

```

1  var iconImg;
2  var int;
3  var pictures = ["Moti.jpg", "Brandon.jpg", "Benny.jpg", "Yassin.jpg"];
4  var descriptions = ["1", "2", "3", "4"];
5  var index = 0;
6  function showImage(){
7      iconImg.setAttribute("src", pictures[index]);
8      iconImg.setAttribute("alt", descriptions[index]);
9      index = (index + 1)%pictures.length;
10 }
11 function startImg(){
12     iconImg = document.getElementById( "imageSet" );
13     int = setInterval(function(){showImage()},2000);
14 }
15
16 function clearImg() {
17     iconImg.setAttribute("src", "gophers-mascot.png");
18     iconImg.setAttribute("alt", "Go Gophers");
19 }
20
21 var isOtherOption = false;
22 var map;
23 var myLatLng;
24 var service;
25 var directionsService;
26 var directionsRenderer;
27 var markerContacts = [];
28 var infoContacts = [];
29 function initMap(){
30     myLatLng = {lat:44.9727, lng:-93.23540000000003};
31
32     map = new google.maps.Map(document.getElementById('map'), {zoom: 13.2, center:
33     • myLatLng});
34     service = new google.maps.places.PlacesService(map);
35     directionsService = new google.maps.DirectionsService();
36     directionsRenderer = new google.maps.DirectionsRenderer();
37     directionsRenderer.setMap(map);
38     directionsRenderer.setPanel(document.getElementById('directionsPanel'));
39     var inputFrom = document.getElementById('from');
40     var inputTo = document.getElementById('to');
41     var autocomplete1 = new google.maps.places.Autocomplete(inputFrom);
42     var autocomplete2 = new google.maps.places.Autocomplete(inputTo);
43
44
45
46     var Place = function(address, name, disc) {
47         this.address = address;
48         this.name = name;
49         this.disc = disc;
50     };

```

```

51     var allPlaces = new Array;
52     var infowindow = new google.maps.InfoWindow();
53
54     //Add Location information to allPlaces
55     var names = ['<p><b>Moti</b></p>', '<p><b>Benny and Yassin</b></p>',
    • '<p><b>Benny and Yassin</b></p>', '<p><b>Brandon</b></p>'];
56     var discs = ['<p>The Radius at 15th</p>', '<p>Classic City Apt.</p>',
    • '<p>Classic City Apt.</p>', '<p>The Quad on Delaware</p>']
57     for(var i = 1; i<5;i++){
58         var address =
    • document.getElementById('contactTable').rows[i].cells[2].innerHTML;
59         var name = names[i-1];
60         var disc = discs[i-1]
61
62         allPlaces.push(new Place(address, name, disc));
63     }
64
65     //Loop over allPlaces, derived from https://developers.google.com/maps/
    • documentation/javascript/geocoding
66
67     for(var i = 0; i< allPlaces.length; i++){
68         createMarker(i, allPlaces, map, infowindow);
69     }
70 }
71 function createMarker(i, allPlaces, map, infowindow){
72     var geocoder = new google.maps.Geocoder();
73     geocoder.geocode({ 'address': allPlaces[i].address}, function(results, status)
    • {
74         if (status == 'OK') {
75             var marker = new google.maps.Marker({
76                 map: map,
77                 position: results[0].geometry.location,
78                 icon: {url: "icon.png", scaledSize: new google.maps.Size(50, 50)}
79             });
80             google.maps.event.addListener(marker, 'click', function(info) {
81                 infowindow.setContent(allPlaces[i].name + "\n" + allPlaces[i].disc);
82                 infowindow.open(map, this);
83             });
84             google.maps.event.addListener(map, "click", function(event) {
85                 infowindow.close();
86             });
87             markerContacts.push(marker);
88             infoContacts.push(infowindow);
89         } else {
90             alert('Geocode was not successful: ' + status);
91         }
92     });
93
94 }
95 function PlaceOption(val){
96     var element = document.getElementById("otherOption");

```

```

97     if(val=='other'){
98         element.disabled=false;
99         isOtherOption=true;
100     } else{
101         element.disabled=true;
102         isOtherOption=false;
103     }
104 }
105
106
107 function findNearby(){
108     var place;
109     var request;
110     var radius = parseInt(document.getElementById("radius").value);
111     if(isOtherOption) {
112         place = document.getElementById("otherOption").value;
113         place = place.toLowerCase();
114     } else {
115         place = document.getElementById("myList").value;
116     }
117     service = new google.maps.places.PlacesService(map);
118     request = {location: myLatLng, radius: radius, type: [place]};
119     // Perform a nearby search.
120     service.nearbySearch({location: myLatLng, radius: radius, type: [place]},
        • function(results, status) {
121 if (status !== 'OK'){
122     return;
123 }
124 createMarkers(results);
125 });
126 for (var i = 0; i < markerContacts.length; i++){
127     markerContacts[i].setMap(null);
128     infoContacts[i].close();
129 }
130 }
131
132
133
134 function createMarkers(places) {
135     var bounds = new google.maps.LatLngBounds();
136     var infowindow = new google.maps.InfoWindow();
137
138     for (var i = 0, place; place = places[i]; i++) {
139         var str = "<p><b>" + place.name + "</p></b>" + "<p>" +
        • place.geometry.location + "</p>";
140
141         var marker = new google.maps.Marker({
142             map: map,
143             title: place.name,
144             html: str,
145             icon: {url: place.icon, primaryColor: "#0000FF", scaledSize: new

```

```

    •      google.maps.Size(30, 30)},
146      position: place.geometry.location,
147      address: place.address_component,
148  });
149  google.maps.event.addListener(marker, 'mouseover', function(info) {
150      infowindow.setContent(this.html);
151      infowindow.open(map, this);
152  });
153  google.maps.event.addListener(map, "mouseout", function(event) {
154      infowindow.close();
155  });
156  markerContacts.push(marker);
157  infoContacts.push(infowindow);
158  }
159  }
160  function directions() {
161      var from = document.getElementById('from').value;
162      var to = document.getElementById('to').value;
163      var elem = document.getElementsByName('dirType');
164      var travelMode;
165
166      for(i = 0; i < elem.length; i++) {
167          if(elem[i].checked) {
168              travelMode = elem[i].value;
169          }
170      }
171      for (var i = 0; i < markerContacts.length; i++){
172          markerContacts[i].setMap(null);
173          infoContacts[i].close();
174      }
175      var request = {
176          origin: from,
177          destination: to,
178          travelMode: travelMode
179      };
180      directionsService.route(request, function(result, status) {
181          if (status == 'OK') {
182              directionsRenderer.setDirections(result);
183          }
184      });

```