CSCI 4131 Spring 2020
Internet Programming Assignment 3

Due Date: Friday, February 21st at 2:30 pm (afternoon)

Late Submissions accepted after the Due Date (above) until: 6 pm (Evening) Saturday, February 22nd – **with penalty**

1 Description

For this assignment, you will add Google Maps to the functionality you have developed on your Contacts and Form Web pages through homework 2, enhance the styling of your Widgets Page, (CSS / HTML), and eliminate errors. There are 11 pages in this assignment description.

The objective of this assignment is to continue to develop your JavaScript skills, introduce you to the Google Maps JavaScript application programming interface (API), the Google Places JavaScript library, the Google Maps Directions Service, and geolocation. While the Google Maps API will be introduced in subsequent lectures, developing and/or bolstering the ability to read library documentation and then use it to develop functionality is an essential skill for today's web developer (or any software developer). New libraries with new APIs are introduced, and existing libraries and their APIs are updated, on a regular basis. Teaching a specific API is counterproductive in such an environment. Instead, one of the objectives of this assignment is to motivate you to develop and/or bolster your skills to learn and use new libraries and APIs.

You will update your Contacts page as follows:

1. You will write code to dynamically mark the addresses of your contacts specified in the web page containing contacts list table contact address on an embedded Google map.
2. You will develop additional functionality to search for specific places near your current location and display them on the map.
3. You will also add functionality to calculate directions and display a route between your current location and destination on your Google map via various modes of transportation.
4. And, finally, you will replace the large picture displayed on your Contacts page in HW 2 with a slide-show capability that enables you to display a slide show of all the icons of your contacts.

You will update your form page as follows:

- You will add a Google map on which you can select a location on the Google map and use the click on point of interest capability to fill in the address field on your form AND/OR use the autocomplete functionality from Google Places to autocomplete the address field on your form. (YOU CAN DO ONE OR BOTH OF THESE)

Finally, if you have not done so already, you will restyle the widgets on your widgets page, so they are better positioned and error-free (so you will have to fix any errors due to the embedded widgets you obtained from YouTube, Twitter, and Instagram).

2 Preparation/Reference

Sign up for the Google Maps JavaScript API – see:

https://developers.google.com/maps/documentation/javascript/get-api-key

Note, you need a credit or debit card to sign up for the Google Maps Application Programmer Interface The uses required by this course will never necessitate a charge for use of Google Maps – it is only required should you deploy your Maps to a production website. See the instructor if you have questions or concerns about this.

2.1 Google API Setup

Some setup is required to use the Maps API – you need to get your API key (see above). Google provides an excellent tutorial for obtaining your API key and setting up your map, and it is recommended to complete the tutorial. The tutorial can be found at the following link:

https://developers.google.com/maps/documentation/javascript/tutorial

When signing up for your API key, use your UMN x500 account

2.2 Google Places JavaScript Library Reference

You can refer the following link to obtain more information about places library:

https://developers.google.com/maps/documentation/javascript/places

2.3 Google Maps Directions Service Reference

You can refer the following link to obtain more information about direction service:

https://developers.google.com/maps/documentation/javascript/directions

2.4 Geolocation Reference

You can refer the following link to obtain more information about geolocation:

https://developers.google.com/maps/documentation/javascript/geolocation

2.6 **Google Click on Points of Interest (used to fill address field on Form when points of interest are selected/clicked on the map next to it):**

https://developers.google.com/maps/documentation/javascript/examples/event-poi

3 Functionality

Update functionality on your Contacts page

1. Remove the functionality that displays a large picture of the last icon that a user hovers over with their selection device, and replace it with a slide show "widget" that you develop using JavaScript (***DO NOT USE jQuery do implement this functionality***). The slide show "widget" you develop should be placed next to the Contacts table as illustrated in figures 4.1 and 4.2 below.

2. Start by modifying the address column of the table from homework 2: include the complete address information for every Contact in table entries, e.g., 100 Union St. SE, Minneapolis. MN 55455

3. Embed a Google map under the table.

4. The map should be centered on University coordinates (44.9727, -93.23540000000003) with zoom level of 14 (or any zoom level that you find appropriate). University coordinates can also be obtained using the following link: https://www.gps-coordinates.net/

5. You should write JavaScript code that ***dynamically*** extracts the names and addresses of your contacts from the DOM objects that are in your Contacts table. Your code should then place a custom marker on the map for each location extracted. The markers should display the name of the contact and their building name or room number at the address(e.g. Room number 2-209 in Keller or Shepherd 383) upon being clicked**. Note, do not use hard-coded latitude and longitude positions to do this**. More specifically, create JavaScript to obtain a list of addresses from your contacts and use the geocoding or places library to obtain a latitude and longitude for each of them. Then create a marker for each unique latitude and longitude. This can be achieved using an information window (see figures 4.3, 4.4, and 4.5 below).

