

LẬP TRÌNH GOOGLE MAP API

Mục lục

1. Lời mở đầu
2. Tổng quan về Google Map API v3
 1. Giới thiệu về công nghệ
 2. Các ứng dụng thực tế
3. Cách sử dụng và phát triển công nghệ
 1. Đăng ký sử dụng Google Map API
 2. Load bản đồ của google map về trang web cá nhân
 3. Lớp phủ lên bản đồ
 1. Markers
 2. Polylines
 3. Polygon
 4. Infowindows
 4. Sự kiện
 5. Chỉ dẫn đường trên bản đồ sử dụng service google
 1. Một ứng dụng chỉ dẫn đường nhỏ
4. Sử dụng PHP, Mysql, XML cùng với google map xây dựng ứng dụng đánh dấu các địa điểm trên bản đồ
 1. Tạo bảng
 2. Xuất ra XML với PHP
 3. Tạo bản đồ
 4. Hướng phát triển của ứng dụng
5. Tài liệu tham khảo

1. Lời mở đầu

- Từ xưa tới nay việc thiết lập một bản đồ địa lý cho một vùng luôn luôn được sự quan tâm lớn đối với con người. Nhắc tới bản đồ ta thường hình dung tới nó chỉ phục vụ cho các nhà thám hiểm các nhà địa lý, khảo cổ. Nhưng không hẳn vậy một bản đồ được xây dựng giúp cho nhà nước, cho tất cả các nghiên cứu khoa học, các thương nhân... Mỗi người dùng đều có mục đích riêng của mình.
- Ngày nay với sự bùng nổ của Internet, có rất nhiều bản đồ số được xây dựng. Nó rất quan trọng cho con người và công việc. Nhu cầu con người ngày càng lớn, ai cũng cần đến bản đồ(bạn thử hình dung khi đầu tiên đến Hà Nội với kiến trúc như bây giờ nếu không có bản đồ trong tay thì sẽ như thế nào?). Vì vậy ai, đơn vị, tổ chức nào? sẽ cung cấp cho chúng ta?
- Google Map cung cấp cho chúng ta toàn bộ bản đồ của thế giới. Không những vậy, các API họ cung cấp giúp cho các nhà phát triển xây dựng nên các ứng dụng đáp ứng phần nào đó nhu cầu của con người. Giờ đây Google đã phát triển nên đến phiên bản gọi là "Google Map API V3".

2. Tổng Quan về Google Map API v3

2.1. Google Map API là gì?

- Google Maps là một dịch vụ ứng dụng và công nghệ bản đồ trực tuyến trên web miễn phí được cung cấp bởi Google, hỗ trợ nhiều dịch vụ khác của Google nổi bật là dẫn đường. Nó cho phép thấy bản đồ đường sá, đường đi cho xe đạp, cho người đi bộ và xe hơi, và những địa điểm kinh doanh trong khu vực cũng như khắp nơi trên thế giới.
- Map API là gì?

- Đó là một phương thức cho phép 1 website B sử dụng dịch vụ bản đồ của site A (gọi là Map API) và nhúng vào website của mình (site B). Site A ở đây là google map, site B là các web site cá nhân hoặc tổ chức muốn sử dụng dịch vụ của google, có thể rê chuột, zoom, đánh dấu trên bản đồ...
- Các ứng dụng xây dựng trên maps được nhúng vào trang web cá nhân thông qua các thẻ javascripts do vậy việc sử dụng API google rất dễ dàng.
- Google Map API đã được nâng cấp lên phiên bản thứ 3. Phiên bản này hỗ trợ không chỉ cho các máy để bàn truyền thống mà cho cả các thiết bị di động. Nhanh hơn và nhiều hơn các ứng dụng.
- Điều quan trọng là các dịch vụ hoàn toàn miễn phí với việc xây dựng một ứng dụng nhỏ. Trả phí nếu đó là việc sử dụng cho mục đích kinh doanh, doanh nghiệp.

2.2. Một số ứng dụng có thể xây dựng

- Đánh dấu các địa điểm trên bản đồ cùng các thông tin cho địa điểm: các khu vui chơi giải trí, nhà hàng khách sạn, các quán ăn ngon, các shop quần áo, nữ trang...
- Chỉ dẫn đường đến các địa điểm cần tìm, chỉ dẫn đường giao thông công cộng, có thể là các địa điểm cung cấp như trên. Ở đây sử dụng các service google cung cấp.
- Khoanh vùng khu vực: các trung tâm kinh tế, khu đô thị, khu ô nhiễm...
- Tình trạng giao thông các khu vực...Đưa ra các giải pháp có thể...

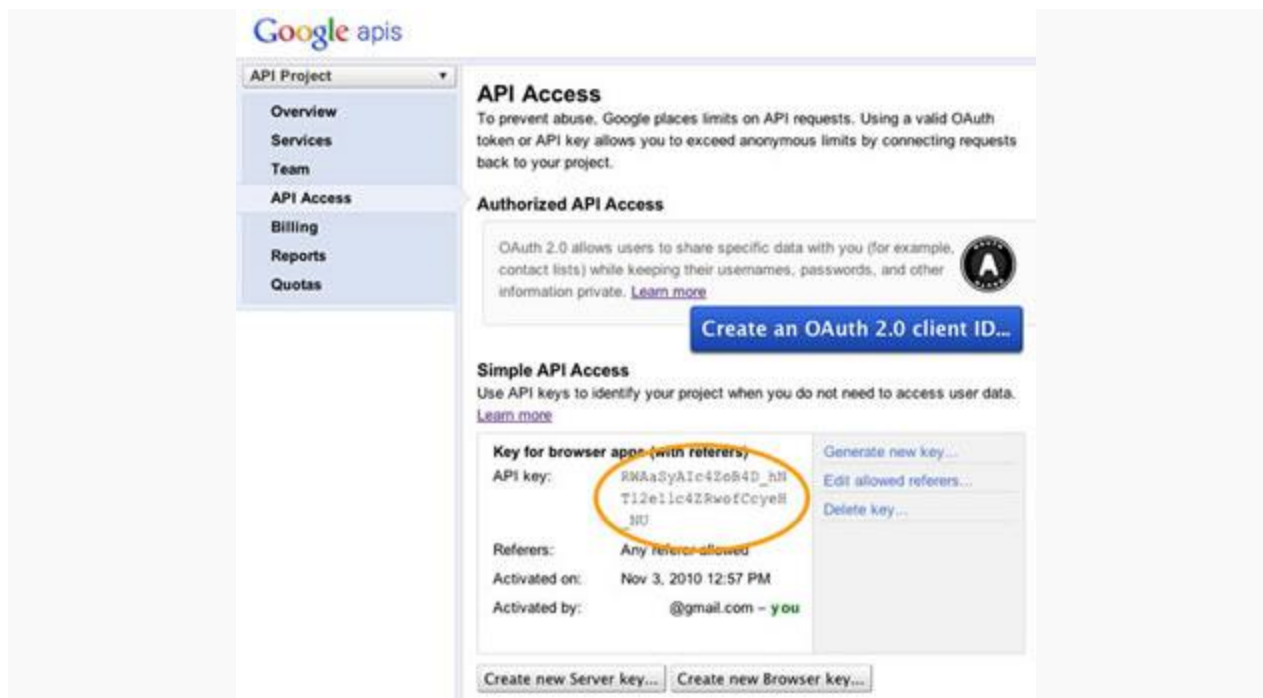
Còn rất nhiều ứng dụng cho phép bạn xây dựng. Quan trọng là đều mang lại lợi ích cho người cung cấp dịch vụ và người sử dụng dịch vụ. Có thể đem lại lợi ích kinh tế nếu như ứng dụng áp dụng tốt trong thực tế!

3. Cách sử dụng và phát triển công nghệ

- Hoàn toàn bằng các mã javascripts.

3.1 Đăng ký sử dụng Google map API

- Tất cả các ứng dụng Maps API nên tải Maps API sử dụng một API key. Một key API cho bạn kiểm soát các ứng dụng của mình và cũng là việc google có thể liên lạc với bạn về ứng dụng có ích bạn đang xây dựng. Từ đó có cơ hội phát triển bản thân mình.
- Tạo một API key:
 - Truy cập vào <https://code.google.com/apis/console> và đăng nhập bằng tài khoản gmail của mình.
 - Click vào Services link bên trái menu.
 - Kéo xuống dưới tìm Google Maps API v3 service và kích hoạt dịch vụ.
 - Click API Access, một API key sẽ hiện lên và bạn sẽ copy lại để sử dụng.



Hình 1.1

3.2 Load bản đồ về trang web cá nhân

- Khi đã có key google cung cấp ta sử dụng key đó trong đoạn mã javascripts trong thẻ <head>:

```
<script type="text/javascript"
src="http://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&sensor=SET_TO_TRUE_OR_FALSE"
</script>
```

- Tạo một hàm trong javascripts:

```
'<script type="text/javascript">
function initialize() {
var myOptions = {
center: new google.maps.LatLng(-34.397, 150.644),
zoom: 8,
mapTypeId: google.maps.MapTypeId.ROADMAP
};
var map = new google.maps.Map(document.getElementById("map_canvas"),
myOptions);
}
</script>'
```

- Đầu tiên tạo một đối tượng bản đồ chứa các biến khởi tạo bản đồ

```
var myOptions = {
```

```
center: new google.maps.LatLng(-34.397, 150.644),
zoom: 8,
mapTypeId: google.maps.MapTypeId.ROADMAP
};
```

- center: một điểm làm trung tâm của bản đồ Latitudes(vĩ độ) và Longitudes(kinh độ). Tương tự như trên khi ta định nghĩa 1 điểm là tọa độ trong Map ta để nó nằm trong new google.maps.LatLng(lat,lng). Một điểm được xác định bởi vĩ độ và kinh độ.
- zoom: độ zoom được quy định khi Map được load.
- Map type: loại Map được hiển thị sau khi load xong. có 4 loại để chọn: ROADMAP, SATELLITE, HYBRID, TERRAIN
- Map object: var map = new google.maps.Map(document.getElementById("map_canvas"), myOptions); Xác định id html chứa đối tượng Map với tùy chọn "myOptions" như trên.

