Homework 2: HTML, CSS, JavaScript, Node.js, jQuery

Objectives:
- o To use HTML, CSS, JavaScript and Node.js.
- o To use and implement sockets.
- o To use jQuery methods to make GET and POST requests.
- o To use leaflet.js interactive maps in web applications.

Submission instructions:

This is an **individual** assignment.

• You should push your solution to your GitHub repository.
• Upload on iLearn:
- o The solution document with cover page, link to your GitHub repository, code, screenshots of outputs.
- o The code files.

Due Date: **Thursday 29th October, 2020**

Useful Resources:
- ▪ **Lectures:**
  - o Socket programming using node.js
  - o jQuery
- ▪ **Leaflet.js Tutorials:** https://leafletjs.com/examples/quick-start/
- ▪ **Routing with leaflet.js:**
  https://developer.mapquest.com/documentation/samples/leaflet/v2.2/routing/single-line/
- ▪ **jQuery Tutorial:** https://www.w3schools.com/jquery/default.asp

Beeline Moto Device and Map Applications for Navigation

The Beeline Moto navigation device can be used to help the driver know his/her way while driving motorcycles (*watch video at:* https://beeline.co/pages/beeline-moto). The device interface guides the user with a single arrow that points directly to the destination (*Figure 1*).

Your task is to simulate the working of the Beeline Moto device such that the arrow direction updates based on the movement on the map.
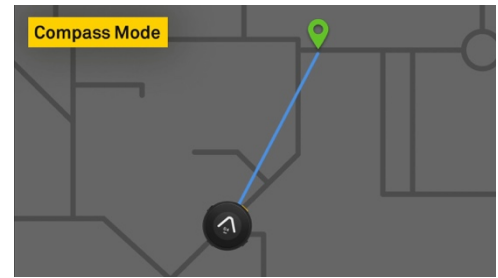

Figure 1: Compass Mode in Beeline Moto device [1]

Requirements:

- ➢ Create 2 web applications:
    1. One with an interactive map to indicate the *from*/*to* positions and see the route.
    2. One with the moto device that updates its arrow direction based on the route.
- ➢ For the first application, use the **leaflet.js** interactive map with the following plugin for routing: https://developer.mapquest.com/documentation/samples/leaflet/v2.2/routing/single-line/
- ➢ In the second application, create a circle with an arrow that represents the moto device. Use **JavaScript** to animate the arrow's movement.
- ➢ Implement a socket server through which the 2 applications can communicate.
- ➢ The process is as follows (*Figure 2*):
    1. Get the current position of the user and mark it on the map. Use the leaflet.js plugins to do that. Alternatively, you can set the current position by clicking on the map.
    2. The user inputs the destination by clicking on the map. Add a marker to show it.
    3. Use the routing plugin to calculate the route and show it on the map.
    4. The map sends a POST of the current and destination positions to the server.
    5. The moto device then does a GET to grab the positions and updates the arrow's direction according to the angle between the *from* and *to* positions.
    6. When the position changes on the map, the map application sends a GET to the socket server which saves it. The moto device then uses a GET to retrieve the new position.

Figure 2: Summary of the connections and process

- ➢ Use **jQuery** methods to do the GET and POST requests.
- ➢ You must use the raw socket server ONLY and parse all the HTTP packets.

- ➢ To simulate the movement on the map, open the applications on your phone and move around in different directions (walk, run, drive). You should see the arrow's direction changing as you move. Add screenshots from your phone in your solution to show that the arrow updates.
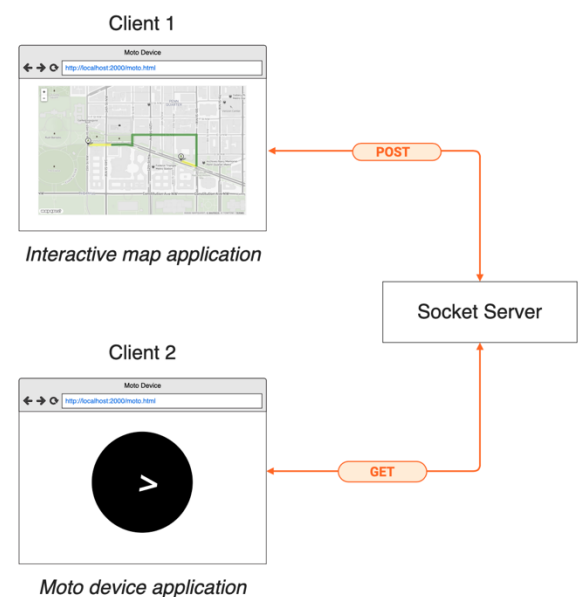
[1] Retrieved from: https://beeline.co/pages/beeline-moto