6. The next step is to insert a form/input area on the right of the map. The elements in this form/input area will require you to use 'Google Places JavaScript library' and 'Google Maps Directions Service'. See figure 4.6 below.

7. The first row of input area allows user to search places, e.g., restaurant near the user's current geolocation. It consists of:
    - A drop down field that lists the category of places to search for. The default categories for this homework are restaurant, hospital, parking, supermarket. You can replace these categories with additional categories present at: https://developers.google.com/places/supported_types. You should ensure that at least a few places exist in the search results for each of the categories you include.
    - A text box that specifies the numeric radius around your current location in which the search results will preferably be shown.
    - A textbox that allows users to enter the keywords of other places that they may search for. This textbook is by disabled. The drop down field should include an extra option named : "Other". When "Other" is selected, the textbox is enabled.

Try one of the following keywords (or another keyword of your choice) when you are testing this - burger, bus, library, laundromat etc.

- A button labelled 'Search: Upon clicking this button, your code finds all the nearby places within the specified radius using 'Google Places JavaScript library' and displays the results by placing a marker on the map for each of the found places. The old markers (e.g. the markers for your event places, and markers for any other category) should be cleared before placing new markers on the map. When mousing over a marker, the name of the place should be displayed.
- Refer to figures 4.6 and 4.7 for an illustration of this functionality.

8. The second search component that should be added is the functionality to get directions from a user's current location (as determined by HTML Geolocation – see https://www.w3schools.com/html/html5_geolocation.asp) to the destination they provide. The group of elements you should use to implement this functionality are as follows:
    - A textbox that allows the user to input the destination location. The source location will be the current location of the user and it can be obtained using geolocation.
    - A radio button group: Three modes of travel should be specified: DRIVING, WALKING, and TRANSIT. One of these must be selected at all times.
    - A button labelled 'Go: Upon clicking this button, the route between the current location and the destination should be displayed on the map. The displayed route will be based on the mode of travel selected by the user. The directions associated with this route should be displayed on a scrollable side panel floating to the left of the map. Directions involving higher number of instructions should be wrapped into this fixed dimension panel with the scrollable feature. The source for the directions is the user's current location extracted using Google geolocation services. The source coordinates should not be hard-coded, they should be extracted dynamically using code (JavaScript). Make use of 'Google Maps Directions Service' for this functionality.
    - Refer to figures 4.8 and 4.9 for an illustration of this functionality

Update Functionality on your Forms page

a.) Add a Google Map to the right of your form

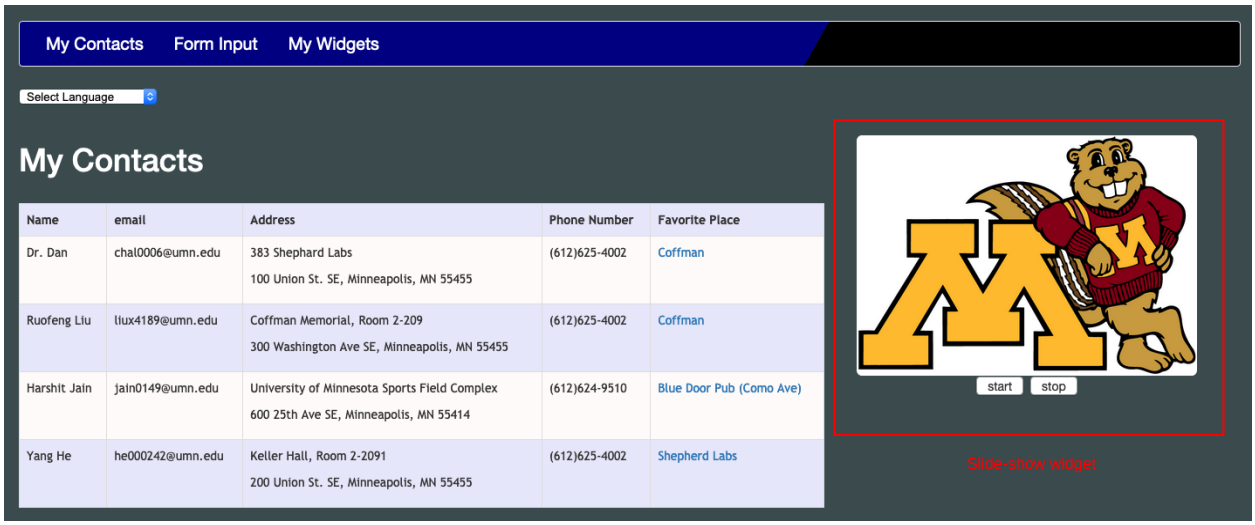
b.) Use Google Map's autocomplete functionality to fill in the address field on the form from locations selected from the Google map to the right of your form AND/ OR autofill the address field with a click on a point of interest on the map. See Figures 4.10 and 4.11 below for an example of click on POI functionality. See the code snippet on Page 9 to help you incorporate autofill functionality

Update the Style on your widgets page and get rid of the errors caused by the embedded widgets obtained from YouTube, Instagram, and Twitter.

