<!DOCTYPE html>

<html lang="en">

<head>

1.1

<meta charset="UTF-8">

<title>Demo</title>

1.2

<link rel="stylesheet"

href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"

integrity="sha384-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u"

crossorigin="anonymous">

1.3

<script src="//code.jquery.com/jquery-3.1.1.min.js"></script>

1.4

<script src="https://use.fontawesome.com/64f18c2489.js" type="text/javascript"></script>

1.5

<script src="webjars/angularjs/1.6.1/angular.min.js" type ="text/javascript"></script>

1.6

<script src="webjars/angularjs/1.6.1/angular-route.js" type ="text/javascript"></script>

1.7

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

1.8

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

</head>

2.1

<body ng-app="ParasolApp" ng-controller="home">

2.2

<nav id="mainNav" class="navbar navbar-default navbar-fixed-top navbar-custom" style="position: relative;">

<!-- Brand and toggle get grouped for better mobile display -->

2.3

<div class="navbar-header page-scroll">

<h1 class="navbar-brand" align="center">Parasol</h1>

<img src="/images/revature\_logo.png" alt="logo" class="logo">

</div>

</nav>

2.4

<!-- Login -->

<div class="container" ng-show="authenticated">

2.5

<a href="/auth/sflogin">Salesforce: </a>

2.6

<button type="button" ng-click="toSalesforce()"

class="btn btn-primary btn-lg btn3d">Login</button>

</div>

<!-- Module -->

2.7

<ul class="nav navbar-nav navbar-right" ng-show="authenticated">

<li><a href="#!/moduleList"><i class="fa fa-home"></i>Modules</a></li>

<li><a href="#!/moduleReg"><i class="fa fa-shield"></i>Module Registration</a></li>

</ul>

<div ng-view></div>

2.8

<!-- Footer -->

<footer class="text-center">

<div class="footer-above">

<div class="container">

<div class="row">

<div class="footer-col col-md-4">

<h3>Location</h3>

<p>11730 Plaza America Dr #205

<br>Reston, VA 20190</p>

</div>

<div class="footer-col col-md-4">

<h3>Connect with us</h3>

<ul class="list-inline">

<li>

<a href="https://www.facebook.com/revature/" class="btn-social btn-outline"><i class="fa fa-facebook"></i></a>

</li>

<li>

<a href="https://plus.google.com/+RevatureLife" class="btn-social btn-outline"><i class="fa fa-google-plus"></i></a>

</li>

<li>

<a href="https://twitter.com/wearerevature?lang=en" class="btn-social btn-outline"><i class="fa fa-twitter"></i></a>

</li>

<li>

<a href="https://www.linkedin.com/company/revature" class="btn-social btn-outline"><i class="fa fa-linkedin"></i></a>

</li>

</ul>

</div>

<div class="footer-col col-md-4">

<h3></h3>

<p>"Code like a Boss"</p>

</div>

</div>

</div>

</div>

<div class="footer-below">

<div class="container">

<div class="row">

<div class="col-lg-12">

Copyright &copy; Revature 2017

</div>

</div>

</div>

</div>

</footer>

3.1

<script type="text/javascript">

3.2

var app = angular.module("ParasolApp", ["ngRoute"]);

3.3

app.config(function($routeProvider) {

3.4

$routeProvider

.when('/', {

templateUrl: '/modList.html'

})

.when('/moduleList', {

templateUrl: '/modList.html'

})

.when('/moduleReg', {

templateUrl: '/modReg.html'

})

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

});

3.5

app.controller("home", function($scope, $http, $location, $window) {

3.6 var self = $scope;

3.7 self.toSalesforce = toSalesforce;

3.8 self.modules = [{"moduleName":"Scheduling","moduleURL":"http://www.google.com"},{"moduleName":"Bootcamp","moduleURL":"http://www.google.com"},{"moduleName":"Associate Info","moduleURL":"http://www.google.com"},{"moduleName":"Associate Evaluation","moduleURL":"http://www.google.com"},{"moduleName":"Reports","moduleURL":"http://www.google.com"}]

//self.modules = "";

3.9

self.setHover = setHover;

self.hover = null;

self.authenticated = false;

self.admin = false;

3.10

/\*$http({

method : 'GET',

url : 'auth/login'

}).then(function(success) {

console.log("Success!");

self.modules = success.data.modules;

self.authenticated = true;

self.admin = success.data.admin;

console.log(self.authenticated)

}, function(error) {

console.log("Inside error");

console.log(error);

self.authenticated = false;

self.admin = false;

});\*/

function toSalesforce()

{

$window.location.href = '/auth/sflogin';

}

function setHover(module)

{

self.hover = module;

}

3.11

//Kevin's Mod

self.url = null;

self.moduleName = null;

self.dataLoading = false;

self.postdata = postdata;