- Mã javascripts:

```
'''<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
    <style type="text/css">
      html { height: 100% }
      body { height: 100%; margin: 0; padding: 0 }
      #map_canvas { height: 100% }
    </style>
    <script type="text/javascript">

src="http://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&sensor=SET_TO_TRUE_OR_FALSE"
    </script>
    <script type="text/javascript">
      function initialize() {
        var myOptions = {
          center: new google.maps.LatLng(-34.397, 150.644),
          zoom: 8,
          mapTypeId: google.maps.MapTypeId.ROADMAP
        };
        var map = new google.maps.Map(document.getElementById("map_canvas"),
          myOptions);
      }
    </script>
  </head>
  <body onload="initialize()">
    <div id="map_canvas" style="width:100%; height:100%"></div>
  </body>
</html>
```

- Trong thẻ <body> gọi onload = "initialize()" để load bản đồ vào trong thân của trang html.
- Bạn có thể xem kết quả tại: <https://google-developers.appspot.com/maps/documentation/javascript/examples/map-simple>
- Với kết quả trên ta chỉ có thể thao tác zoom, rê chuột, xem trên bản đồ. Để làm được các thao tác khác như: vẽ, đánh dấu điểm, lấy tọa độ trên bản đồ, chỉ dẫn đường...chúng ta cần xem thêm các mục được xây dựng dưới đây.

3.1 Lớp phủ trên bản đồ(Overlays)

- Sau khi load được bản đồ ta thêm các đoạn code xây dựng lớp phủ sau khi đối tượng map được khởi tạo: **var map = new google.maps.Map(document.getElementById("map_canvas"), myOptions);**. Sau này cũng vậy các ứng dụng khác nếu muốn được thêm vào thì các đoạn code được khai báo sau khi đối tượng map được khởi tạo.
- **Tổng quan:** Lớp phủ(overlays) là các đối tượng trên bản đồ và được gắn với vĩ độ, kinh độ cho nên nó sẽ di chuyển cùng bản đồ khi ta kéo hoặc zoom bản đồ. Overlays phản ánh các đối tượng mà bạn thêm vào bản đồ như points, line, areas, hoặc các "collections of object" tạm gọi là bộ sưu tập đối tượng, các đối tượng mà bạn muốn xây dựng. Vd: 1 khu vực công nghiệp, khu vực sông, khu vui chơi giải trí...
- **Các loại lớp phủ:** *markers, polylines, areas, info windows, polygons.*
- **Thêm các lớp phủ:** Đầu tiên phải xác định lớp phủ nào cần xây dựng để có thể hiển thị trên Map. Thêm lớp phủ trực tiếp lên bản đồ sử dụng phương thức setmap(). Ví dụ sau thêm lớp phủ Markers để đánh dấu điểm trên map
 - mã javascripts:

```
var myLatLng = new google.maps.LatLng(-25.363882, 131.044922);
var myOptions = {
  zoom: 4,
  center: myLatLng,
  mapTypeId: google.maps.MapTypeId.ROADMAP,
}
var map = new google.maps.Map(document.getElementById("map_canvas"),
myOptions);

var marker = new google.maps.Marker({
  position: myLatLng, //vị trí này sẽ xuất hiện điểm đánh dấu với icon mặc
định của google.
  title:"Hello World!"
});
marker.setMap(map); // đối tượng map được khởi tạo ở trên
```

- Lớp phủ Markers được thêm vào bởi đối tượng trong javascripts cùng các tùy chọn của lớp phủ:

```
var marker = new google.maps.Marker({
  position: myLatLng, // vị trí này sẽ xuất hiện điểm đánh dấu với icon mặc
định của google.
  title:"Hello World!"
});
```

sau đó gọi phương thức setmap() để thêm vào bản đồ "marker.setMap(map)"

- **Loại bỏ lớp phủ:** Để loại bỏ ta gọi phương thức setmap() của lớp phủ và đặt là null "setmap(null)". Nhưng đây mới chỉ loại bỏ tạm thời không cho chúng xuất hiện chứ chưa xóa hoàn toàn.
- Nhưng trên bản đồ với một ứng dụng nhỏ thì ít nhất ta cũng phải sử dụng trên 2 lớp phủ. Vì vậy để quản lý chúng thật sự dễ dàng ta tạo một mảng chứa các lớp phủ. Khi muốn tạo một lớp phủ ta chỉ setmap() trên các phần tử của mảng hoặc loại bỏ chúng cũng vậy. Điều quan trọng là có thể xóa các lớp phủ khi cho độ dài của mảng bằng 0.

```
var map;
var markersArray = [];

function initialize() {
  var haightAshbury = new google.maps.LatLng(37.7699298, -122.4469157);
  var mapOptions = {
    zoom: 12,
    center: haightAshbury,
```

```

        mapTypeId: google.maps.MapTypeId.TERRAIN
    };
    map = new google.maps.Map(document.getElementById("map_canvas"),
    mapOptions);

    google.maps.event.addListener(map, 'click', function(event) {
        addMarker(event.latLng);
    });
}

function addMarker(location) {
    marker = new google.maps.Marker({
        position: location,
        map: map
    });
    markersArray.push(marker);
}

// Removes the overlays from the map, but keeps them in the array
function clearOverlays() {
    if (markersArray) {
        for (i in markersArray) {
            markersArray[i].setMap(null);
        }
    }
}

// Shows any overlays currently in the array
function showOverlays() {
    if (markersArray) {
        for (i in markersArray) {
            markersArray[i].setMap(map);
        }
    }
}

// Deletes all markers in the array by removing references to them
function deleteOverlays() {
    if (markersArray) {
        for (i in markersArray) {
            markersArray[i].setMap(null);
        }
        markersArray.length = 0;
    }
}

```

- Xem kết quả tại: <https://google-developers.appspot.com/maps/documentation/javascript/examples/overlay-remove>
- Sau đây chúng ta sẽ tìm hiểu kỹ về các lớp phủ.

3.3.1 Markers

- Markers dùng để xác định một điểm trên bản đồ hoặc đánh dấu địa điểm dựa trên vĩ độ và kinh độ. Theo mặc định sử dụng icon của google làm hình ảnh hiện lên điểm đánh dấu. Hoặc muốn một icon của mình thì ta gọi phương thức `setIcon()`.
- Sau đây là các trường được xây dựng trong **`new google.maps.Marker:`**

- Position(bắt buộc): quy định là tọa độ LatLng của điểm được đánh dấu.
- map(tùy chọn): quy định đối tượng bản đồ được đánh dấu. nếu thuộc tính này được khai báo trong đây nó sẽ thay thế cho Markers.setMap(map).
- Icon(tùy chọn): hình ảnh tùy chọn mà bạn muốn hiển thị thay thế hình ảnh mặc định. Dùng tùy chọn này thay thế cho marker.setIcon(link đến hình ảnh trong thư mục).
- Title(tùy chọn): tiêu đề của địa điểm.
- draggable(tùy chọn): thuộc tính động của điểm đánh dấu, thể hiện sự chuyển động của điểm được đánh dấu. Nếu **TRUE** tính động được khởi động.
- animation: cách thức chuyển động của điểm đánh dấu. CÓ 2 cách thức DRAG, Bounce.

```
var myLatLng = new google.maps.LatLng(-25.363882,131.044922);
var myOptions = {
  zoom: 4,
  center: myLatLng,
  mapTypeId: google.maps.MapTypeId.ROADMAP
}
var map = new google.maps.Map(document.getElementById("map_canvas"),
myOptions);
var icon = "link đến hình ảnh";
var marker = new google.maps.Marker({
  position: myLatLng,
  map: map,
  icon:icon,
  title:"Hello World!",
  draggable:true,
  animation:google.maps.Animation.DROP
});
```

- xem KQ ví dụ:<https://google-developers.appspot.com/maps/documentation/javascript/examples/marker-simple>
- Hoặc Icon tùy chọn mã nguồn:<http://www.mediafire.com/?tbagn3ab456tjuq>.
- Tính động của Marker:<https://google-developers.appspot.com/maps/documentation/javascript/examples/marker-animations> hoặc theo icon tùy chọn:<http://www.mediafire.com/?s2s772yc7edib62>

- Mã javascripts sau mô tả việc đánh dấu nhiều địa điểm trên bản đồ:

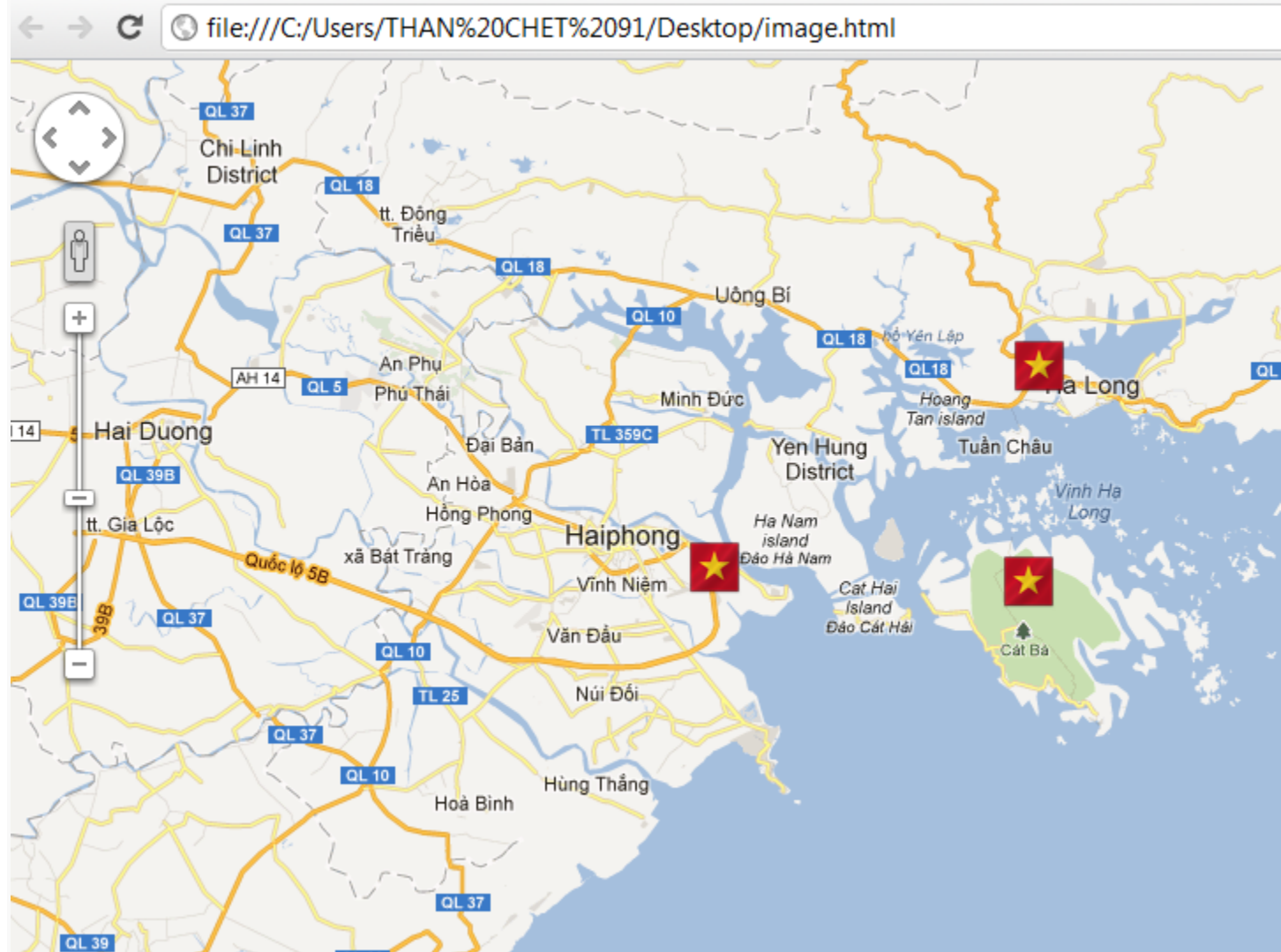
```
<script type="text/javascript">
function initialize() {
  var myOptions = {
    zoom: 10,
    center: new google.maps.LatLng(20.779227,107.0755),
    mapTypeId: google.maps.MapTypeId.ROADMAP
  }
  var map = new
google.maps.Map(document.getElementById("map_canvas"),myOptions);
  var image = "vietnam_32.png";
  for (var i = 0; i < toado.length; i++) {
    var beach = toado[i];
    var myLatLng = new google.maps.LatLng(beach[0], beach[1]);
    var marker = new google.maps.Marker({
      position: myLatLng,
      map: map,
      icon: image
    });
  }
};
```

```

    }
    //setMarkers(map, beaches);
}
var toado = [
    [20.817741,106.751404],
    [20.964004,107.004089],
    [20.807472,106.99585]
];
</script>