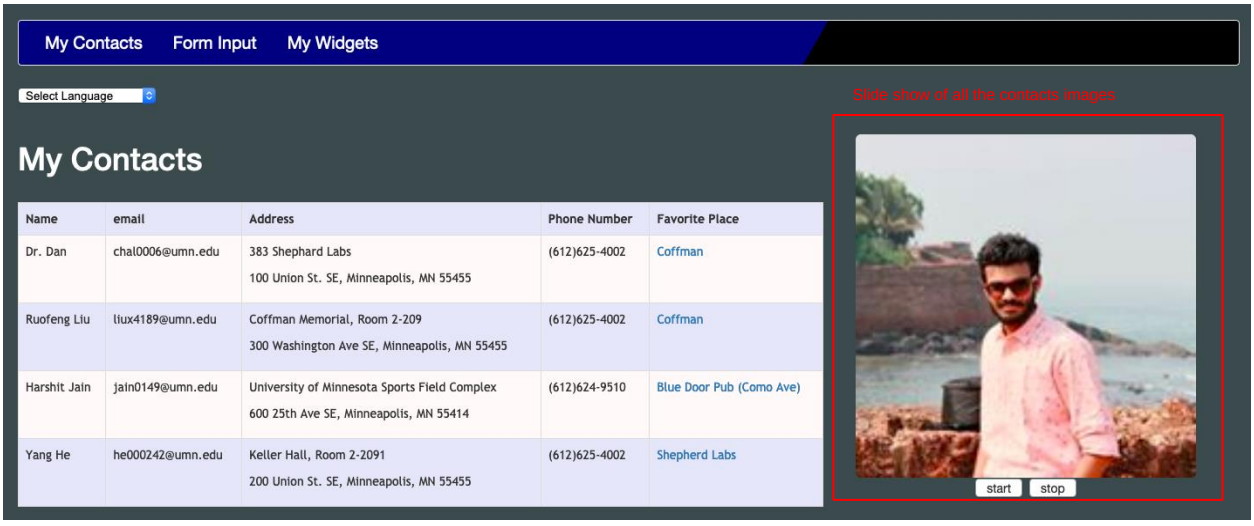
See figure 4.12 below for an improvement on styling your widgets page (if you have not already done so).

4 Screenshots

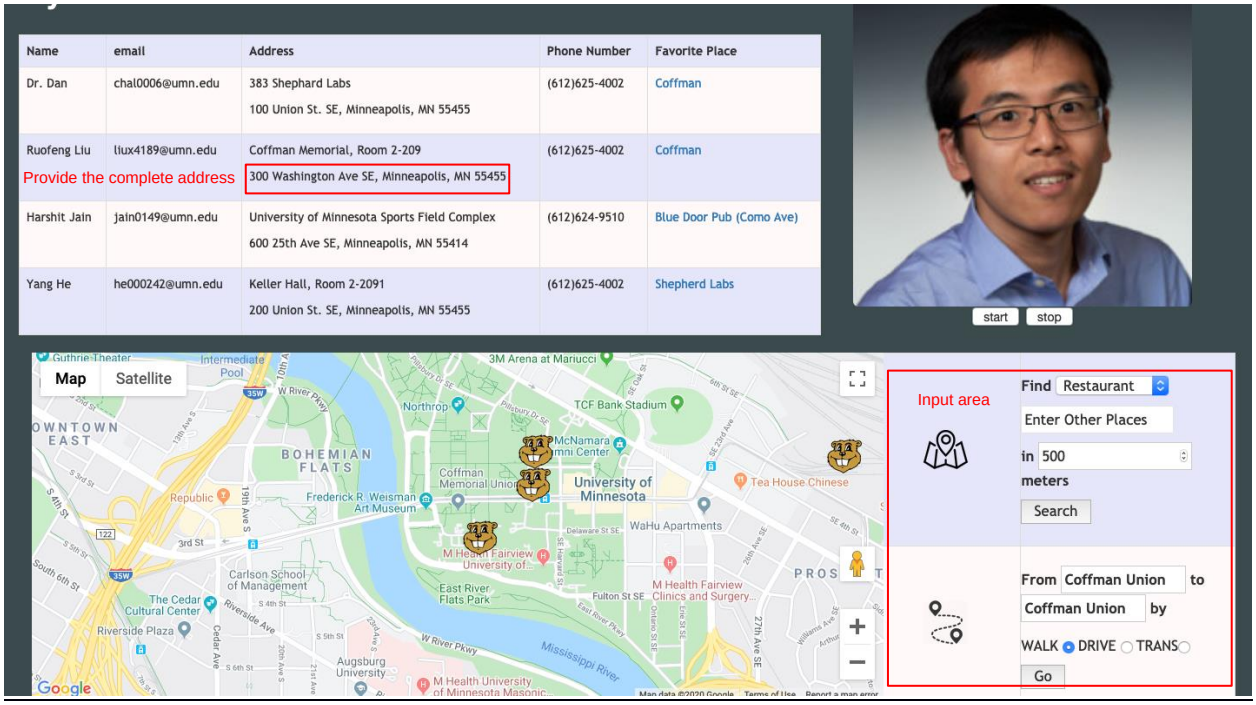
4.1 Add slide show "widget" to the right of the Contacts.