3.12

self.checkedData = [

{name: "roles", value: "VP of Technology", checked: true},

{name: "roles", value: "CFO", checked: true},

{name: "roles", value: "CMO", checked: true},

{name: "roles", value: "COO", checked: true},

{name: "roles", value: "Content and Quality", checked: true},

{name: "roles", value: "BDM", checked: false},

{name: "roles", value: "HR", checked: false},

{name: "roles", value: "HR Director", checked: false},

{name: "roles", value: "Trainers", checked: false},

{name: "roles", value: "Training Recruiters", checked: false},

{name: "roles", value: "Training Recruitment Director", checked: false},

{name: "roles", value: "Associate", checked: false}

];

3.13

self.validdata = [];

3.14

function postdata (url, moduleName, checkedData ) {

angular.forEach(self.checkedData, function(value, key){

if(value.checked == true) self.validdata.push(value);

});

3.15

var data = {

url: url,

moduleName: moduleName,

checkedData: self.validdata

};

dataLoading = true;

3.16

$http({

method : 'POST',

data: data,

url : '/moduleregistration'

}).then(function(success) {

self.validdata = [];

if (success.data.result.error != null) {

self.msg = success.data.result.error;

}else {

self.msg = "Post Data Submitted Successfully!";

}

}, function(error) {

self.msg = "Service does not exist";

self.validdata = [];

});

};

});

</script>

</body>

</html>

Head

* 1. – Generic Header Things

These are automatically given when you make an HTML page. You can add more head attributes in here if you need them, like defining the page icon.

* 1. – Bootstrap CDN

This is the CSS file for Bootstrap. This is obtained externally so Spring Security doesn’t freak out and prevent access to your files.

* 1. – jQuery CDN

This is the jQury file necessary to run Angular. This is obtained externally so Spring Security doesn’t freak out and prevent access to your files.

* 1. – Font Awesome CDN

This is the Font Awesome file that provides the icons for modules, and will allow you to use any of the features of Font Awesome. It is obtained externally to allow quick version updates.

* 1. – Angular minimum JavaScript

This script provides the base functionality of angular such as directives, controller functionality, etc.

* 1. – Angular Routes JavaScript

The script used for angular routes allowing url redirection.

* 1. – Style CSS
* The main styling of the user interface.
  1. – modReg CSS

The styling for the module registration element. This includes the header and angular form validation classes.

Body

2.1 – opening Body tag

The body tag defines an angular application “ng-app” and an angular controller “ng-controller”.

2.2 – mainNav

Represents the container that holds the header for the page.

2.3 – navbar header

The header for the page.

2.4 – login container

The container that holds the Salesforce login href and button.

2.5 – href login

The href that directs you to SalesForce.

2.6 – login button

The button that redirects you to SalesForce. Uses the angular ng-click directive that calls toSalesforce() which is defined in the home controller. The toSalesforce() redirects you to SalesForce.

2.7 – modules and module registration

A unorderdedlist of href tags that populates the page with modules and/or a module registration page.

2.8 – Footer

The footer area that contains information related to Revature.

Script

3.1 – Script type

Define the type of script as javaScript.

3.2 – Parasol App Module

Define an angular module called “ParasolApp” and it is dependent on the ngRoute module.

3.3 - $routeProvider Configuration

Configure and register the $routeProvider.

3.4 - $routeProvider Logic

Selects a templateUrl based on a url string such as '/', '/moduleList', and '/moduleReg'. Otherwise there will be a redirection to '/route'.

3.5 – Controller Home

Define a controller called “home’ and inject with scope, $http, $location and $window services.

3.6 – self variable

Assign the $scope to the self variable.

3.7 - self.toSalesforce

Assign the toSalesForce function to the scope variable so that it can be accessed in html.

3.8 - self.modules

Assign the set of module data to the $scope which is referenced by self.modules.

3.9 - self.setHover, self.hover, self.authenticated, self.admin

setHover sets a hover status to the module that is passed in to the function, the hover variable is set to false at startup and changed when the setHover function is called. Self.uthenticated is set false by default and changed if a user is authenticated. It is used to show modules and the module registration page when that user is authenticated. Self.admin is set to false by default and is set to true if a user is an admin.

3.10 - $http 'auth/login'

Ajax get method call that retrieves the modules of a user.

3.11 - self.url, self.moduleName, self.dataLoading, self.postdata

Self.url is the scope variable that holds the url that a user wants to register. Self.moduleName is the module name that a user wants to register. Self.dataLoading is a scope variable that is used so that when a module is being registered no other registration requests can be made. Self.postdata is the scope variable that posts data to the '/moduleregistration' url via ajax.

3.12 - self.checkedData

An array of checkbox data that has two-way binding. It is used so that we can identify which roles have been checked.

3.13 - self.validdata = [];

An empty array that will be used to store checked checkboxes. This data will be sent to the server so that only checked data is processed.

3.14 - postdata (url, moduleName, checkedData )

The function that parses the checkedData array and checks if an element is checked or not. If an array element is checked it will be pushed to the validdata array.

3.15 - var data = {

Prepares data to be passed into an ajax call. It takes the url, moduleName, and validdata array variables and sets inside the data.

3.16 - $http module registration

The ajax POST that sends the url, moduleName, and validdata[] array to the server.