe="currentIcon.html"></div>



    <button id="weather1" ng-click="Weather1()">
      Start Location Weather
    </button>
    <br>
    <button id="weather2" ng-click="Weather2()">
      End Location Weather
    </button >
    </div>




    <br/><br/>
    <div id="map-canvas" style="height: 400px"></div>
</div>
</body>
</html>


<script>
    angular.module('Direction', [])
        .controller('googlemapoutput', function ($scope) {

            var map;
            var mapOptions;
            var directionsDisplay = new google.maps.DirectionsRenderer({
                draggable: true
            });
            var directionsService = new google.maps.DirectionsService();

            $scope.initialize = function () {
                var pos = new google.maps.LatLng(0, 0);
                var mapOptions = {
                    zoom: 3,
    center: pos
  };

  map = new google.maps.Map(document.getElementById('map-canvas'),
    mapOptions);

  new google.maps.places.Autocomplete(
    (document.getElementById('startlocation')), {
      types: ['geocode']
    });

  new google.maps.places.Autocomplete(
    (document.getElementById('endlocation')), {
      types: ['geocode']
    });




};
$scope.calcRoute = function () {
  var end = document.getElementById('endlocation').value;
  var start = document.getElementById('startlocation').value;

  var request = {
    origin: start,
    destination: end,
    travelMode: google.maps.TravelMode.DRIVING
```

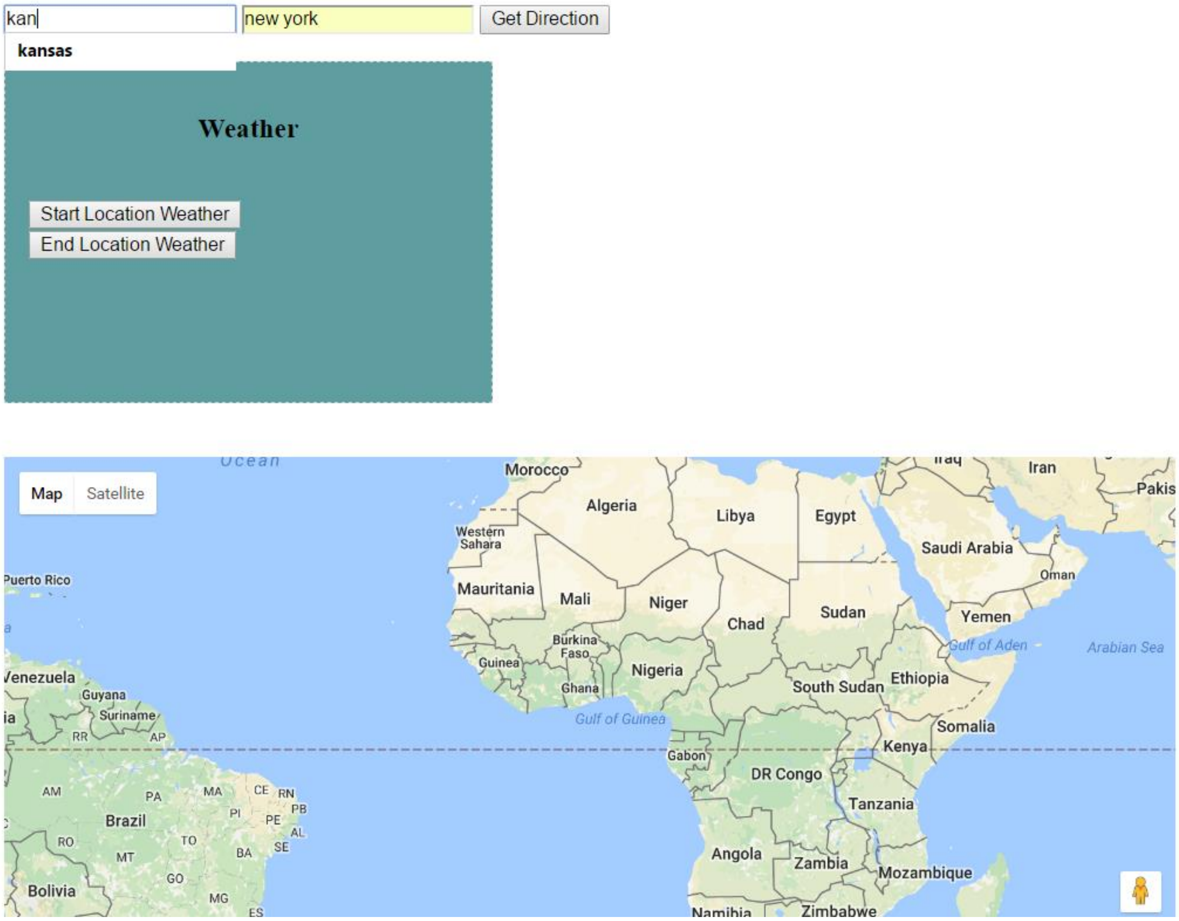
```
      };


      directionsService.route(request, function (response, status) {
        if (status == google.maps.DirectionsStatus.OK) {
          directionsDisplay.setMap(map);
          directionsDisplay.setDirections(response);
          console.log(status);
        }