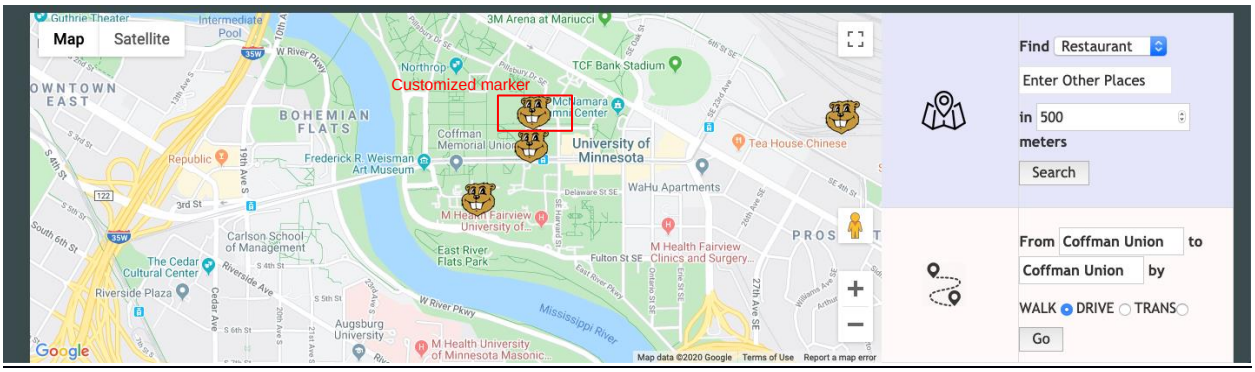
4.2 The slide show widget should display the unique images shown as thumbnail pictures of your Contacts once in any linear order you choose. Each thumbnail should be shown only once in the display sequence, and the display sequence should start over beginning with the first picture of the sequence after the last picture in the display sequence is shown. The display sequence should be initiated with the start button, and be halted with the stop button. Once started, the display sequence should continue until it is halted with the stop button – and it can be restarted with the start button. Each picture should be displayed for 2 seconds before the slide show widget displays the next picture in the display sequence.



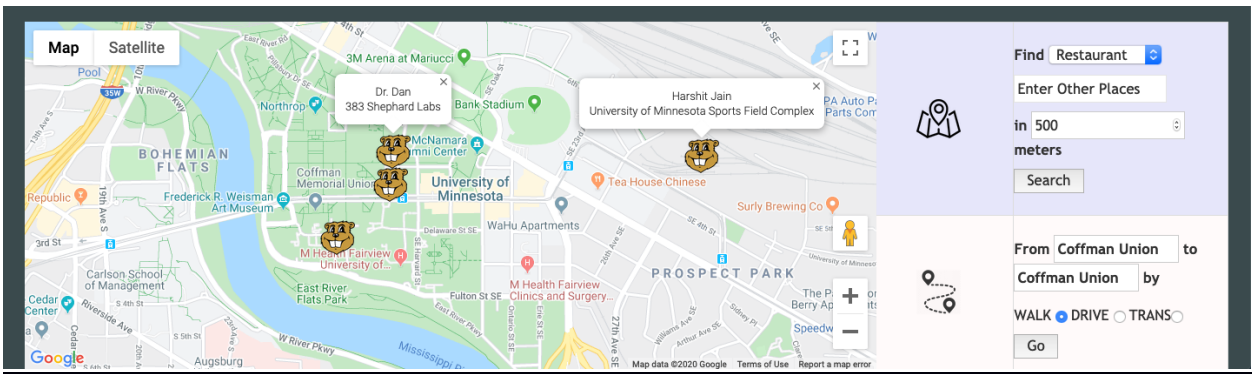
4.3 Table with a list of contacts, google map and form/input area floating to the right of the map



