

# Google for Work Partners

If you are a Google for Work partner working towards your credential, you should complete this project and submit it for evaluation using the instructions provided in your Maps Developer Learning Path.

Udacity students should feel free to complete this optional project but it should not be submitted for evaluation using the Udacity Reviews system and will not be evaluated.

## **Project Description**

In order to further demonstrate the concepts and functionality encountered in the Google Maps APIs, we will add several features into the real-estate listing site you have built throughout the course. We've listed out the features that you must add to your site, which we built DURING the course, as well as additional features to take your practice to the next level. You may need to reference the Google Maps API documentation. You are free to embellish upon and add to the website in other ways to enhance the user experience - the

example presented is only meant to enhance your understanding of the requirements and expectations for API usage.

## **How do I Complete this Project?**

Starting with the application (referred to as a “real estate listings site”) that we have demonstrated building throughout the course, add on features and update code as defined in the rubric. When each item has been completed, you have completed the project.

## **Evaluation**

Criteria	Requirements
<b>Built with instructor during course:</b>	
Loading a map	<ul style="list-style-type: none"> <li>• Load a Google Map in your site using the JavaScript API</li> </ul>
	<ul style="list-style-type: none"> <li>◦ The JavaScript API should be loaded asynchronously</li> <li>◦ Use your API Key</li> <li>◦ Load the latest release version of the API</li> </ul>
Adding Markers to the Map	<ul style="list-style-type: none"> <li>• Create an array of markers               <ul style="list-style-type: none"> <li>◦ Markers should have a unique title</li> <li>◦ Markers should have a custom icon</li> <li>◦ Markers should change color when moused over</li> </ul> </li> <li>• Allow the user to display and hide all the markers on the map               <ul style="list-style-type: none"> <li>◦ Hiding them should not delete them</li> </ul> </li> </ul>
Displaying data in Infowindows	<ul style="list-style-type: none"> <li>• An Infowindows should appear on each listing marker when that marker is clicked</li> <li>• Do not allow more than one infowindow to be opened at once</li> <li>• Infowindows should contain:               <ul style="list-style-type: none"> <li>◦ The title of the listing</li> <li>◦ The street view panorama of the listing (see below)</li> <li>◦ Other information at your discretion!</li> </ul> </li> <li>• The infowindow should not be "recreated" if it's already open on a marker</li> </ul>
Creating a street view panorama	<ul style="list-style-type: none"> <li>• Create a street view panorama of each marker's location               <ul style="list-style-type: none"> <li>◦ The panorama should only be created when the marker is clicked</li> <li>◦ It should be displayed in the infowindow</li> <li>◦ The street view should appear facing the address of the listing and looking slightly up</li> </ul> </li> </ul>
Using the Drawing and Geometry Libraries	<ul style="list-style-type: none"> <li>• Display the drawing controls on the map, allowing polygons to be drawn</li> <li>• Allow the user to hide or display the drawing controls               <ul style="list-style-type: none"> <li>◦ If the controls are hidden, also clear any shapes drawn</li> </ul> </li> <li>• Allow the user to draw an editable polygon on the map</li> <li>• Allow the user to search for listing within the polygon               <ul style="list-style-type: none"> <li>◦ When a polygon is drawn, display any markers within the polygon and hide any outside of it</li> <li>◦ If the polygon is edited, the search should be re-done</li> </ul> </li> </ul>
Styling the Map	<ul style="list-style-type: none"> <li>• Apply custom styles to the map on initialize</li> <li>• Remove the Map Type controls from the map</li> </ul>
Using the Geocoding service in the JS API	<ul style="list-style-type: none"> <li>• Allow the user to enter a location into the browser</li> <li>• On a user action, geocode the location to get a lat/lng               <ul style="list-style-type: none"> <li>◦ Bias the geocoding results to the city you are focusing on</li> </ul> </li> <li>• Update the center and zoom of the map to focus on that location</li> </ul>

