CSCI 4131 Fall 2018
Internet Programming Assignment 3

Version 2, posted 10/8

Due Date: Friday October 12th at 2pm (afternoon)

Late Submissions accepted until: 2 am (Morning) Saturday October 13 – with penalty

1 Description

For this assignment, you will add google maps to the functionality you have developed on your Webpages through homework 2. You will write code to dynamically mark your event places on the embedded google map. You will develop additional functionality to search for specific places near your current location. In addition to this, you will be asked to calculate directions and display route between your current location and destination using various modes of transportation.

The objective of this assignment is to continue to develop your JavaScript skills, introduce you to 'Google Maps JavaScript API ', 'Google Places JavaScript library', 'Google Maps Directions Service', geolocation, and Google translate API. 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.

2 Preparation/Reference

Sign up for google maps JavaScript API – see:

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

2.1 Google API Setup

A small amount of setup is required to use the Maps API – you need to get your API key. 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

2.5 Google Translate API

https://www.w3schools.com/howto/howto_google_translate.asp

3 Functionality

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

2. Embed a google map under the table. You can remove the image which was earlier present under the table.

3. 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/

4. You should write JavaScript code that ***dynamically*** extract the names and addresses of your events from the DOM objects that are in your schedule table. Your code should than place a custom marker on the map for each location extracted. The markers should display the name of the place upon being clicked. Note, do not use hardcoded latitude and longitude positions to do this. Use JavaScript to obtain a list of addresses from your calendar and use the geocoding or places library to obtain a latitude and longitude for each of them. Then create a marker for each unique latitutde and longitude. This can be achieved using an information window (see figures 4.1, 4.2, and 4.3 below).

5. 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 4.4 below.

6. 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 textbox 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 4.5 and 4.6 for and illustration of this functionality.

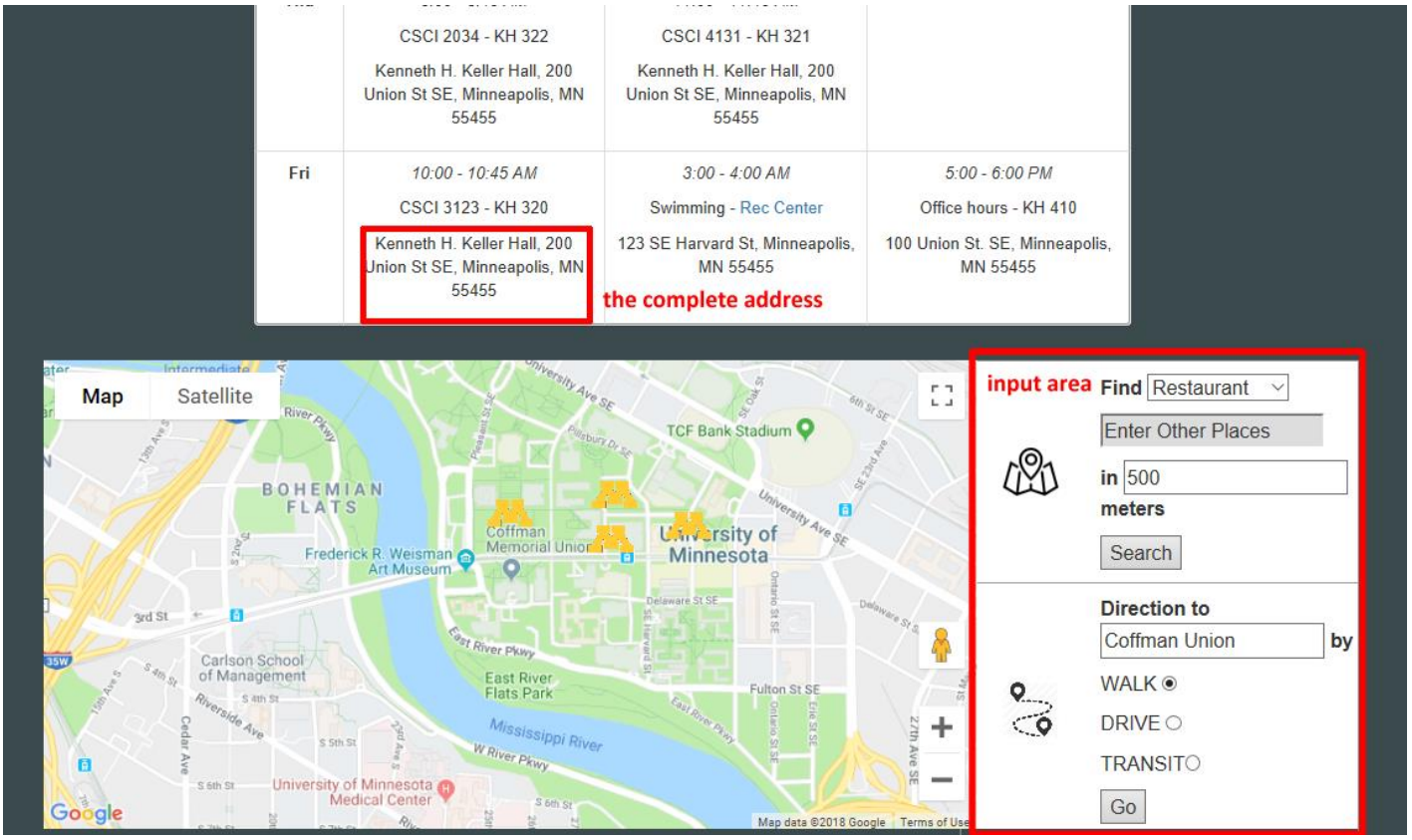
7. The second component that should be added is the functionality to get directions from a user's current location to the destination they provide. The group of elements you should use to implement this functionality are as follows:
    - A textbox that allows the user to input the destination location. 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 4.6 and 4.7 for an illustration of this functionality

