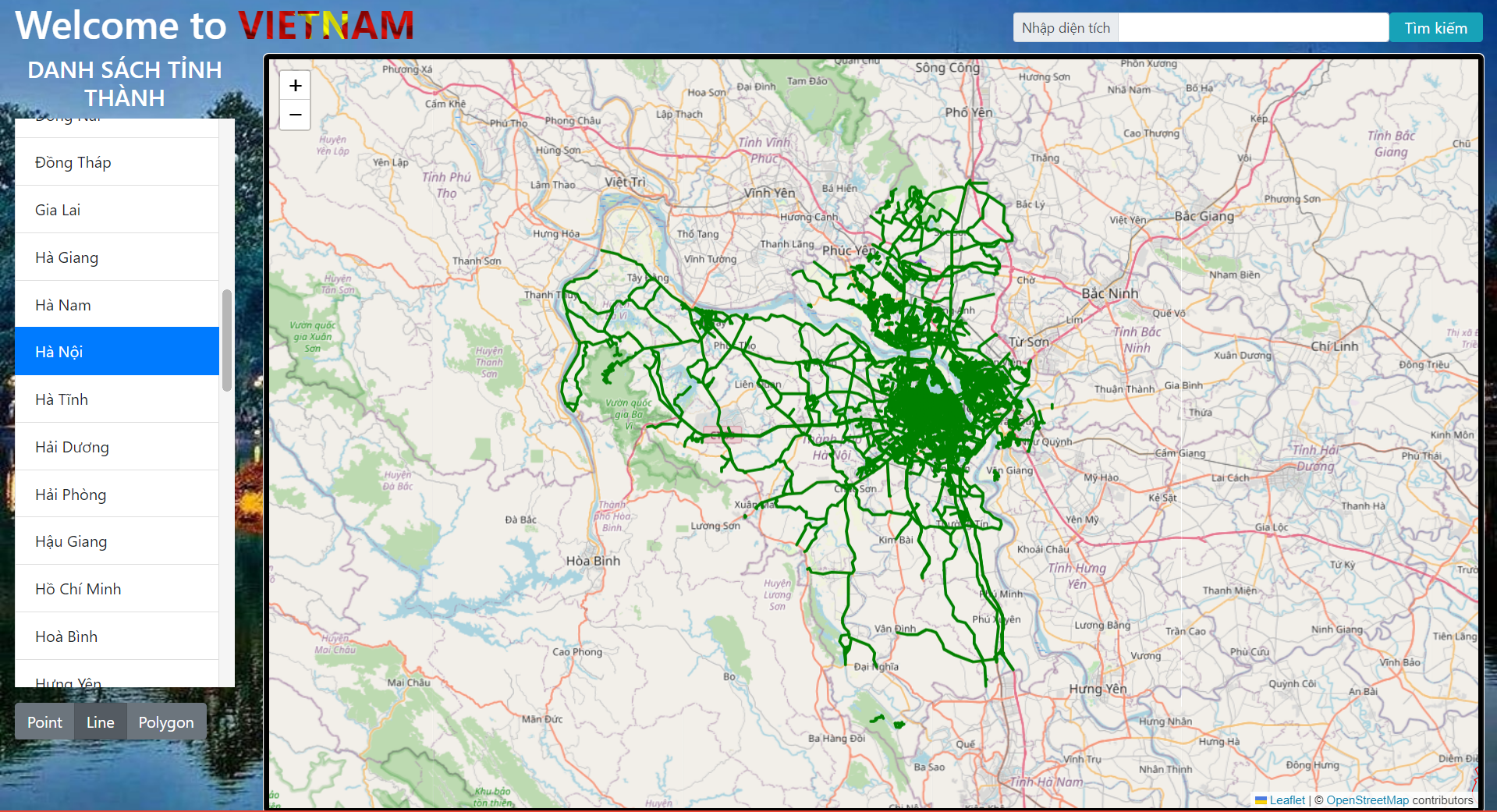
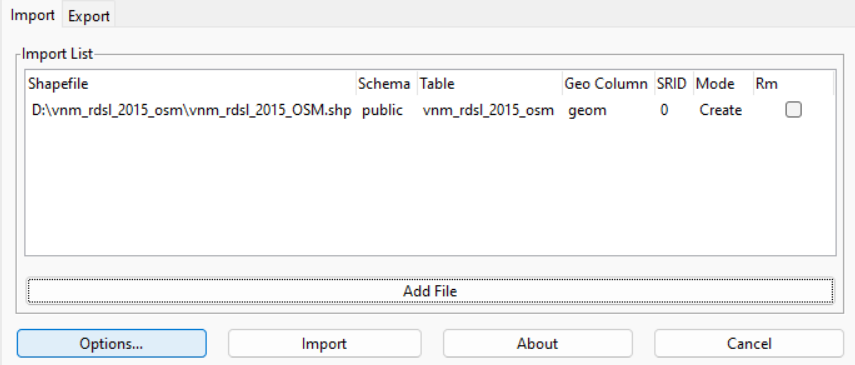
Đọc dữ liệu dạng đường không gian từ PostGreSQL và hiển thị trên bản đồ với Laravel