```

Hình ảnh của kết quả:



hoặc link mã nguồn: <http://www.mediafire.com/?cdps776024e7xe2>

3.3.2 Polyline

- Polyline dùng để thể hiện đường kết nối trên bản đồ dựa vào các tọa độ. Các đoạn thẳng được hiển thị với các tùy chọn cho nó như màu sắc, độ đậm nhạt, độ rộng của đường. Phải có tối đa 2 điểm để tạo nên 1 đường thẳng.
- Cũng giống như lớp phủ Markers ta khai báo lớp phủ Polyline và các thuộc tính của nó như sau:

```
new google.maps.Polyline({
```



```
path: //giá trị là một mảng chứa các tọa độ cần nối với nhau
strokeColor: "#FF0000", // tùy chọn màu sắc
strokeOpacity: 1.0, // độ đậm nhạt của màu sắc
strokeWeight: 2 // độ rộng của đường
})
```

- Mã javascripts sau sẽ hiển thị đường nối các điểm được khai báo trong 1 mảng.

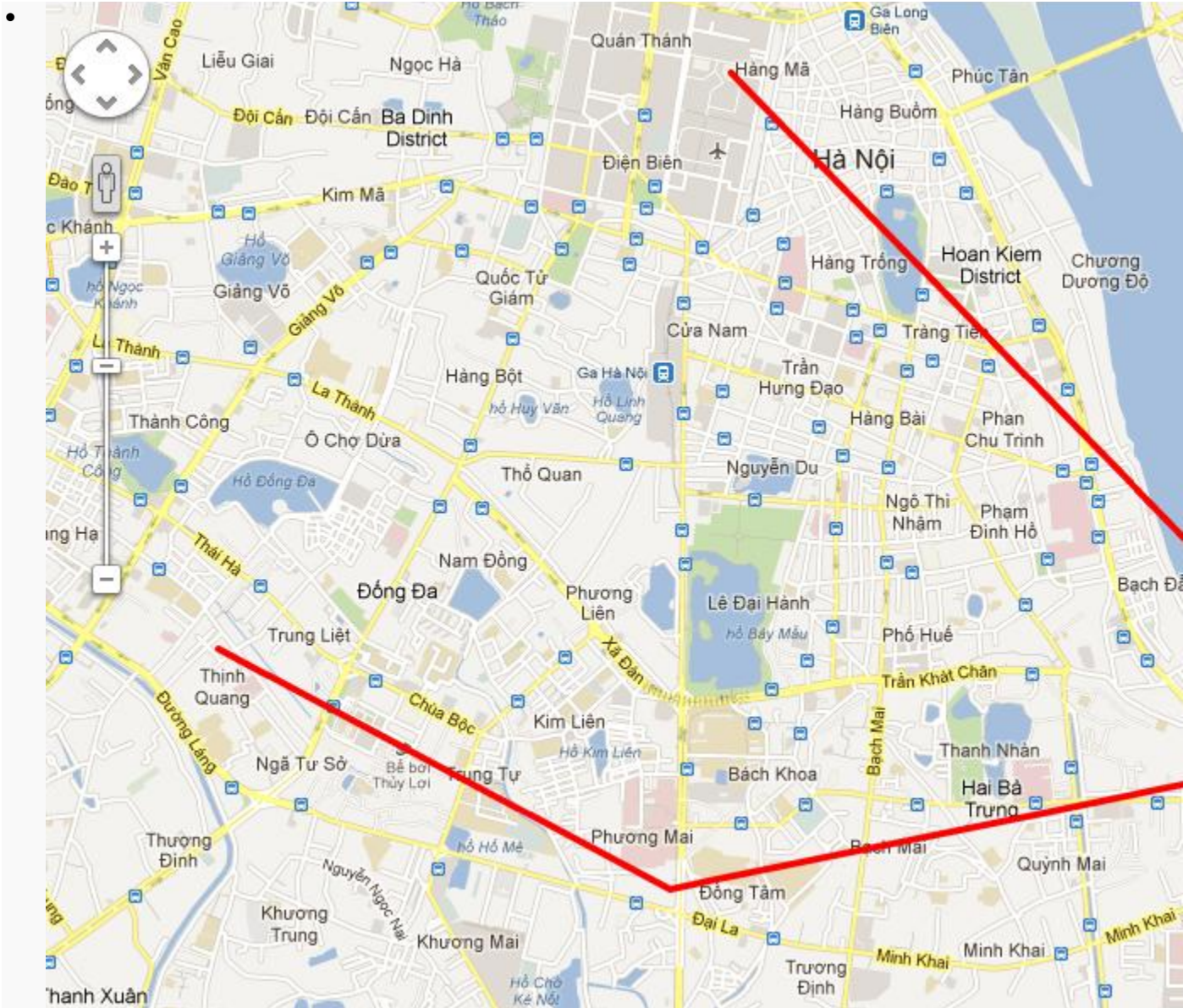
```
<script type="text/javascript">

function initialize() {
    var myLatLng = new google.maps.LatLng(21.010163,105.817823);
    var myOptions = {
        zoom: 14,
        center: myLatLng,
        mapTypeId: google.maps.MapTypeId.ROADMAP
    };

    var map = new google.maps.Map(document.getElementById("map_canvas"),
myOptions);

    var flightPlanCoordinates = [
        new google.maps.LatLng(21.010163,105.817823),
        new google.maps.LatLng(20.998785,105.84074),
        new google.maps.LatLng(21.005676,105.876961),
        new google.maps.LatLng(21.037403,105.84383)
    ];
    var flightPath = new google.maps.Polyline({
        path: flightPlanCoordinates,
        strokeColor: "#FF0000",
        strokeOpacity: 1.0,
        strokeWeight: 2
    });

    flightPath.setMap(map);
}
</script>
```