8. You should keep the Date input field on your current calendar (*not currently shown in figure 4.8*) and 2 extra elements – an input element labelled: 'Additional Info" that accepts a text description, and an input URL element that accepts a URL (and only input in the form of a URL) as input. See 4.8 for an illustration of this form.

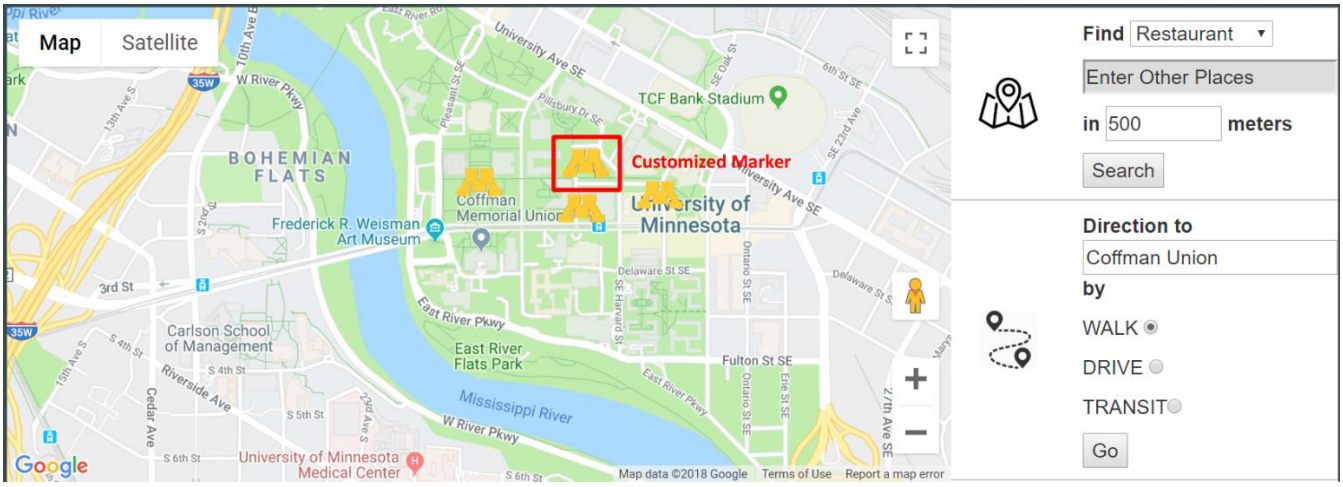
9. Add a google Translate widget under the nav bar. Use Google translate service to translate your pages into a different language. See figure 4.9 for an illustration of this.