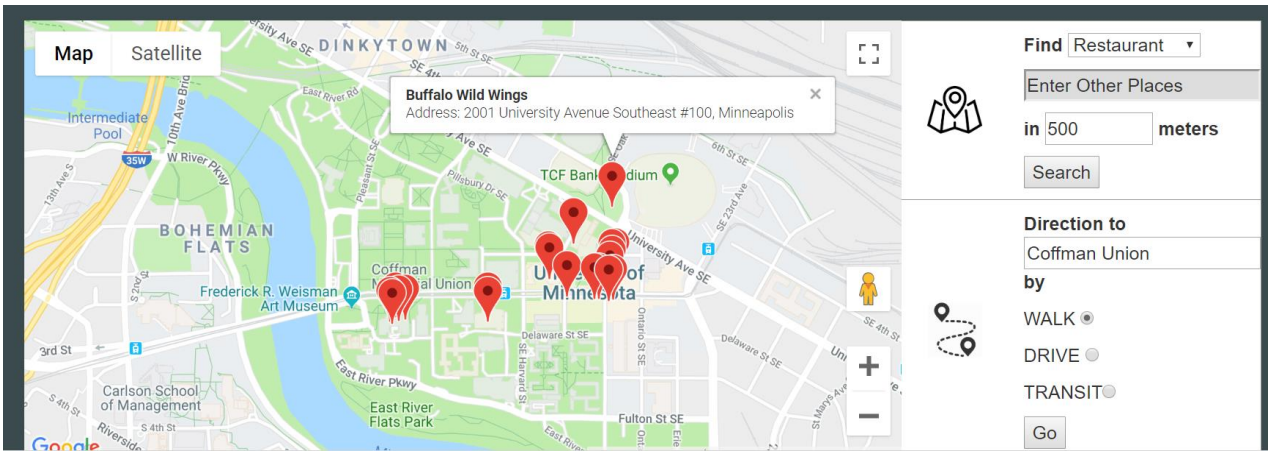
4.4 Map with the contact address from the table (you should use custom icons for markers)



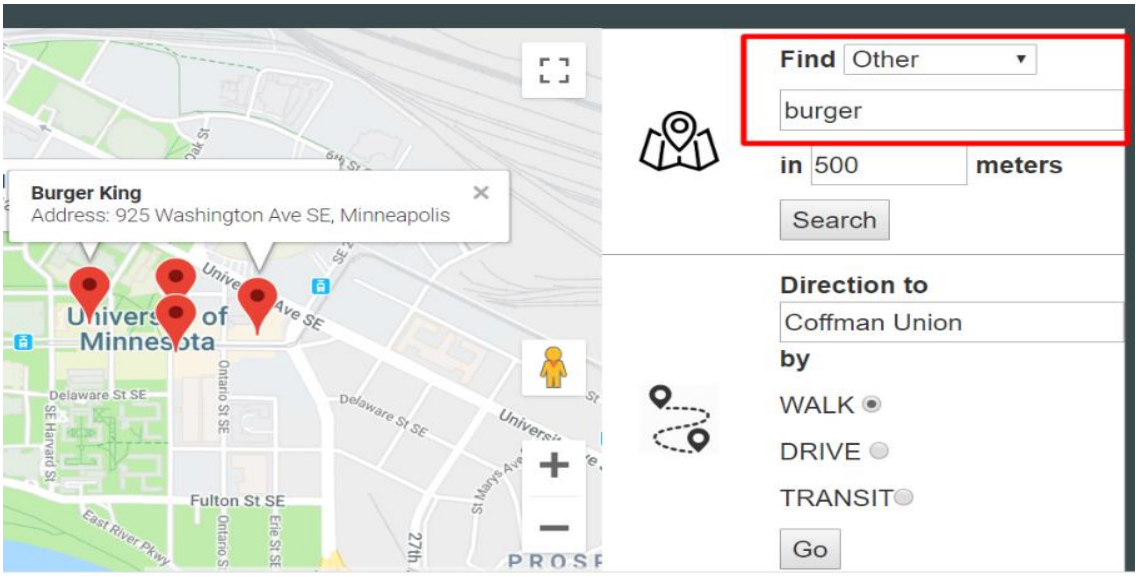
4.5 Marker shows the name of the contact and the place on click