- Mã nguồn: <http://www.mediafire.com/?tfww26l2flwrqi5>

3.3.3 Polygon

- Cũng giống như đối tượng Polyline, Polygon xây dựng dựa trên một loạt các tọa độ, tuy nhiên thay vì mở thì nó hoàn toàn khép kín trong một khu vực.

```
new google.maps.Polygon({
  paths: triangleCoords, //tập các tọa độ với tọa độ đầu trùng tọa độ cuối
  đảm bảo cho việc khép kín.
  strokeColor: "#FF0000", //màu sắc của đường nối điểm
  strokeOpacity: 0.8, //độ đậm nhạt
  strokeWeight: 2, //độ rộng
  fillColor: "#FF0000", //màu sắc của đa giác
});
```

```
        fillOpacity: 0.35//độ đậm nhạt của đa giác
    });

<script type="text/javascript">

function initialize() {
    var myLatLng = new google.maps.LatLng(21.037403,105.84383);
    var myOptions = {
        zoom: 14,
        center: myLatLng,
        mapTypeId: google.maps.MapTypeId.ROADMAP
    };

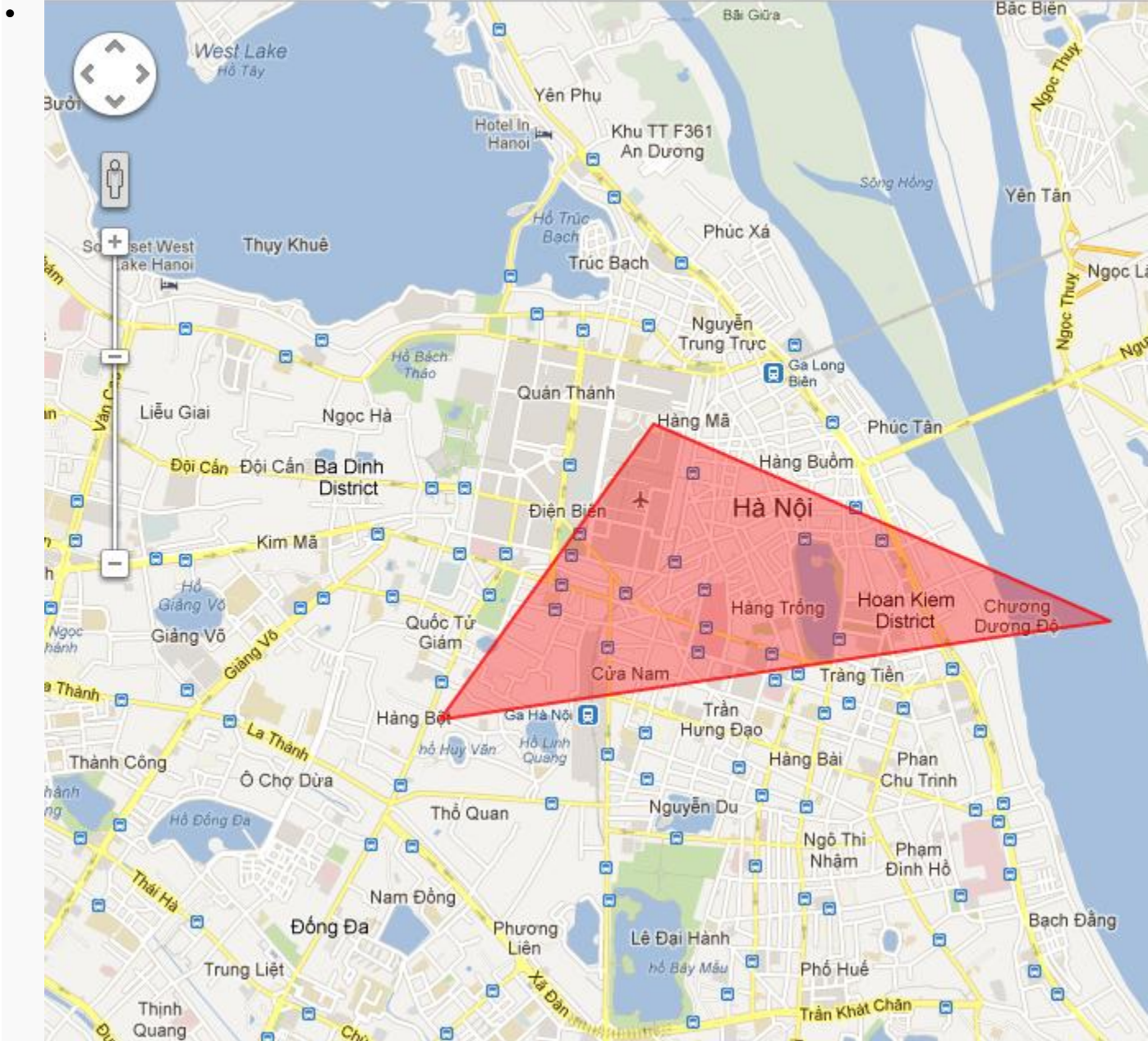
    var bermudaTriangle;

    var map = new google.maps.Map(document.getElementById("map_canvas"),
        myOptions);

    var triangleCoords = [
        new google.maps.LatLng(21.037403,105.84383),
        new google.maps.LatLng(21.022983,105.832672),
        new google.maps.LatLng(21.02779,105.867691),
        new google.maps.LatLng(21.037403,105.84383)
    ];

    // Construct the polygon
    bermudaTriangle = new google.maps.Polygon({
        paths: triangleCoords,
        strokeColor: "#FF0000",
        strokeOpacity: 0.8,
        strokeWeight: 2,
        fillColor: "#FF0000",
        fillOpacity: 0.35
    });

    bermudaTriangle.setMap(map);
}
</script>
```

- Mã nguồn: <http://www.mediafire.com/?4b7i1glj488zj2b>

3.3.4 Infowindows

- Infowindows là một cửa sổ chứa các thông tin. Trông như hình ảnh sau:



- Tạo một Infowindows cũng giống như các lớp phủ khác:

```
new google.maps.InfoWindow({
  content: 'text'//chứa nội dung Infowindows
});
```

- Gọi `Infowindows.open()` để có thể mở được cửa sổ.
- Xem và tham khảo tại: <https://google-developers.appspot.com/maps/documentation/javascript/examples/infowindow-simple>

3.4 Sự kiện

- Một số đối tượng trong Maps API được thiết kế để đáp ứng với sự kiện người sử dụng chẳng hạn như các sự kiện chuột hoặc bàn phím. Một đối tượng `google.maps.Marker` có thể lắng nghe người sử dụng các sự kiện sau đây, ví dụ :
 - 'click'
 - 'dblclick'
 - 'mouseup'
 - 'mousedown'
 - 'mouseover'

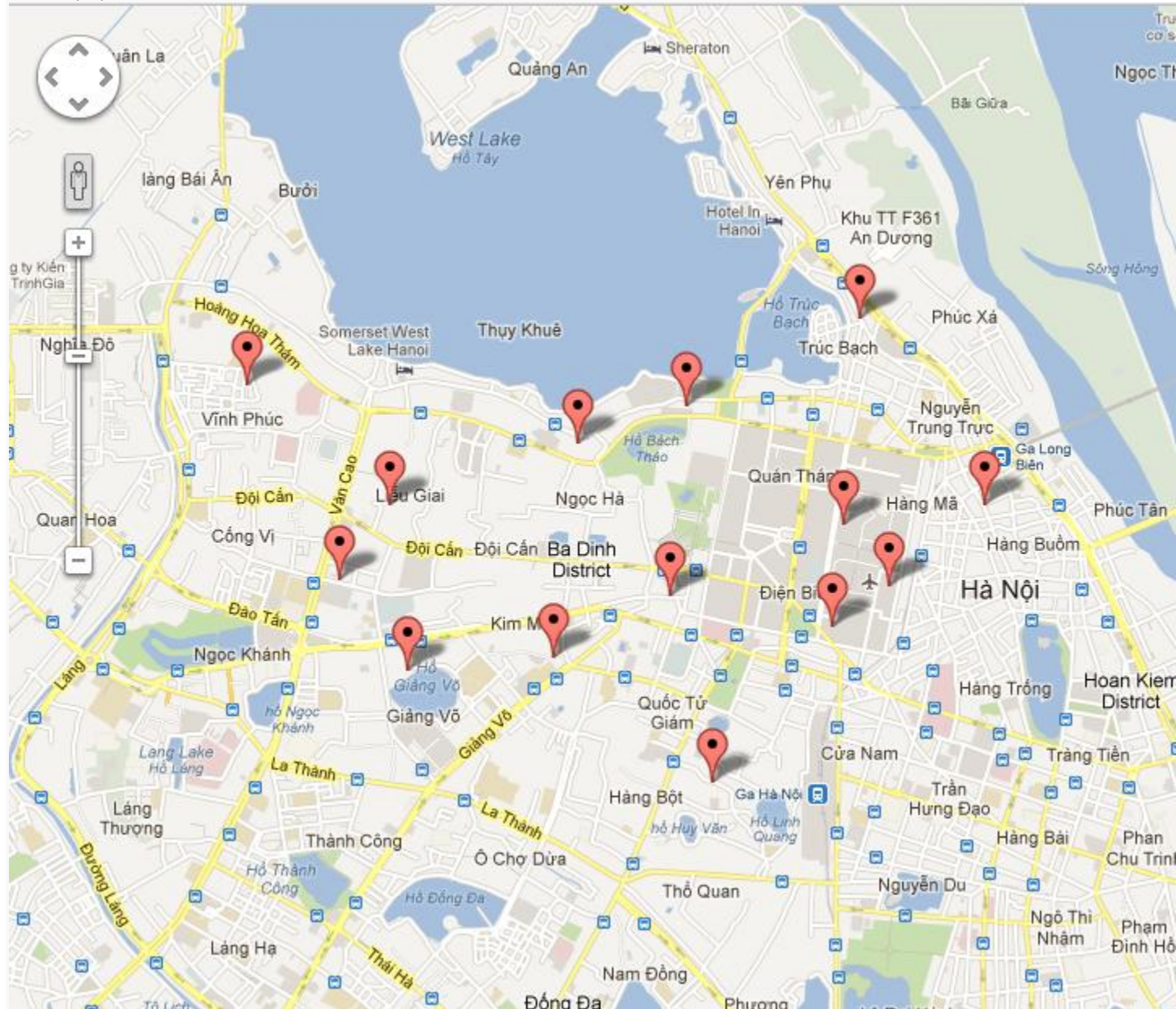
- 'mouseout'
- Một sự kiện được gọi như sau:

```
google.maps.event.addListener(đối tượng, sự kiện gọi, function() {  
}
```

- Sau đây sẽ trình bày các sự kiện trên các đối tượng map:

```
<script type="text/javascript">  
  
function initialize() {  
    var myLatLng = new google.maps.LatLng(21.037403,105.84383);  
    var myOptions = {  
        zoom: 14,  
        center: myLatLng,  
        mapTypeId: google.maps.MapTypeId.ROADMAP  
    };  
    var map = new google.maps.Map(document.getElementById("map_canvas"),  
        myOptions);  
    google.maps.event.addListener(map, 'click', function(event) {  
        placeMarker(event.latLng);  
    });  
    function placeMarker(location) {  
        var marker = new google.maps.Marker({  
            position: location,  
            map: map  
        });  
    }  
}  
</script>
```


Khi click chuột vào map thì các điểm đánh dấu sẽ xuất hiện trông giống như hình ảnh sau các icon là mặc định, nếu muốn thay đổi thì đặt lại icon:



- Mã nguồn: <http://www.mediafire.com/?6oc3xo6lq7oab59>
- Sự kiện trên đối tượng markers:

```
<script type="text/javascript">
```

```
function initialize() {  
    var myLatLng = new google.maps.LatLng(-25.363882,131.044922);  
    var myOptions = {  
        zoom: 4,  
        center: myLatLng,  
        mapTypeId: google.maps.MapTypeId.ROADMAP  
    }  
}
```

```
var map = new google.maps.Map(document.getElementById("map_canvas"),
myOptions);

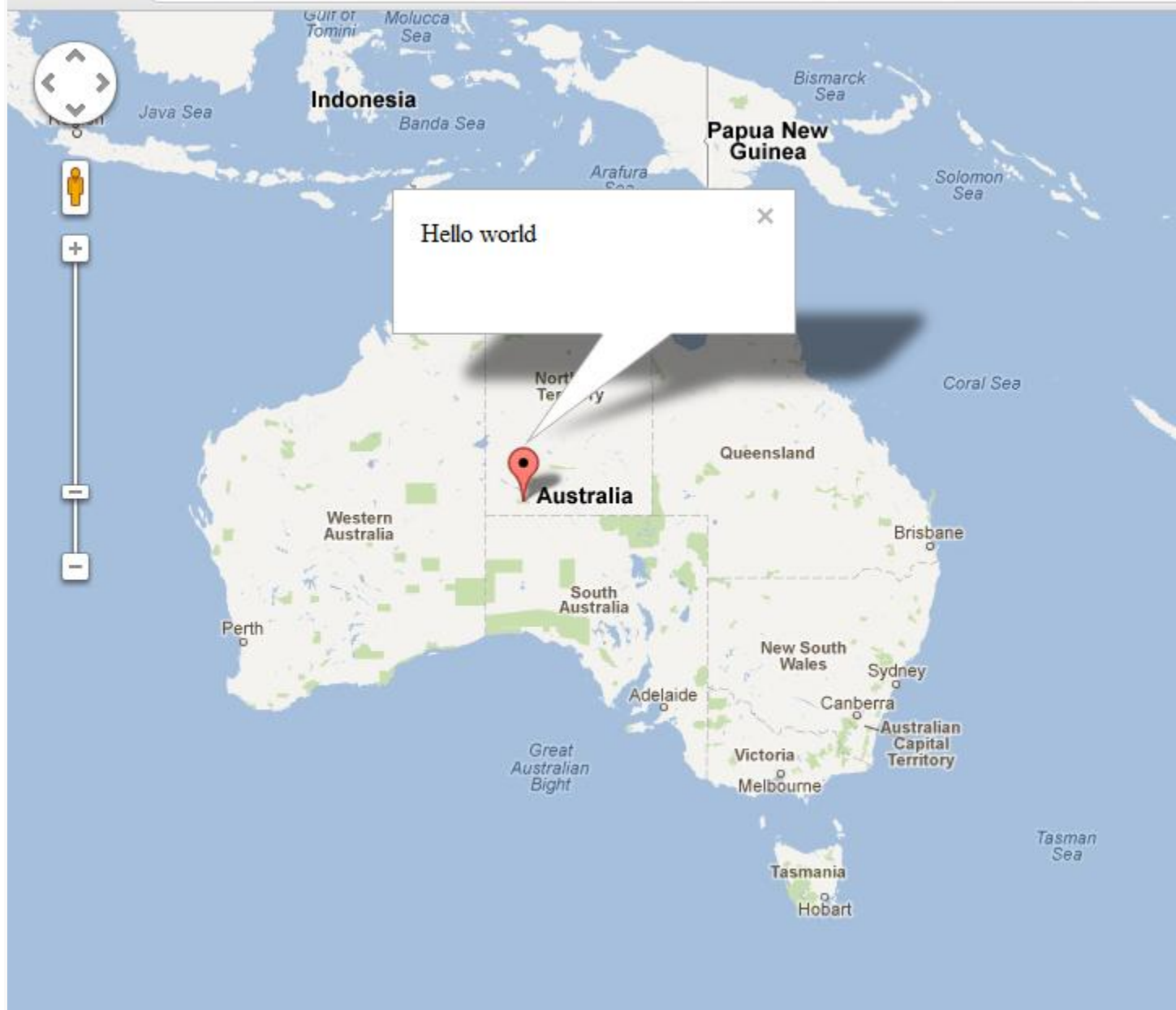
var contentString = 'Hello world'

var infowindow = new google.maps.InfoWindow({
  content: contentString
});

var marker = new google.maps.Marker({
  position: myLatLng,
  map: map,
  title:"Uluru (Ayers Rock)"
});

google.maps.event.addListener(marker, 'click', function() {
  infowindow.open(map,marker);
});
</script>
```


Sự kiện xuất hiện khi click chuột vào marker một cửa sổ thông tin sẽ xuất hiện:



3.5 Chỉ dẫn đường trên bản đồ sử dụng service google

- Google service cung cấp cho chúng ta một tiện ích rất hay là chỉ dẫn đường cho người sử dụng. Khi người dùng muốn đi từ điểm đầu tới đích. Dịch vụ trả về một hướng dẫn men theo các con đường.
- Ứng dụng này được sử dụng phổ biến trong google map, chức năng chỉ dẫn đường "direction".
- Không cần vào google map ta cũng có thể sử dụng ứng dụng này chỉ bằng một vài đoạn mã javascripts:

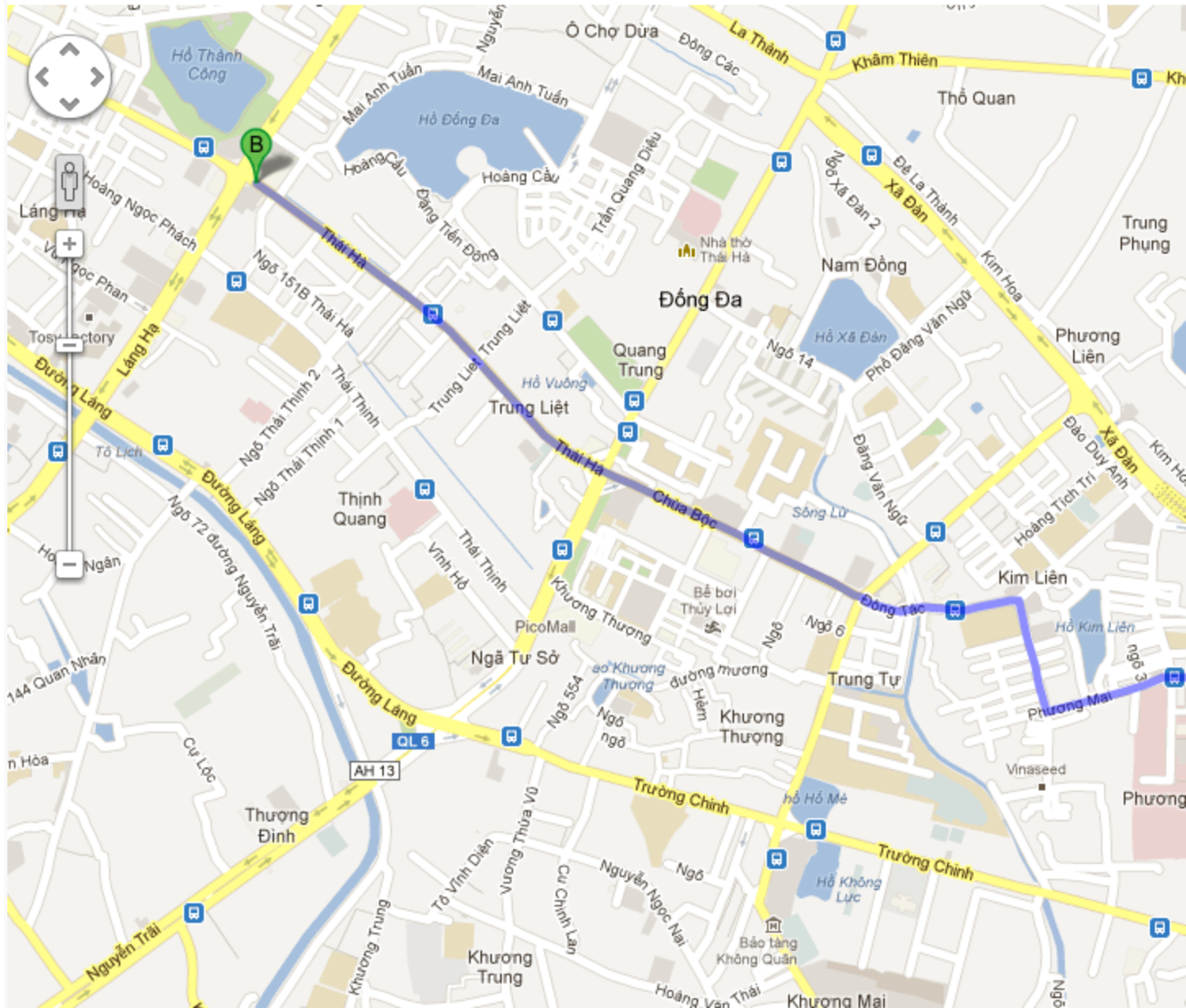
```
<script type="text/javascript">
  var directionDisplay;
  var directionservice = new google.maps.DirectionsService();
  var map;
```

```
var haight = new google.maps.LatLng(21.00183,105.841212);
var oceanBeach = new google.maps.LatLng(21.016493,105.815377);

function initialize() {
    directionsDisplay = new google.maps.DirectionsRenderer();
    var myOptions = {
        zoom: 14,
        mapTypeId: google.maps.MapTypeId.ROADMAP,
        center: haight
    }
    map = new google.maps.Map(document.getElementById("map_canvas"),
myOptions);
    directionsDisplay.setMap(map);
}

function calcRoute() {
    var request = {
        origin: haight,
        destination: oceanBeach,
        travelMode: google.maps.DirectionsTravelMode.DRIVING
    };
    directionservice.route(request, function(response, status) {
        if (status == google.maps.DirectionsStatus.OK) {
            directionsDisplay.setDirections(response);
        }
    });
}
</script>
```

Kết quả như:



- Để làm được điều này chúng ta tìm hiểu từng chi tiết sau:
 - Để sử dụng hướng dẫn chỉ đường trong V3, tạo ra một đối tượng của loại `DirectionsService` và gọi `DirectionsService.route()` để bắt đầu một yêu cầu dịch vụ Hướng dẫn. đối tượng `DirectionsRenderer` chứa các kết quả trả ra. Khác với cách load bản đồ thông thường chúng ta gọi thêm:
 - Đối tượng `DirectionService` được khởi tạo:

```
var directionservice = new google.maps.DirectionsService();
```

- Kết quả trả về cho đối tượng map

```
directionsDisplay = new google.maps.DirectionsRenderer();
```

```
directionsDisplay.setMap(map);
```

- Request tới service google:

```
directionservice.route(request, function(response, status) {  
    if (status == google.maps.DirectionsStatus.OK) {  
        directionsDisplay.setDirections(response);  
    }  
});
```

- Yêu cầu phải có điểm đầu điểm cuối và loại đường đi như: ô tô, đi bộ. Các thuộc tính này là của đối tượng DirectionRequest gửi đến service google xử lý. Sau đó kết quả được trả lại phù hợp với các thuộc tính.