4 Screenshots

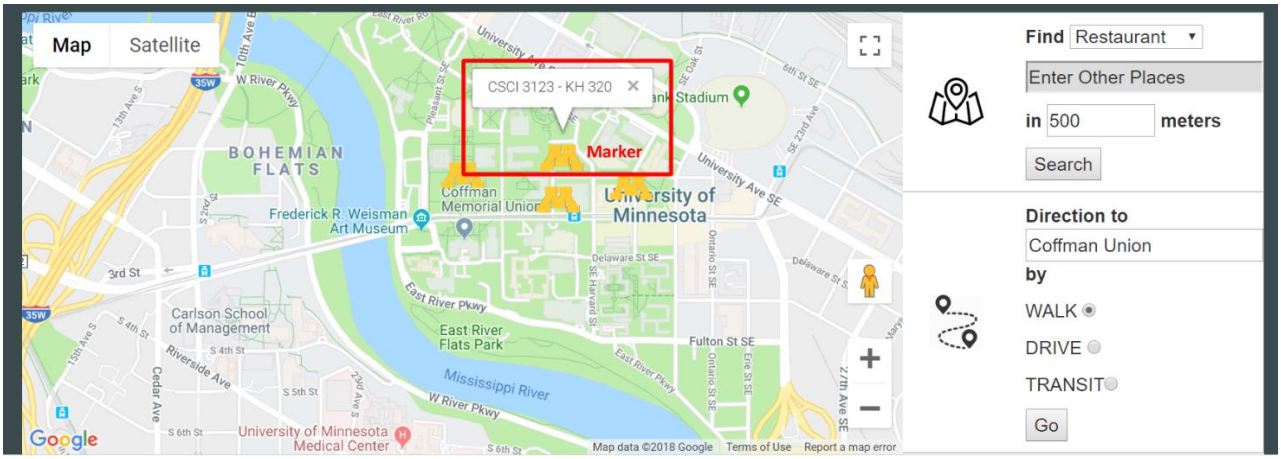
4.1 Table with a list of events, google map and form/input area floating to the right of the map



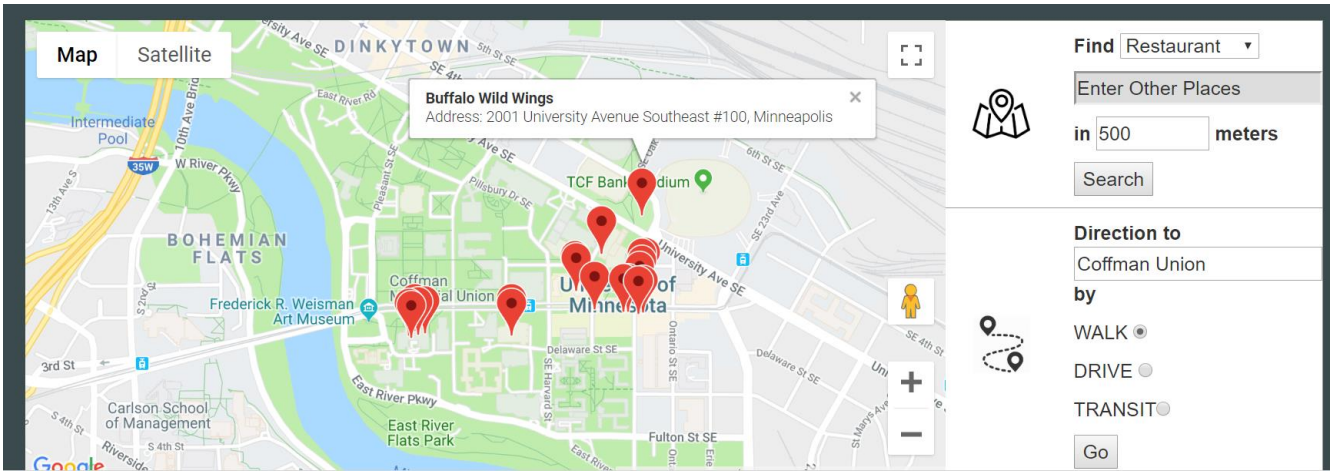
4.2 Map with the event places from the table (you should use custom icons for markers)