      });
    };


    $scope.Weather1 = function() {
      var start = document.getElementById("startlocation").value.toString();

$http.get('http://api.wunderground.com/api/5f2d268c33a07857/conditions/q/'+start+'.json').s
uccess(function(sourcedata){
        console.log(sourcedata);
        sourcetemp = sourcedata.current_observation.temp_f;
        sourceicon = sourcedata.current_observation.icon_url;
        sourceweather = sourcedata.current_observation.weather;
        console.log(sourcetemp);
        $scope.currentweather= $sce.trustAsHtml("<p><b>At "+start+"</b><br>Currently
"+sourcetemp+ "&deg; F and "+ sourceweather+"</p>");
        $scope.currentIcon=$sce.trustAsHtml("<img src='" + sourceicon  +"'/><br>");
      })
    };
    $scope.Weather2 = function() {
            var end = document.getElementById("endlocation").value.toString();


            $http.get(
'http://api.wunderground.com/api/36b799dc821d5836/conditions/q/'+end+'.json').success(function(destinationdata){
                console.log(destinationdata);
                temp2 = destinationdata.current_observation.temp_f;
                icon2 = destinationdata.current_observation.icon_url;
                weather2 = destinationdata.current_observation.weather;
                console.log(temp2);
                $scope.currentweather = $sce.trustAsHtml("<p><b>At "+ end +"</b><br>Currently " +temp2 +" &deg; F and " + weather2 + "</p>")
                $scope.currentIcon= $sce.trustAsHtml("<img src='" + icon2  +"'/>")
            });
        };




        google.maps.event.addDomListener(window, 'load', $scope.initialize);


    });




</script>
```

Image:



Weather

```
Currently 52.2 ° F and
Clear
```



kan

kansas

new york

TASK2 :

Visualizing the Twitter friends list through D3.JS

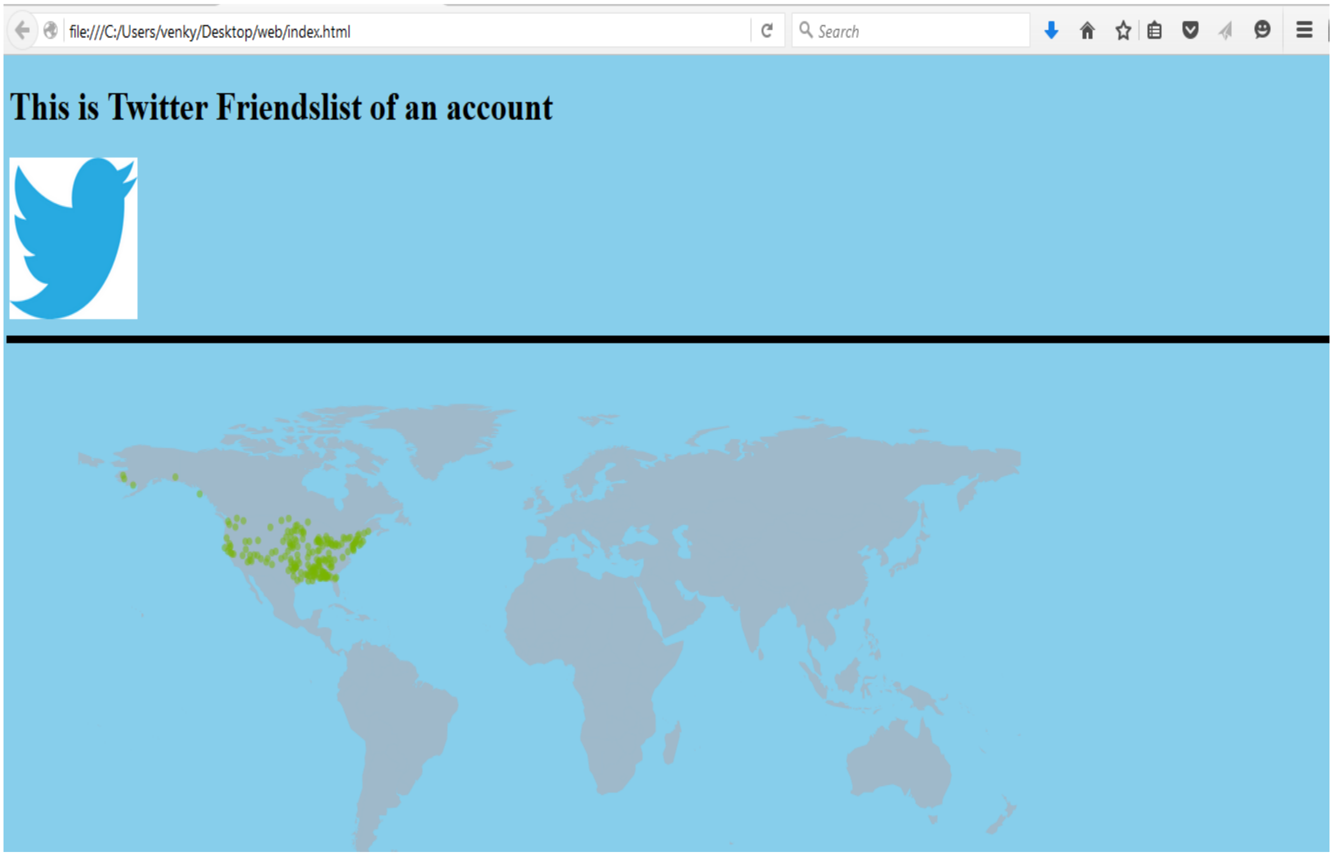
HTML code:

```
<html>
 <head><style>
 img {
   width:100%;
};
 p {
   border: 1px solid black;
   outline-color:red;
}
p.solid {outline-style: solid;}
</style>
  <script src="http://d3js.org/d3.v3.min.js"></script>
  <script src="gistfile1.js"></script>
  </head>
  <body style="background-color:skyblue;">
  <!-- RGB Value

In HTML, a color can be specified as an RGB value, using this formula:

rgb(red, green, blue)

Each parameter (red, green, and blue) defines the intensity of the color between 0 and 255.

For example, rgb(255, 0, 0) is displayed as red, because red is set to its highest value (255) and the others are set to 0.

To display the color black, all color parameters must be set to 0, like this: rgb(0, 0, 0).

To display the color white, all color parameters must be set to 255, like this: rgb(255, 255, 255).
-->
<p class="dashed"><h1>This is Twitter Friendslist of an example account</h1></p><!-- Outline Color

The outline-color property is used to set the color of the outline.

The color can be set by:
  name - specify a color name, like "red"

  RGB - specify a RGB value, like "rgb(255,0,0)"

  Hex - specify a hex value, like "#ff0000"

  invert - performs a color inversion (which ensures that the outline is visible, regardless of color background) -->

        <img src="twitter.png" alt="twitter Icon" style="width:128px;height:128px;">

        <!-- Twitter (/ˈtwɪtər/) is an online news and social networking service where users post and interact with messages, "tweets", restricted to 140 characters. Registered users can post tweets, but those who are unregistered can only read them. Users access Twitter through its website interface, SMS or a mobile device app.[10] Twitter Inc. is based in San Francisco, California, United States, and has more than 25 offices around the world.[11]  -->


        <p class="solid"><script>
         var data = []


         var h = 540, w = 1080


         var vis = d3.select("body")
         .append("svg:svg")
         .attr("width", w)
         .attr("height", h)


         xy = d3
         .geo
         .equirectangular()
         .scale(150)
         .translate([w/2, h/2])
```

```
path = d3

.geo

.path()

.projection(xy)


countries = vis

.append('svg:g')

.attr('id', 'countries')


countries.selectAll('path')

.data(countries_data.features)

.enter()

.append('svg:path')

.attr('d', path)

.attr('fill', 'rgba(255,99,71,0.2)')

.attr('stroke-width', 1);


d3.geo.path()


d3.csv("friend.csv", function(parsedRows) {

    data = parsedRows

    for (i = 0; i < data.length; i++) {

      var mapCoords = this.xy([data[i].long, data[i].lat])

      data[i].lat = mapCoords[0]

      data[i].long = mapCoords[1]

    }
```

```
                vis.selectAll("circle")
                .data(data)
                .enter().append("svg:circle")
                .attr("cx", function(d) { return d.lat })
                .attr("cy", function(d) { return d.long })
                .attr("stroke-width", "none")
                .attr("fill", function() { return "rgb(120,180,0)" })
                .attr("fill-opacity", .5)
                .attr("r", function() { return 3 })
         })