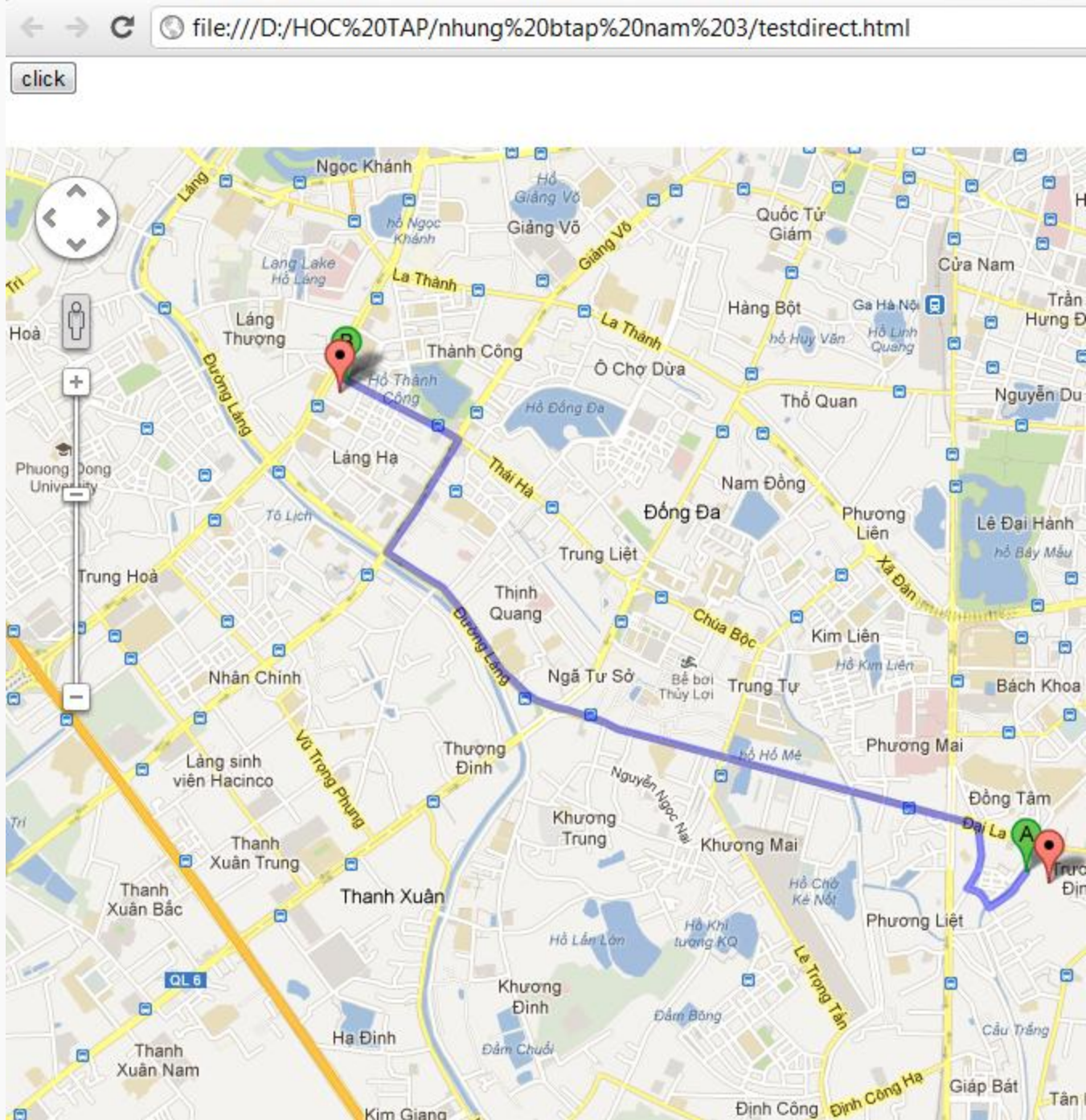
Đoạn mã sau khai báo các thuộc tính và chúng được gửi đến service google:

```
var request = {  
    origin: //điểm đầu  
    destination: điểm cuối  
    travelMode: google.maps.DirectionsTravelMode.DRIVING //chỉ dẫn theo  
đường ô tô(DRIVING), đi bộ WALKING.  
};  
directionservice.route(request, function(response, status) {  
    if (status == google.maps.DirectionsStatus.OK) {  
        directionsDisplay.setDirections(response);  
    }  
});
```

3.5.1 Ứng dụng chỉ dẫn đường nhỏ

- Tôi sẽ sử dụng service google làm nên một ứng dụng nhỏ giống như ứng dụng chỉ dẫn đường trong google map "Direction".
- Ứng dụng có chức năng: chấm 2 điểm bất kỳ trên map sau đó click button "click" để xem chỉ dẫn đường từ điểm đầu đến điểm cuối. Kết quả đảm bảo giống như kết quả tìm đường của google map.
- Ứng dụng sẽ vẫn cho kết quả cho dù google map có sự cố gần đây với dịch vụ tìm đường.

- Giao diện và kết quả như hình:



- Mã nguồn: <http://www.mediafire.com/?tml48vosn5dqbd2>

4. Sử dụng PHP, Mysql, XML cùng với google map xây dựng ứng dụng đánh dấu các địa điểm trên bản đồ

- Với lượng kiến thức tôi trình bày ở trên, và một số kiến thức về PHP/Mysql cũng đủ giúp chúng ta hiểu và làm được ứng dụng này.
- Hướng dẫn dành cho người hay sử dụng PHP, Mysql và làm thế nào để sử dụng Google Maps với một cơ sở dữ liệu Mysql.
- Bản đồ sẽ phân biệt giữa 2 loại địa điểm: quán karaoke - nhà hàng ăn. Một cửa sổ thông tin với tên và thông tin địa chỉ sẽ hiển thị ở trên một điểm đánh dấu khi click vào.
- Hướng dẫn được chia thành các bước sau:

4.1 Tạo bảng

- Để phù hợp ta tạo các trường: tên và địa chỉ của địa điểm cho cửa sổ thông tin, lat và lng là các kinh độ và vĩ độ, ngoài ra còn có id và type để phân biệt nhà hàng hay quán karaoke.
- Tạo bảng với các cột sau:

MySQL Table Editor

Table Name: Database: Comment: InnoDB free: 11264 kB

Columns and Indices | Table Options | Advanced Options

| Column Name | Datatype | NOT NULL | AUTO INC | Flags | Default Value | Comment |
|-------------|--------------|-------------------------------------|-------------------------------------|--|---------------|---------|
| id | INTEGER | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL | NULL | |
| name | VARCHAR(255) | <input checked="" type="checkbox"/> | | <input type="checkbox"/> BINARY | | |
| address | VARCHAR(255) | <input checked="" type="checkbox"/> | | <input type="checkbox"/> BINARY | | |
| lat | FLOAT | <input checked="" type="checkbox"/> | | <input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL | 0 | |
| lng | FLOAT | <input checked="" type="checkbox"/> | | <input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL | 0 | |
| type | VARCHAR(45) | <input checked="" type="checkbox"/> | | <input type="checkbox"/> BINARY | | |

Indices | Foreign Keys | Column Details

PRIMARY

Index Settings

Index Name:

Index Kind:

Index Type:

Index Columns (Use Drag'n'Drop)

id

Apply Changes Discard Changes Close

- Sau đó nhập đầy đủ các dữ liệu cho bảng. Các dữ liệu của tôi nhập vào csdl:

```
karaoke Ban Mai,"121 Đại La, Trương Định, Hai Bà Trưng Hanoi, Vietnam",
20.996662,105.846963,karaoke
karaoke Hoa Hồng,"74 phố Vọng, Phương Mai, Đống Đa Hanoi,
Vietnam",20.997783,105.842242,karaoke
nhà hàng Xuân Tuyết,"226, Đồng Tâm, Hai Bà Trưng, Hanoi,
Vietnam",21.002231,105.843186,nhahang
karaoke New Star,"345 phố Vọng, Đồng Tâm, Hai Bà Trưng Hanoi,
Vietnam",20.993536,105.842199,karaoke
nhà hàng Be Bon,"546 Bạch Mai, Trương Định, Hai Bà Trưng Hanoi,
Vietnam",20.996181,105.850396,nhahang
Nhà hàng Hoàng Gia,"4 Tạ Quang Bửu, Bách Khoa, Hai Bà Trưng Hanoi,
Vietnam",21.008,105.847607,nhahang
nhà hàng Mái Đò,"29 Đại Cồ Việt, Hai Bà Trưng Hanoi,
Vietnam",21.008601,105.849581,nhahang
karaoke Alibaba,"5 Hoa Lư, Lê Đại Hành, Hai Bà Trưng Hanoi,
Vietnam",21.010364,105.847907,karaoke
Nhà hàng Hải Hưng,"283 HỒ BA MẪU, Phương Liên, Đống Đa Hanoi,
Vietnam",21.010364,105.84074,nhahang
```

4.2 Xuất ra XML với PHP

- Bây giờ chúng ta cần phải viết một số câu lệnh PHP để xuất dữ liệu bảng vào một định dạng XML. Để Bản đồ có thể lấy dữ liệu thông qua các cuộc gọi JavaScript không đồng bộ.
- Đưa thông tin kết nối CSDL ra một file riêng biệt CSDL.php

```
<?
$username="username";
$password="password";
$databasename="username-databaseName";
?>
```