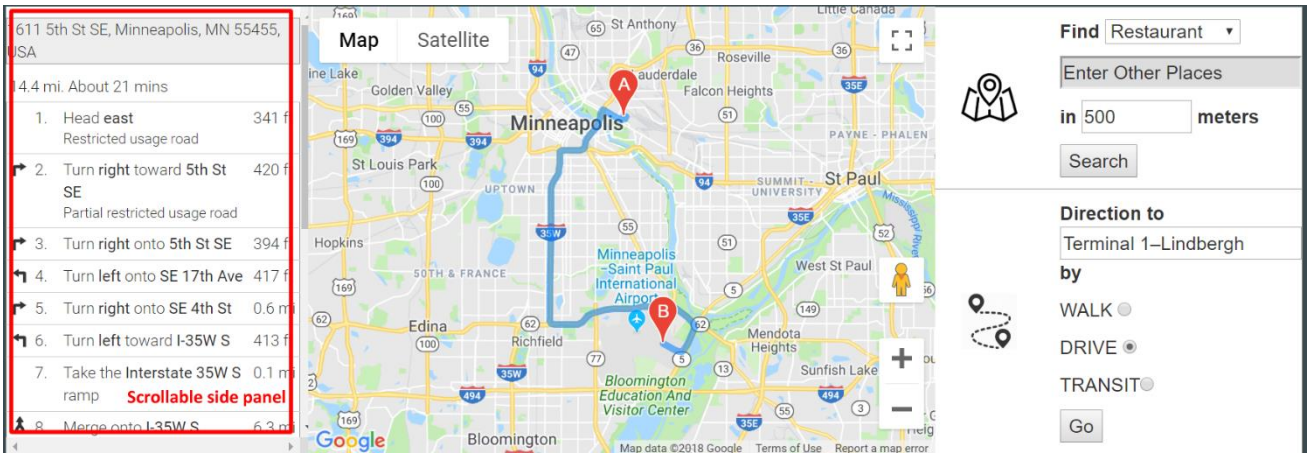
4.6 Map showing the result of nearby restaurants in a radius of 500 meters



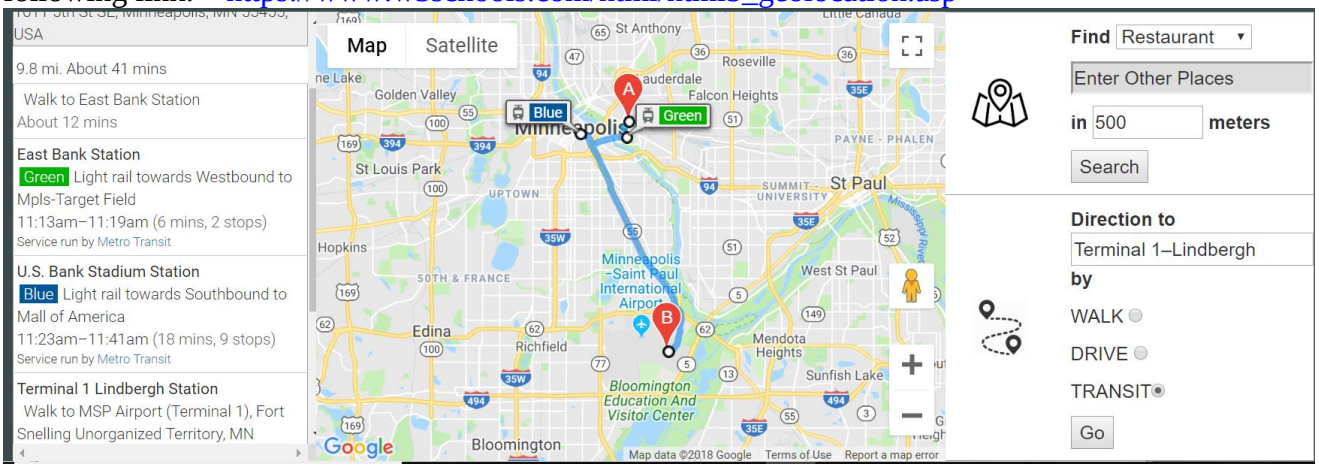
4.7 When “Other” is selected in the dropdown list, user can enter the keyword to search for items related to places they are looking for, e.g., “burger”. The nearby places with the keyword in their names will show up when “search” button is clicked (See the results for “burger” below)



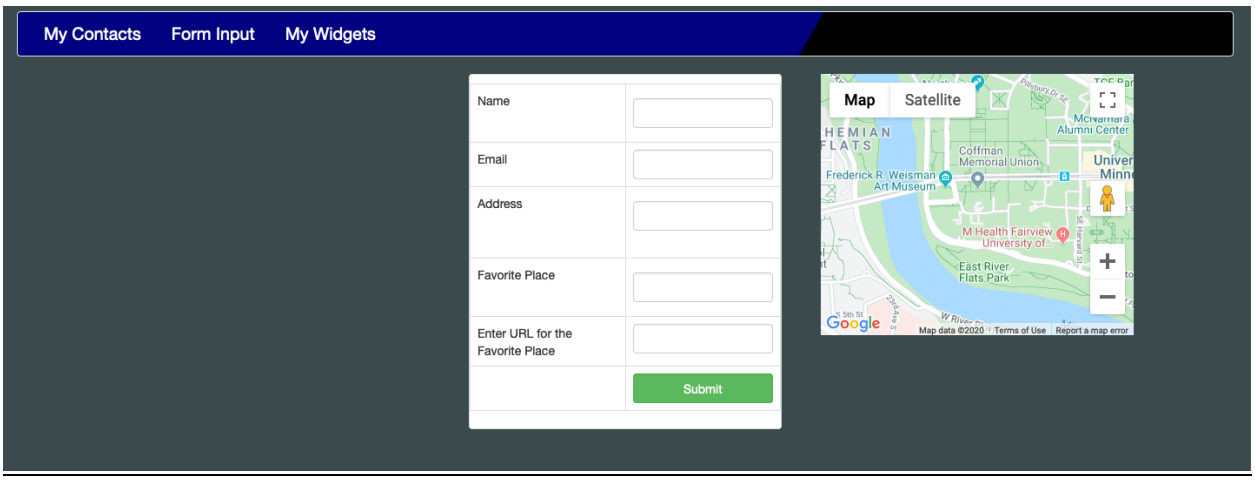
4.8 Map showing the DRIVING route between the current location and Terminal 1–Lindbergh