        </script></p>
  </body>
</html>
```

Output:



This is Twitter Friendslist of an account

The CSV data for latitude and longitude as follows:

Here the symbol(-) is before longitude(long).

lat	long
31.95376	-89.2345
30.68586	-95.0179
38.94575	-104.57
42.74135	-78.0521
30.68801	-81.9059
34.49167	-88.2011
32.85049	-86.6115

43.08751	-88.1779
40.67331	-80.6414
40.44726	-92.227
33.93011	-89.3429
46.88385	-96.3509
41.51962	-87.4011
31.42128	-97.797
39.60417	-116.005
32.46047	-85.68
41.98934	-88.1012
48.88434	-99.6209
33.53457	-89.3126
41.43157	-74.3919
41.97602	-114.658
41.30717	-85.0643
32.52884	-94.9717
42.57451	-84.8114
41.11668	-98.0503
32.52944	-86.3282
48.3008	-102.406
40.65139	-98.0798
32.76125	-89.5301
32.11931	-88.1275
31.07447	-81.4278
41.63695	-99.3404

33.9915	-90.0781
38.86872	-77.0075
35.98531	-85.8093
32.93679	-87.0889
36.37094	-82.1737
61.93396	-162.893
44.42506	-70.8078
42.60356	-97.2526
42.59136	-72.5228
44.11673	-72.8271
41.25131	-72.0316
42.40419	-88.6334
42.11389	-96.7289
42.35083	-86.2561
44.98731	-99.9535
34.98561	-106.009
35.71766	-108.202
33.60167	-97.7756
41.29029	-98.9906
35.38899	-95.6017
36.17528	-96.1518
34.19593	-96.6756
43.10318	-78.7033
40.39944	-96.1714

41.46736	-84.5066
42.88062	-76.7816
39.00445	-99.8929
38.36674	-84.2841
31.60017	-85.2388
31.04247	-86.3116
31.3649	-85.3097
42.73608	-94.2452
37.71799	-117.238
35.76827	-115.33
36.31109	-114.464
33.88455	-91.5343
35.63778	-88.128
36.01123	-88.1233
36.1166	-87.7382
34.36677	-89.9001
32.82588	-91.1877
38.14612	-120.648
39.94377	-122.171
38.53146	-121.786
40.02764	-124.073
40.5221	-121.818
31.29701	-89.8128
31.01622	-87.4468
29.97576	-92.0842

31.56683	-91.5001
31.96366	-92.6603
31.99072	-93.3074
48.95897	-119.412
48.04981	-122.801
38.53916	-106.046
40.51491	-98.9457
37.14172	-79.0164
43.30695	-100.628
36.70972	-110.237
42.40266	-88.3759
44.94996	-94.0669
40.53717	-81.9544
41.62787	-74.1338
41.95323	-116.188
31.8833	-85.4849
41.22581	-79.441
42.04808	-88.0526
31.39699	-84.8953
30.219	-96.3743
41.893	-89.0783
47.81833	-92.2917
31.05126	-87.068
40.04636	-98.0601
44.43746	-93.9127

43.45214	-85.2648
37.85158	-96.2917
40.96676	-74.7802
31.40989	-86.0888
33.18262	-90.8307
45.89858	-94.8739
33.19156	-99.7179
31.13781	-83.4531
34.77231	-88.1659
61.43706	-142.904
60.90583	-162.439
41.09406	-83.2125
31.74328	-84.4193
42.99845	-123.31
40.78142	-82.9747
31.00694	-84.8776
36.7743	-102.51
31.86127	-89.8029
39.70548	-75.033
58.98897	-159.05
34.3401	-83.1335
36.69591	-84.3916
34.17402	-83.5607
33.12546	-90.0256
39.77842	-74.9478
56.3288	-133.61

35.01619	-84.3463
35.2226	-83.419
36.61064	-83.7374
36.39728	-85.6416
32.43878	-86.5104
45.00839	-69.2398
42.2913	-73.7103
43.61534	-73.2746
42.00013	-71.1968
41.69597	-88.1292
45.23054	-96.566
42.35004	-83.4583
43.40333	-98.8295
44.73107	-94.7147
45.29329	-97.5137
44.86248	-98.5295
34.14699	-97.1227
34.0751	-97.1067
34.1404	-96.812
33.17533	-97.8284
41.391	-83.6301
41.15186	-81.4166
35.99221	-113.817
41.13144	-81.7649
43.18167	-76.1278

38.72752	-90.5083
39.07045	-88.5335
38.47149	-91.8153
39.30251	-88.0041
42.36425	-74.066
40.54281	-86.6817
30.84578	-85.6014
34.85007	-97.5864
35.15507	-97.4704
36.5667	-98.8523
37.45419	-100.992
30.08721	-90.5827
37.78746	-114.422
37.74533	-111.57
37.8597	-112.816
34.83398	-92.2579
33.14518	-90.5153
34.28035	-87.6004
36.59288	-86.4769
36.52589	-88.9156
37.12072	-92.3112
40.82788	-72.7487
39.21915	-74.7948
40.97115	-74.9975
40.57028	-75.4883

46.64858	-97.0056
34.23759	-98.742
38.99017	-122.9
38.20242	-121.268
35.73338	-98.7306
41.26321	-122.272
31.67101	-92.1585
31.60585	-90.4093
30.87046	-87.8174
47.10392	-122.287
46.27111	-106.624
46.32763	-119.971
48.19019	-116.909
42.24715	-111.338
36.80834	-107.651
42.00943	-101.769