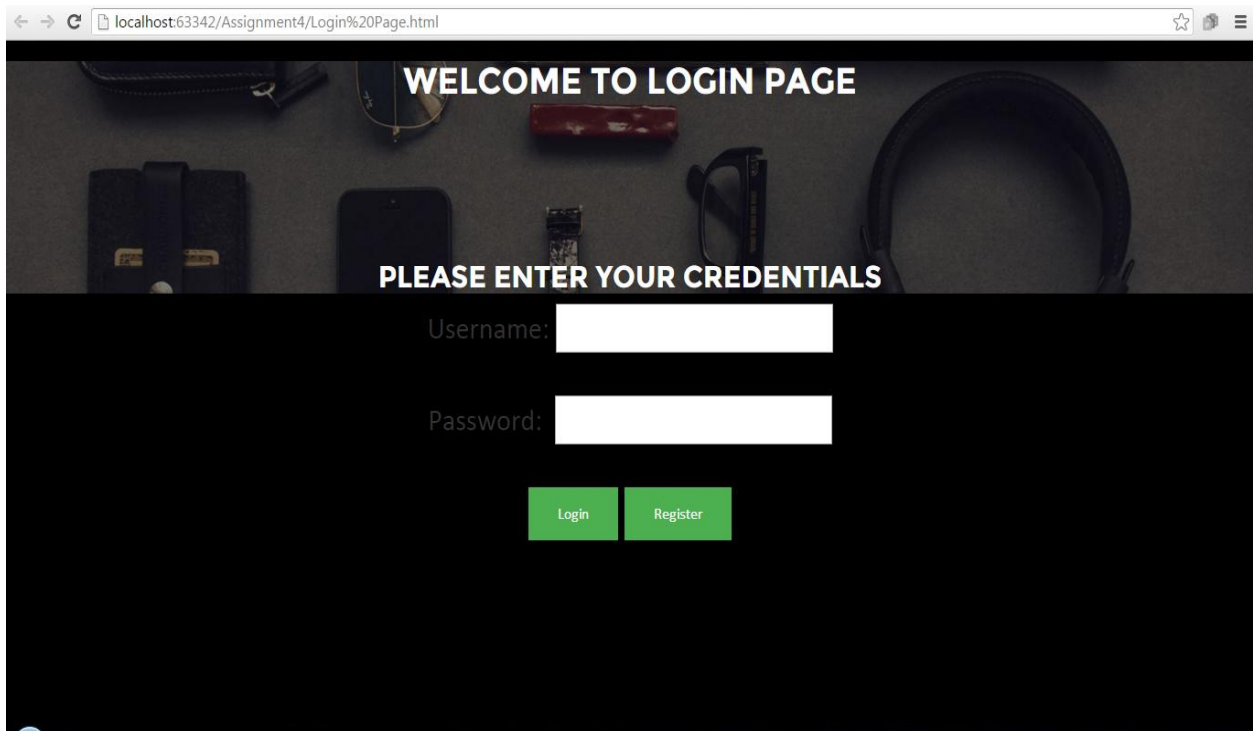


## ASSIGNMENT 4

### Login Page:

Creating a login page.



The screenshot shows a web browser window with the address bar displaying "localhost:63342/Assignment4/Login%20Page.html". The page has a dark background with a top-down view of various items like a wallet, keys, a phone, and headphones. The text "WELCOME TO LOGIN PAGE" is at the top in white. Below it, "PLEASE ENTER YOUR CREDENTIALS" is also in white. There are two white text input fields: the first is labeled "Username:" and the second is labeled "Password:". At the bottom, there are two green buttons: "Login" and "Register".

This login page contains:

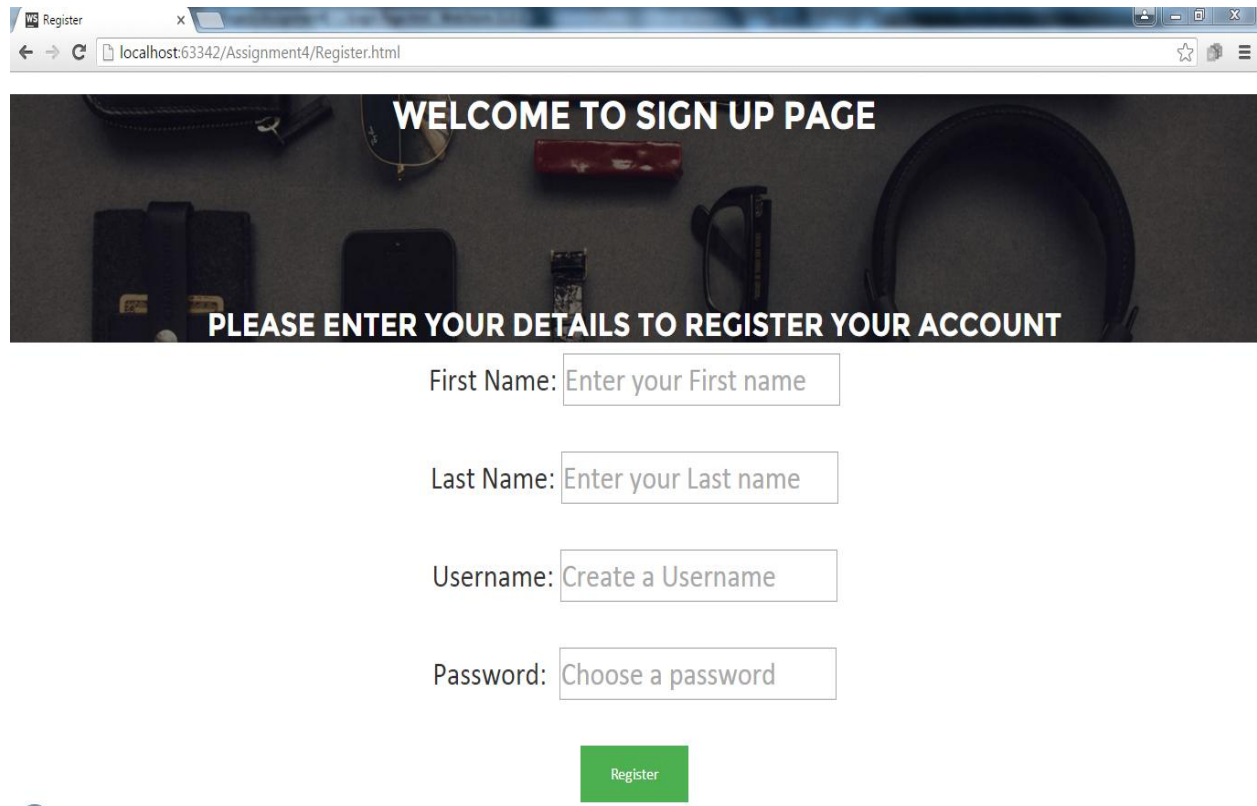
2 labels

2 text fields for username and password

2 buttons for Login and Register.

## Register Page:

On clicking register in login page, register page appears.



WS Register x

localhost:63342/Assignment4/Register.html

**WELCOME TO SIGN UP PAGE**

**PLEASE ENTER YOUR DETAILS TO REGISTER YOUR ACCOUNT**

First Name:

Last Name:

Username:

Password:

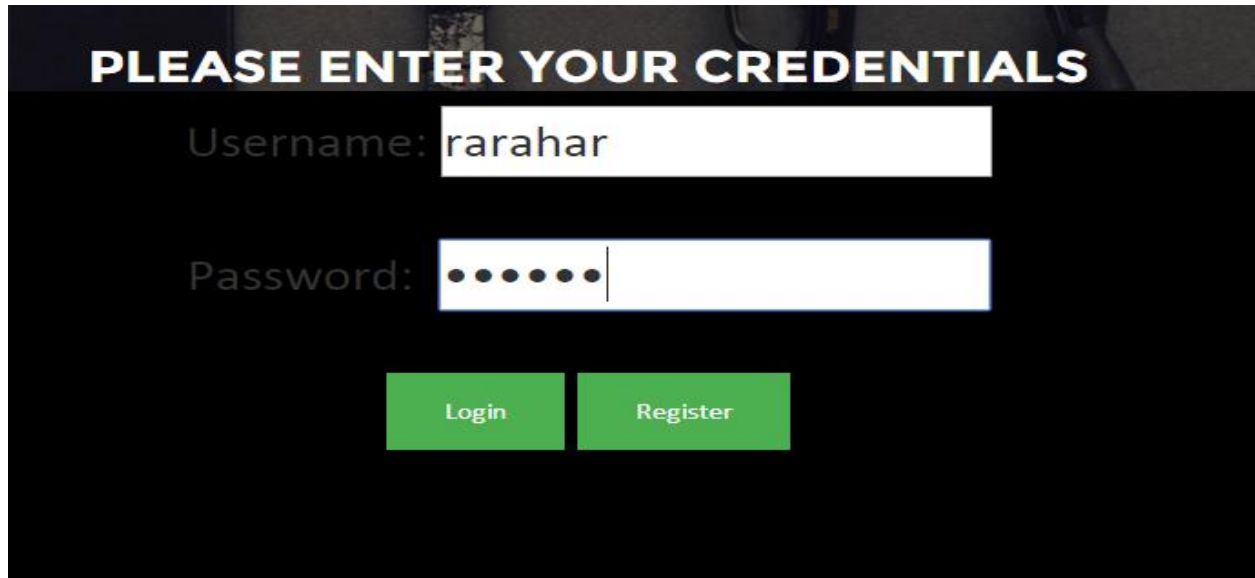
4 labels

4 text fields: Each for First name, last name, username, password

1Button: For Register

## Home Page:

On successfyl login

A screenshot of a login and registration form. The form has a dark background with a header that says "PLEASE ENTER YOUR CREDENTIALS" in white, bold, uppercase letters. Below the header, there are two input fields: "Username:" with the text "rarahar" entered, and "Password:" with a series of dots indicating a masked password. Below these fields are two green buttons: "Login" and "Register".