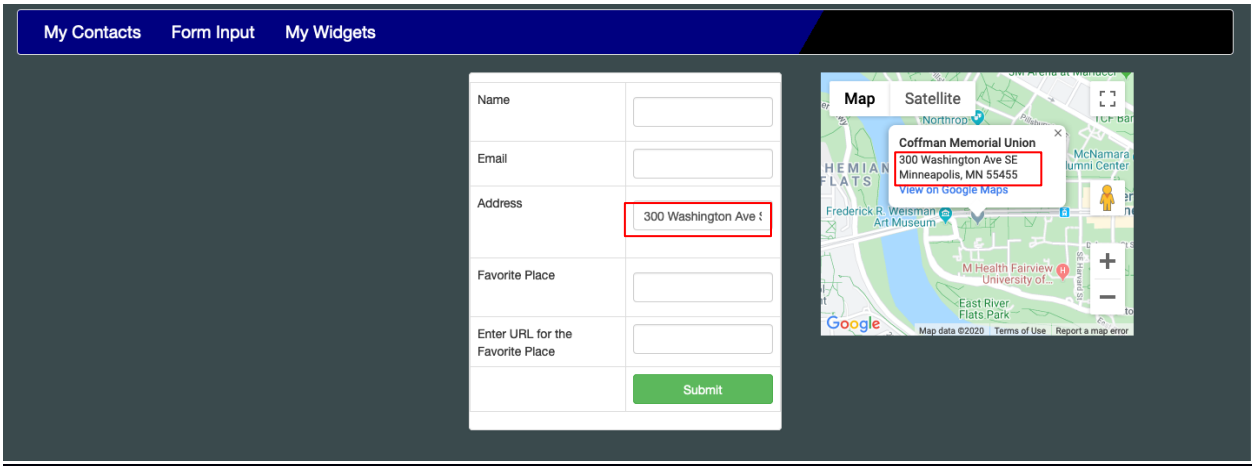
4.9 Map showing the TRANSIT route between the current location and Terminal 1–Lindbergh
Note: you can use HTML5 Geolocation to set your current location – see the information at the following link: https://www.w3schools.com/html/html5_geolocation.asp



4.10 Add a Google Map to your Form Page



4.11 When a place is selected on the Map, the address field on your form should be populated automatically with the address of the place you selected for click on POI functionality.