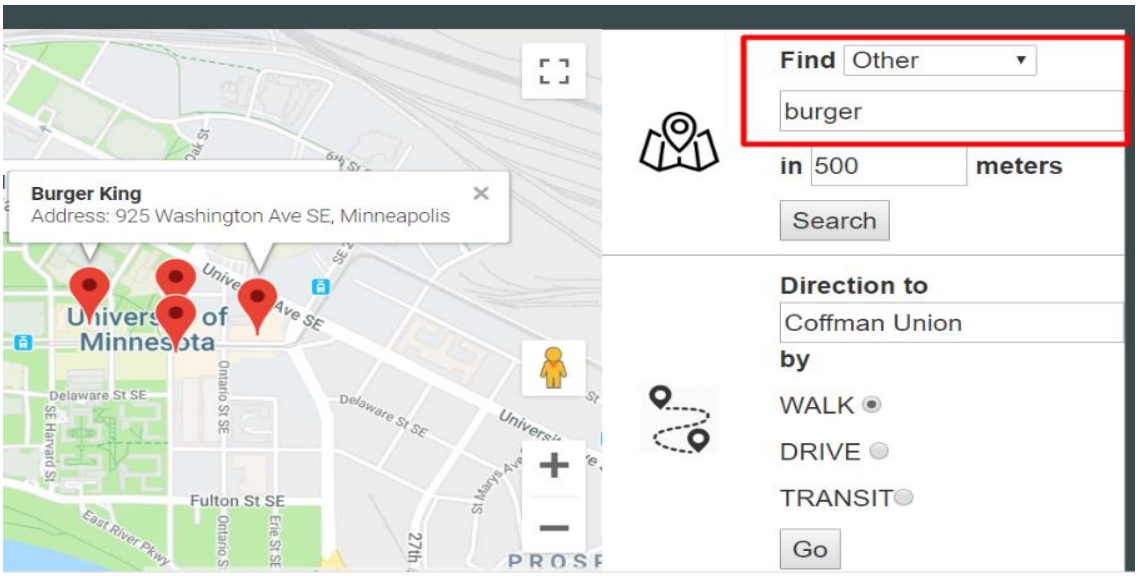
4.3 Marker shows the name of the event place on mouseover



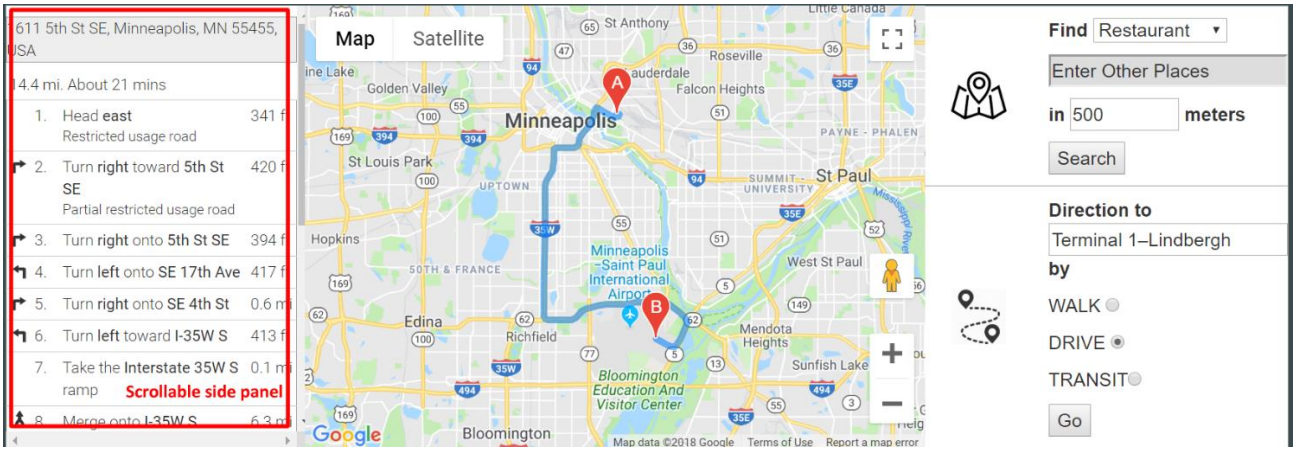
4.4 Map showing the result of nearby restaurants in a radius of 500 meters



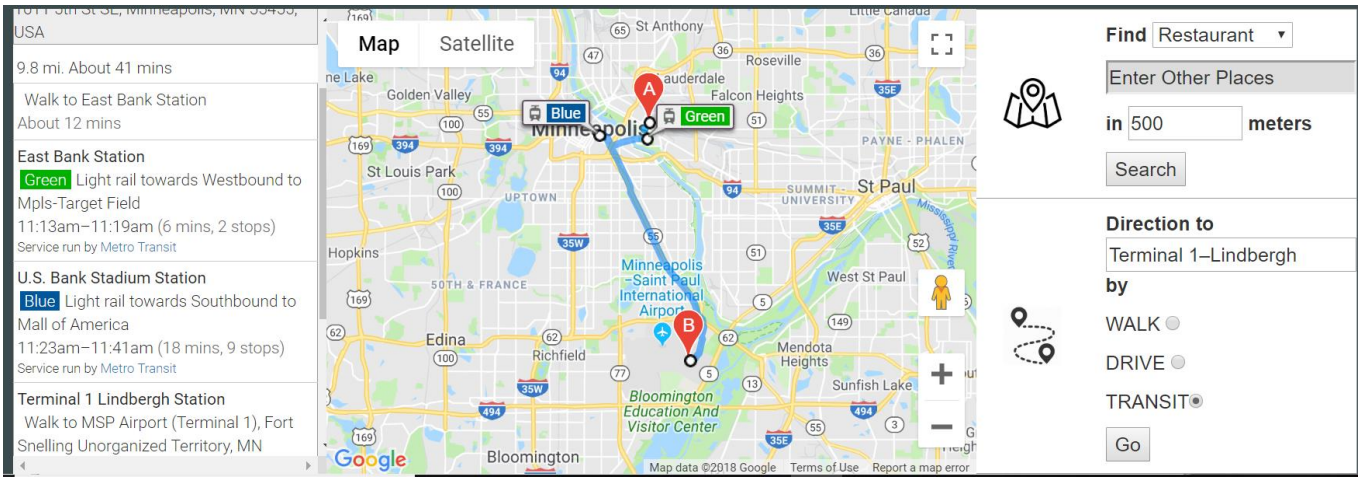
4.5 When "Other" is selected in the dropdown list, user can enter the keyword of places, e.g., "burger". The nearby places with this keyword in their names will show up when "search" button is clicked.