Using the Distance Matrix service in the JS API	<ul style="list-style-type: none"> <li>• Allow the user to enter a location, travel mode, and maximum travel time in minutes</li> <li>• Display only the listing markers that have a journey time from the user entered point, that is within the entered time maximum, using the entered travel mode</li> <li>• Hide the listing markers that are outside of that journey time</li> <li>• Display the journey time in minutes and the distance in miles above each marker, in a small <u>infowindow</u> <ul style="list-style-type: none"> <li>◦ The <u>infowindow</u> should close if the marker is clicked</li> </ul> </li> </ul>
Using the Directions service in the JS API	<ul style="list-style-type: none"> <li>• Expand the <u>infowindow</u> from the above example to include a "Show Route" button</li> <li>• Allow the user to show the route on the map from the marker to the user-entered point, using the entered travel mode</li> </ul>
Using Autocomplete on a text input	<ul style="list-style-type: none"> <li>• Update all text input options to use Places autocomplete</li> <li>• Bias the autocomplete to within the map area</li> </ul>
Using <u>SearchBox</u> on a text input	<ul style="list-style-type: none"> <li>• Add a Places <u>SearchBox</u> capability to the site, taking in a user-entered value</li> <li>• When a query or places is selected, execute a places search</li> <li>• Bias the places search to within the map area</li> <li>• Show the resulting place markers on the map <ul style="list-style-type: none"> <li>◦ Use the returned place icons as the icon</li> <li>◦ Use the Places Name as the title</li> </ul> </li> </ul>
Using Places Textsearch	<ul style="list-style-type: none"> <li>• Expand the <u>SearchBox</u> capability so that, if a user does not select a value from the picklist, the user can click a button to execute a Places <u>Textsearch</u> for places</li> <li>• Bias the places search to within the map area</li> <li>• Show the resulting place markers on the map <ul style="list-style-type: none"> <li>◦ Use the returned place icons as the icon</li> <li>◦ Use the Places Name as the title</li> </ul> </li> </ul>
Making Place Details requests	<ul style="list-style-type: none"> <li>• Expand the previous example to execute a place details search whenever a place marker is selected, using the place ID.</li> <li>• Show these details: <ul style="list-style-type: none"> <li>◦ Place name</li> <li>◦ Formatted Address</li> <li>◦ Telephone number</li> <li>◦ Operating days/hours</li> <li>◦ The first photo in the result</li> </ul> </li> <li>• Populate the place details into an <u>infowindow</u>. <ul style="list-style-type: none"> <li>◦ Only allow one place details infowindow to be open at once</li> <li>◦ Do not re-execute a place details request if a marker is</li> </ul> </li> </ul>
	clicked, with the <u>infowindow</u> already open on it
<b>Advanced Features: Not built with instructor during course</b>	
Traffic Layer	Add a toggleable <a href="#">traffic layer</a> to the map.

Styled Map type	Name the style created in the in course project and have it as toggleable <a href="#">styled maptype</a> option
Directions Display	<ul style="list-style-type: none"> <li>• In addition to having the route show on the map, show the step-by-step directions when the user selects "show route"</li> <li>• Use the directions renderer already created, and create a panel on the right-hand side of the screen.</li> <li>• Allow the user to re-set the page without refreshing by removing the directions panel and route from the map.</li> </ul>
Places Photos capabilities	<p>Allow the user to scroll through multiple place PHOTOS in the place details <a href="#">infowindow</a> which appears</p> <ul style="list-style-type: none"> <li>• You can use "previous" and "next" icons, buttons, links - anything which allows the user to select to go forward and backward</li> <li>• Use only the photo reference IDs returned with the place details request</li> <li>• You'll need to store the place details information returned with the request, in order to allow the user to go through the array of place photos returned</li> <li>• If there is only one place photo, hide the "previous" and "next" capability.</li> </ul>

## Links