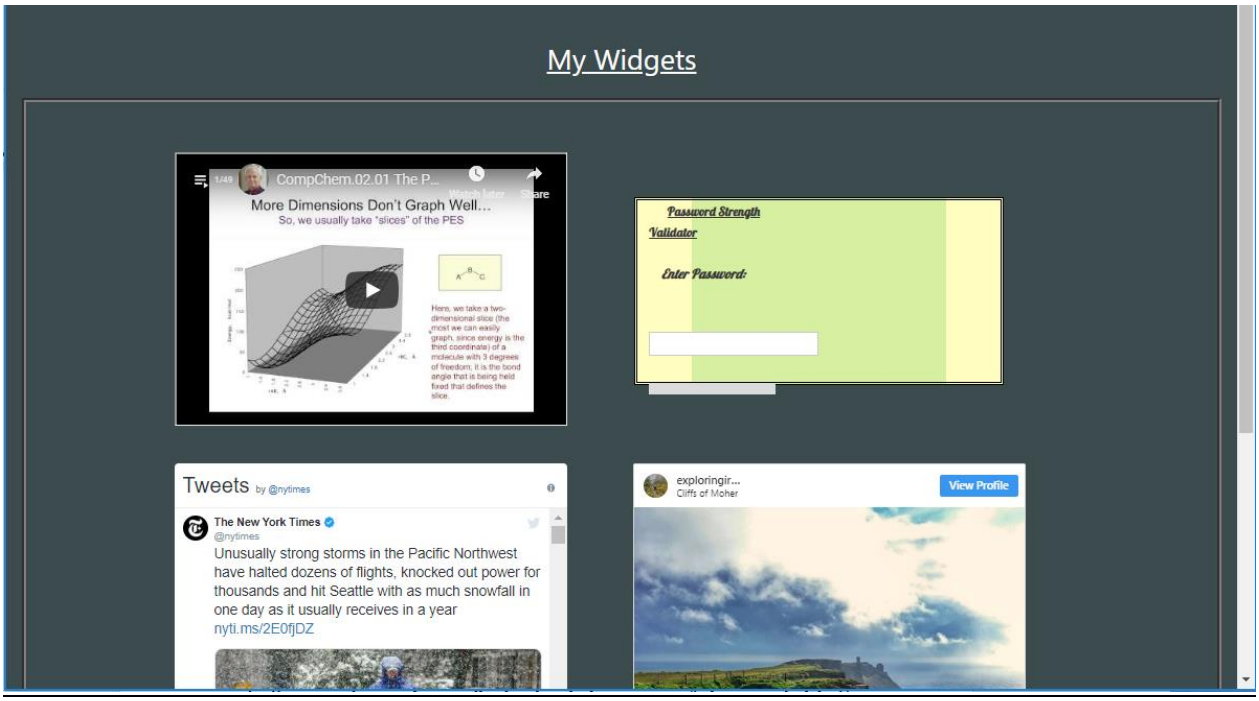
The "two lines" below can be used to help incorporate Google Maps Auto-complete functionality into the address text field on your form.

```
<input id="addressTextField" type="text" size="50">
   <script>
      function init() {
          var input =      I
                   document.getElementById('addressTextField');
          var autocomplete =
                   new google.maps.places.Autocomplete(input);
      }
   <script>
```

For “click on POI functionality”, you will need to add the JavaScript API (with your key) to your forms page to include another Google map on the page. See the example at the link specified in section 2.6 on page one of this document :

```
<script type="text/javascript"
src="https://maps.googleapis.com/maps/api/js?key=YOUR-GOOGLE-MAPS-KEY=places&callback=initMap"></script>
```

4.12 Improve the styling on your widgets page, and get rid of HTML and CSS errors caused by the Instagram, YouTube, and Twitter widgets (check to ensure you have done that with the HTML and CSS validators).



5 Submission Instructions

1. For this assignment, you are expected to separate the HTML, CSS, and JavaScript files.
2. Your submission should include a ***minimum*** of 6 files (4 HTML, and 1 each for JavaScript and CSS). You can include more than one JavaScript and/or CSS file. However, it is required that the CSS is separated from HTML.
3. Zip these files and the name of the zipped folder should be **yourx500id_hwk03**. *PLEASE ENSURE TO TEST YOUR CODE ON CSE LAB MACHINES.*

6 Evaluation

Your submission will be graded out of 105 points on the following items:

1. ALL HTML files you provide files w3schools validators (http://validator.w3.org) without errors. Warnings are accepted (**5 points**).
2. All CSS files pass w3schools validator (http://jigsaw.w3.org/css-validator/) without errors. Warnings are accepted (**5 points**).
3. Slide show widget is placed on the Contacts page next to the Contacts with proper alignment as shown in the pictures, and functions correctly. **(10 points)**
4. Google Map is placed on webpage below the Contacts with proper alignment as shown in pictures (**10 points**).

5. Custom markers are dynamically placed on the locations specified in your Contacts. You must dynamically obtain the addresses from your Contacts and use the Google Maps Geocoding or Places functionality to create markers and place them on the map. (**10 points**).
6. When the markers created in step 5 above a selected (via a click), the markers display the address (**5 points**).
7. Nearby places can be searched within a specific radius and corresponding markers are created on map **(10 points)**.
8. Upon mouseover of the markers placed on map in response to a place search, markers display the name of the nearby place (**5 points**).
9. All previously displayed markers are removed before the results of a new search are displayed **(5 points)**.
10. Correctly working search capability to find and mark places on the Google Map. **(10 points)**
11. Correctly working functionality for finding directions to a custom destination from the user's current location.
    - Accept destination through input text box and displaying directions on side panel (**5 points**).
    - Functioning capability to switch between multiple travel modes (**2 points**).
    - Style and alignment of the side panel with scroll feature as shown in picture (**3 points**).
12. The form page is modified to include a Google Map. **(10 points)**
13. On the forms page, when a location on the Google Map is selected, it will automatically populate the address field of the form. **(5 points)**
14. **Enable autocomplete functionality on the address field of the form (5 points)**
15. All the files required for your solution should be packaged in a tar or zip file, and submitted via the link on the Moodle class site. The name of the zipped file should be **yourx500id_hwk03.zip**. Failure to do this correctly may result in a deduction ranging from 10 points up to full credit for the assignment.

Note, it is possible to get 105 points on this assignment (5 bonus points)