**PLEASE ENTER YOUR CREDENTIALS**

Username:

Password:

A home page appears that shows

- 2 Text fields for source and destination and a button for results
- A map that gives directions from source to destination.
- A table which gives weather conditions in both the cities.

MapSatellite

+  
-

Map data ©2016 Google, INEGI Terms of Use

## Weather

|             | Source       | Destination  |
|-------------|--------------|--------------|
| Location    | Atlanta, US  | Chicago, US  |
| Temperature | 41°F         | 24°F         |
| Description | Sky is Clear | Sky is Clear |
| Pressure    | 995.32 mb    | 1005.21 mb   |
| Wind Speed  | 3.21 mph     | 3.46 mph     |

The weather part gives temperature,pressure and windspeed,etc.

## HTML Part:

```
<body style="background-color: lightgoldenrodyellow">

<div class="wrapp" ng-controller="Mapcontroller">
  <div class="row">
    <form ng-submit="showDirection()">
      <input type="text" ng-model="fromlocation" id="fromlocation" placeholder="Source location"/>
      <input type="text" ng-model="tolocation" id="tolocation" placeholder="Destination location"/>
      <button id="btn" ng-click="getfromData();gettoData()">
        Get Direction and weather
      </button>
    </form>
  </div>

  <div id="map-canvas" style="height: 300px;"></div>

</div>

<h1>Weather</h1>
<table style="text-align: center;" cellpadding="5px" border="1px">

  <tr>
    <th></th>
    <th>Source</th>
    <th>Destination</th>
  </tr>
  <tr>
    <td>Location</td>
    <td><span id="result">{{ fromdata.data.name }}, {{ fromdata.data.sys.country }}</span></td>
    <td><span id="result">{{ todata.data.name }}, {{ todata.data.sys.country }}</span></td>
  </tr>

  <tr>
    <td>Temperature</td>
    <td><span id="result">{{ fromdata.data.main.temp | kelvinToFahrenheit | number: 0 }}&deg;F</span></td>
    <td><span id="result">{{ todata.data.main.temp | kelvinToFahrenheit | number: 0 }}&deg;F</span></td>
  </tr>
  <tr>
    <td>Description</td>
    <td><span id="result">{{ fromdata.data.weather[0].description }}</span></td>
    <td><span id="result">{{ todata.data.weather[0].description }}</span></td>
  </tr>
  <tr>
    <td>Pressure</td>
    <td><span id="result">{{ fromdata.data.main.pressure }} mb</span></td>
    <td><span id="result">{{ todata.data.main.pressure }} mb</span></td>
  </tr>
  <tr>
    <td>Wind Speed</td>
    <td><span id="result">{{ fromdata.data.wind.speed }} mph</span></td>
    <td><span id="result">{{ todata.data.wind.speed }} mph</span></td>
  </tr>
</table>

</div>
</div>
```

## JS part:

```
<script>
    var app = angular.module('MapApp', [])
    app.controller('Mapcontroller', function ($scope, $http) {

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

        $scope.getfromData = function () {
            $http.get("http://api.openweathermap.org/data/2.5/weather?q=" + document.getElementById('fromlocation').value +
                ",uk&appid=44db6a862fba0b067b1930da0d769e98")
                .then(function (response) {
                    $scope.fromdata = response;
                }, function (response) {
                    $scope.fromdata = "Something went wrong";
                });
        }

        $scope.gettoData = function () {
            $http.get("http://api.openweathermap.org/data/2.5/weather?q=" + document.getElementById('tolocation').value +
                ",uk&appid=44db6a862fba0b067b1930da0d769e98")
                .then(function (response) {
                    $scope.todata = response;
                }, function (response) {
                    $scope.todata = "Something went wrong";
                });
        }
    });

```

```

    $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);
    };

    $scope.showDirection = function () {
        var end = document.getElementById('tolocation').value;
        var start = document.getElementById('fromlocation').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);
            }
        });
    };

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

```

```
        google.maps.event.addDomListener(window, 'load', $scope.initialize);
    });
    app.filter('kelvinToFahrenheit', function() {
        return function(kelvin) {
            return (parseFloat(kelvin) - 273.15) * 1.80 + 32.00;
        };
    });
</script>
```