- Sử dụng chức năng domxml PHP xuất ra XML. file "xuat_xml.php"

```
<?php
require("CSDL.php");

$dom = new DOMDocument("1.0");
$node = $dom->createElement("points");
$parnode = $dom->appendChild($node);

$conconnection=mysql_connect (localhost, $username, $password);
if (!$conconnection) { die('Not connected : ' . mysql_error());}

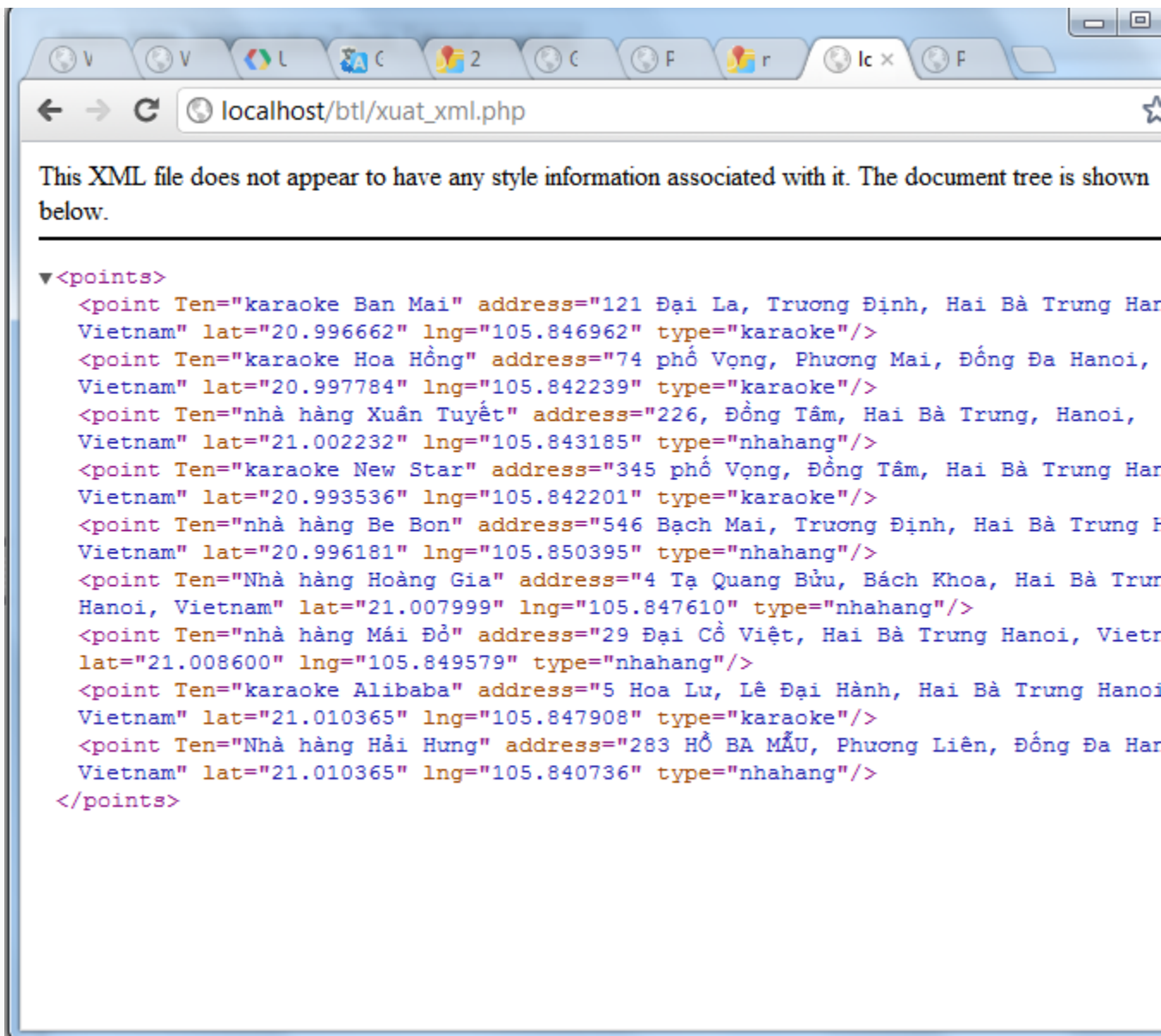
$db_selected = mysql_select_db($databasename, $conconnection);
mysql_query("SET NAMES utf8");
if (!$db_selected) {
    die ('Can\'t use db : ' . mysql_error());
}

$query = "SELECT * FROM point";
$result = mysql_query($query);
if (!$result) {
```

```
    die('Invalid query: ' . mysql_error());
}
header("Content-type: text/xml");
while (($row = mysql_fetch_array($result)){

    $node = $dom->createElement("point");
    $newnode = $parnode->appendChild($node);
    $newnode->setAttribute("Ten", $row['name']);
    $newnode->setAttribute("address", $row['address']);
    $newnode->setAttribute("lat", $row['lat']);
    $newnode->setAttribute("lng", $row['lng']);
    $newnode->setAttribute("type", $row['type']);
}
echo $dom->saveXML();

?>
```

4.3 Tạo bản đồ

- Để tải các tập tin XML vào các trang web, bạn có thể tận dụng lợi thế của các trình duyệt cung cấp đối tượng XMLHttpRequest. Đối tượng này cho phép bạn lấy một tập tin nằm trên cùng tên miền với trang web yêu cầu, và là cơ sở của lập trình "AJAX".
- Vì vậy, bạn có thể xác định chức năng của riêng để tải các tập tin, và gọi nó là downloadUrl (). Chức năng có hai tham số:
 - url xác định đường dẫn đến tập tin XML của bạn hoặc kịch bản PHP mà tạo ra các tập tin, nếu bạn muốn XML được tự động cập nhật khi CSDL thay đổi.
 - callback chức năng đó được gọi lại khi XML được trả về JavaScript.

```
function downloadUrl(url, callback) {
  var request = window.ActiveXObject ?
```

```

new ActiveXObject('Microsoft.XMLHTTP') :
new XMLHttpRequest;

request.onreadystatechange = function() {
    if (request.readyState == 4) {
        request.onreadystatechange = doNothing;
        callback(request, request.status);
    }
};

request.open('GET', url, true);
request.send(null);
}

```

- Với function đã được định nghĩa ta thực hiện gọi file xml và nhận kết quả trả về và lặp lại từ đầu đến cuối file xml để xử lý chúng. Ta lấy tất cả các trường: Ten, address, lat, lang, type để tạo các điểm đánh dấu và thông tin khi click vào.

```

downloadUrl("xuat_xml.php", function(data) {
    var xml = data.responseXML;
    var markers = xml.documentElement.getElementsByTagName("point");
    for (var i = 0; i < markers.length; i++) {
        var name = markers[i].getAttribute("Ten");
        var address = markers[i].getAttribute("address");
        var type = markers[i].getAttribute("type");
        var point = new google.maps.LatLng(
            parseFloat(markers[i].getAttribute("lat")),
            parseFloat(markers[i].getAttribute("lng")));
        var html = "<b>" + name + "</b> <br/>" + address;
        var icon = customIcons[type] || {};
        var marker = new google.maps.Marker({
            map: map,
            position: point,
            icon: icon.icon,
            shadow: icon.shadow
        });
        bindInfoWindow(marker, map, infoWindow, html);
    }
});

```

- Các Icon của nhà hàng và quán karaoke

```

var customIcons = {
    nhahang: {
        icon: 'http://labs.google.com/ridefinder/images/mm_20_blue.png',
        shadow: 'http://labs.google.com/ridefinder/images/mm_20_shadow.png'
    },
    karaoke: {
        icon: 'http://labs.google.com/ridefinder/images/mm_20_red.png',
        shadow: 'http://labs.google.com/ridefinder/images/mm_20_shadow.png'
    }
};

```

- Hàm tạo thông tin khi click vào điểm đánh dấu:

```
function bindInfoWindow(marker, map, infoWindow, html) {
    google.maps.event.addListener(marker, 'click', function() {
        infoWindow.setContent(html);
        infoWindow.open(map, marker);
    });
}
```