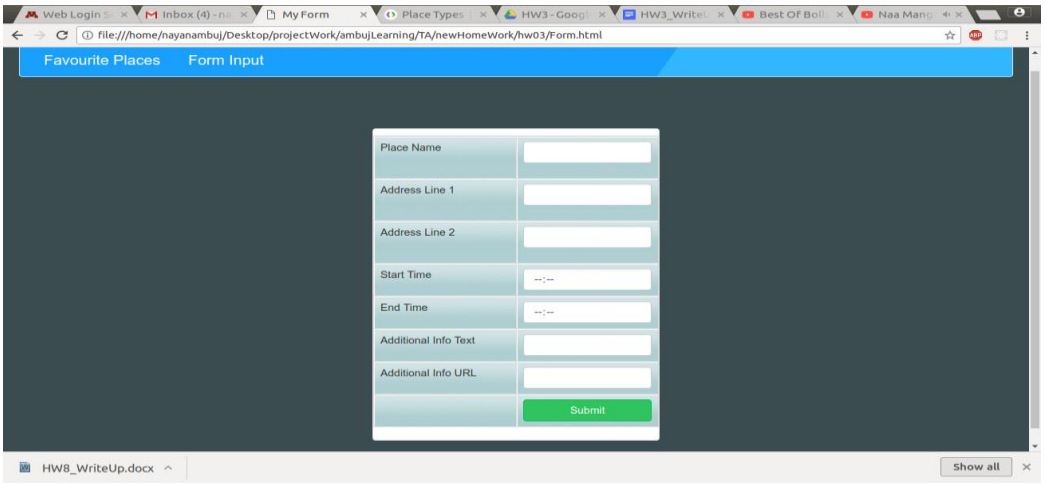
4.6 Map showing the DRIVING route between the current place and Terminal 1–Lindbergh



4.7 Map showing the TRANSIT route between the current place and Terminal 1–Lindbergh



4.8 Form – should be updated to include additional Info Text box, and a URL HTML element



4.9 Google Translate Widget

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 5 files (3 HTML, and 1 each for JavaScript and CSS).
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 100 points on the following items:

1. HTML files pass w3schools validator (http://validator.w3.org) without errors. Warnings are accepted (**5 points**).
2. CSS files pass w3schools validator (http://jigsaw.w3.org/css-validator/) without errors. Warnings are accepted (**5 points**).
3. Google Map is placed on webpage below the table with proper alignment as shown in pictures (**10 points**).
4. Custom markers are dynamically placed on the locations specified in your table (**15 points**).
5. When the mouse hover over the marker, markers display the name of the event and location (**5 points**).
6. Nearby places can be searched within a specific radius and corresponding markers are created on map **(10 points)**.
7. Upon mouseover of the marker, markers display the name of the nearby place (**5 points**).
8. All previously displayed markers are removed before the results of a new search are displayed **(5 points)**.
9. Correctly working search capability to find and mark places on the Google Map. **(10 points)**

10. 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 (**8 points**).
    - Providing option to switch between multiple travel modes (**7 points**).
    - Style and alignment of the side panel with scroll feature as shown in picture (**5 points**).
11. The google translate widget translates the pages into different language. **(5 points)**
12. The form page is modified to contain additional input fields to accept a text description and a URL. The new input elements should be labelled as illustrated in figure 4.8 (**5 points**).
13. 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.