- Dưới đây là toàn bộ mã kết hợp các yếu tố trên tạo thành 1 ứng dụng hoàn chỉnh:

```
<!DOCTYPE html >
<head>
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
    <meta http-equiv="content-type" content="text/html; charset=UTF-8"/>
    <title>PHP/MySQL & Google Maps Example</title>
    <script type="text/javascript"
src="http://maps.googleapis.com/maps/api/js?sensor=false"></script>
    <script type="text/javascript">
    //

    var customIcons = {
        nhahang: {
            icon: 'http://labs.google.com/ridefinder/images/mm_20_blue.png',
            shadow: 'http://labs.google.com/ridefinder/images/mm_20_shadow.png'
        },
        karaoke: {
            icon: 'http://labs.google.com/ridefinder/images/mm_20_red.png',
            shadow: 'http://labs.google.com/ridefinder/images/mm_20_shadow.png'
        }
    };

    function load() {
        var map = new google.maps.Map(document.getElementById("map"), {
            center: new google.maps.LatLng(20.997663,105.842628),
            zoom: 14,
            mapTypeId: 'roadmap'
        });

        var infoWindow = new google.maps.InfoWindow;
        downloadUrl("xuat_xml.php", function(data) {
            var xml = data.responseXML;
            var markers = xml.documentElement.getElementsByTagName("point");
            for (var i = 0; i &lt; markers.length; i++) {
                var name = markers[i].getAttribute("Ten");
                var address = markers[i].getAttribute("address");
                var type = markers[i].getAttribute("type");
                var point = new google.maps.LatLng(
                    parseFloat(markers[i].getAttribute("lat")),
                    parseFloat(markers[i].getAttribute("lng")));
                var html = "&lt;b&gt;" + name + "&lt;/b&gt; &lt;br/&gt;" + address;
                var icon = customIcons[type] || {};
                var marker = new google.maps.Marker({
                    map: map,
                    position: point,
                    icon: icon.icon,
                    shadow: icon.shadow
                });
            }
        });
    }
    ]]]&gt;
&lt;/script&gt;
&lt;/head&gt;
&lt;body&gt;
    &lt;div id="map"&gt;
    &lt;/div&gt;
&lt;/body&gt;
&lt;/html&gt;</pre>
</div>
```

```

        bindInfoWindow(marker, map, infoWindow, html);
    }
});
}

function bindInfoWindow(marker, map, infoWindow, html) {
    google.maps.event.addListener(marker, 'click', function() {
        infoWindow.setContent(html);
        infoWindow.open(map, marker);
    });
}

function downloadUrl(url, callback) {
    var request = window.ActiveXObject ?
        new ActiveXObject('Microsoft.XMLHTTP') :
        new XMLHttpRequest;

    request.onreadystatechange = function() {
        if (request.readyState == 4) {
            request.onreadystatechange = doNothing;
            callback(request, request.status);
        }
    };

    request.open('GET', url, true);
    request.send(null);
}

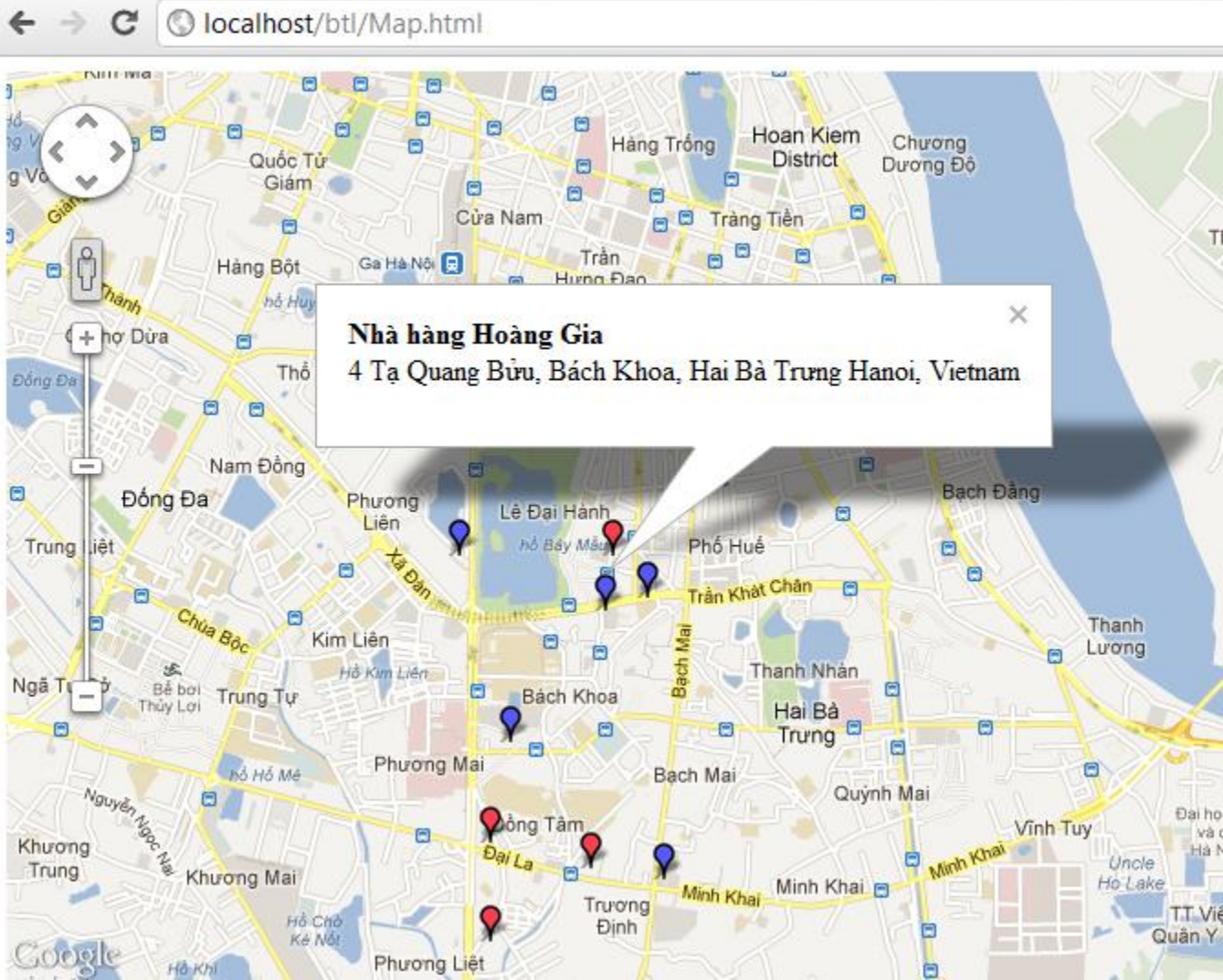
function doNothing() {}
</script>

</head>

<body onload="load()">
    <div id="map" style="width: 960px; height: 500px"></div>
</body>

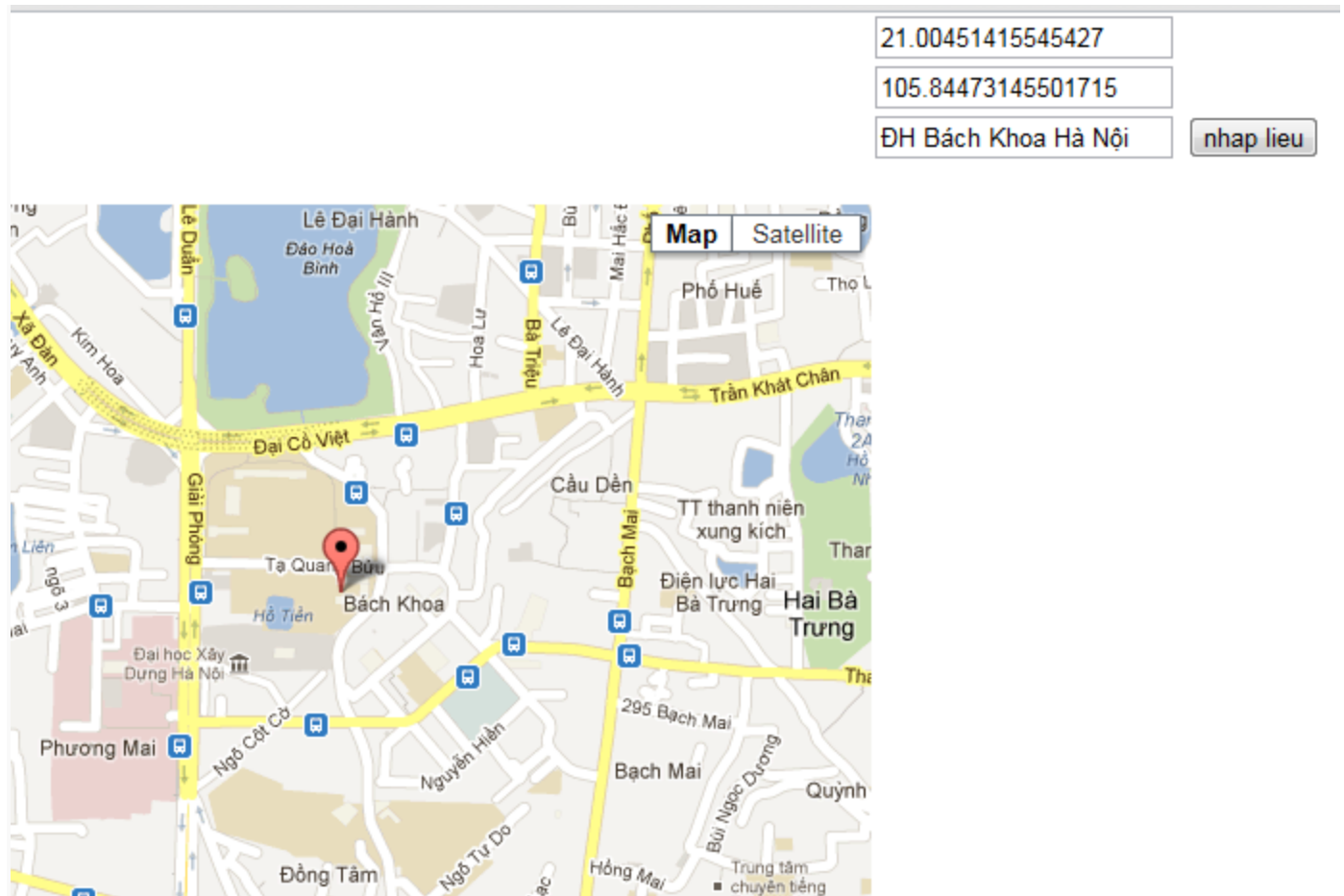
</html>

```

- 
- Bạn hãy thử làm một cái sẽ thấy khá thú vị!

4.4 Hướng phát triển của ứng dụng

- Các dữ liệu trên chỉ mang tính minh họa.
- Với ứng dụng trên sẽ giúp ích cho người sử dụng dễ dàng tìm được địa điểm như nhà hàng hay một quán karaoke nào đó để thỏa mãn nhu cầu của mình. Hoặc những người chủ của các cửa hàng hay các quán karaoke muốn cho người nào có nhu cầu sử dụng tìm đến dịch vụ của họ, thì chỉ việc post địa điểm của mình lên ứng dụng Map này. Lưu ý rằng google map không hề cho các địa điểm đó lên Map của họ. Và từ đó người xây dựng ứng dụng sẽ được một khoản chi phí nhỏ nhờ từ những người chủ này.
- Có thể bạn đã từng đi đến một địa điểm nào đó và rất thích nó không chỉ là nhà hàng hay quán karaoke. Mà bất cứ địa điểm nào bạn lưu địa điểm của nó lên trên Map để ghi nhớ cho những lần sau nếu muốn đến lần nữa.
- Để làm được điều này bạn hãy thử làm thêm trong ứng dụng một chức năng đánh dấu trên bản đồ mà tôi đã hướng dẫn ở trên. Và nhập các thông tin cần thiết. Với các tọa độ lấy trực tiếp trên bản đồ. Sau đó submit vào cơ sở dữ liệu. Các điểm đánh dấu sẽ tự động cập nhật. Tôi đã làm thử một chức năng nhập liệu đơn giản như hình ảnh sau:



- Như vậy với các điểm đã được đánh dấu người sử dụng muốn đến đâu chỉ cần mở ứng dụng tra cứu mà không cần phải ghi nhớ trong đầu.
- Có thể bạn không thông thuộc đường đi đến các điểm đó. Chúng ta lại xây dựng thêm một ứng dụng là tìm đường đi đến các điểm đánh dấu sử dụng Service của google. Với dịch vụ direction tôi đã trình bày ở trên thì điều đó cũng thật là dễ dàng!
- Ứng dụng sẽ chia sẻ các địa điểm được đánh dấu trên bản đồ với tất cả mọi người. Và cộng đồng tham gia sẽ dần dần tăng nhanh theo thời gian.
- Ngày nay với sự phát triển của các thiết bị di động với thiết bị định vị GPS để làm được điều này cũng thật đơn giản, và tiện lợi. Google Map API v3 hỗ trợ trên cả di động do vậy ứng dụng có thể phát triển trên cả các thiết bị này.
- Như vậy hướng phát triển và tương lai của ứng dụng cũng rất lớn mặc dù tôi chỉ mới đưa ra một vài ví dụ nhỏ. Vậy nếu hứng thú hãy tham gia vào các nhà phát triển Google Map API.
- Rất mong được sự đóng góp hoặc có thông tin trao đổi xin liên lạc qua địa chỉ: datnv030691@gmail.com

5. Tài liệu tham khảo

- Document tutorial của các ngôn ngữ hay thiết bị tìm kiếm tại: <https://developers.google.com/maps/>
- Hoặc tutorial phổ biến nhất: <https://developers.google.com/maps/documentation/javascript/>