(Nối tiếp Job 30)



# 1. Chuẩn bị dữ liệu

* Tải xuống dữ liệu roads dạng Shapefile của Việt Nam [tại đây](https://data.humdata.org/dataset/viet-nam-roads?).
* Import bảng đó vào PostgreSQL



# 2. Điều hướng

Tại file **routes\web.php** thêm dòng code cưới đây

| Route::get('/get-line', [HomeController::class, 'getLine']);  Route::get('/get-highlightLine', [HomeController::class, 'highlightLine\_click']);  Route::get('/get-distance', [HomeController::class, 'getDistance']); |
| --- |

# 3. Controller

Tại file **app\Http\Controllers\HomeController.php** thêm hàm dưới đây

| public function getLine(Request $request)  {  $lat = $request->input('lat');  $lng = $request->input('lon');  $roads = DB::select("  SELECT g.name\_1, ST\_ASGeoJSON(o.geom) AS road  FROM vnm\_rdsl\_2015\_osm AS o  JOIN gadm41\_VNM\_1 AS g ON ST\_Within(o.geom, g.geom)  WHERE ST\_Within(ST\_MakePoint($lng, $lat), g.geom)  ");    return response()->json($roads);  }  public function highlightLine\_click(Request $request)  {  $lat = $request->input('lat');  $lng = $request->input('lon');  $roads = DB::select("  SELECT ST\_ASGeoJSON(geom) AS road  FROM vnm\_rdsl\_2015\_osm  WHERE ST\_Intersects(geom, ST\_Buffer(ST\_GeomFromText('POINT($lng $lat)'), 0.0001));  ");    return response()->json($roads);  }  public function getDistance(Request $request)  {  $distance = $request->input('distance');  $roads = DB::select("  SELECT ST\_ASGeoJSON(geom) AS road  FROM vnm\_rdsl\_2015\_osm  where shape\_leng >= $distance  ");    return response()->json($roads);  } |
| --- |

# 4. Mã nguồn JS

## 4.1. Khởi tạo

Lớp overlay để vẽ dữ liệu dạng đường và phần highlight

| var roadsLayerGroup = L.layerGroup().addTo(mymap);  var LineHighlightLayerGroup = L.layerGroup().addTo(mymap); |
| --- |

Tọa độ 63 tỉnh thành Việt Nam:

| var coordinatesVietnamProvinces = {  'Hà Nội' : { lat: 21.0285, lon: 105.8542},  'Hồ Chí Minh' : { lat: 10.8231, lon: 106.6297},  'Đà Nẵng' : { lat: 16.0544, lon: 108.2022},  'Hải Phòng' : { lat: 20.8627, lon: 106.6837},  'Cần Thơ' : { lat: 10.0451, lon: 105.7469},  'An Giang' : { lat: 10.5372, lon: 105.1924},  'Bà Rịa - Vũng Tàu' : { lat: 10.4601, lon: 107.1709},  'Bạc Liêu' : { lat: 9.2850, lon: 105.7246},  'Bắc Kạn' : { lat: 22.1385, lon: 105.8077},  'Bắc Giang' : { lat: 21.2715, lon: 106.1946},  'Bắc Ninh' : { lat: 21.1860, lon: 106.0560},  'Bến Tre' : { lat: 10.2354, lon: 106.3753},  'Bình Dương' : { lat: 11.1665, lon: 106.5304},  'Bình Định' : { lat: 13.9454, lon: 108.1265},  'Bình Phước' : { lat: 11.7519, lon: 106.8860},  'Bình Thuận' : { lat: 10.9330, lon: 108.0700},  'Cà Mau' : { lat: 9.1769, lon: 105.1500},  'Cao Bằng' : { lat: 22.6664, lon: 106.2519},  'Đắk Lắk' : { lat: 12.6654, lon: 108.0365},  'Đắk Nông' : { lat: 12.0311, lon: 107.6910},  'Điện Biên' : { lat: 21.3926, lon: 103.0072},  'Đồng Nai' : { lat: 10.9465, lon: 107.1485},  'Đồng Tháp' : { lat: 10.5120, lon: 105.0976},  'Gia Lai' : { lat: 13.9750, lon: 108.2081},  'Hà Giang' : { lat: 22.8337, lon: 104.9833},  'Hà Nam' : { lat: 20.5420, lon: 105.9227},  'Hà Tĩnh' : { lat: 18.3428, lon: 105.9057},  'Hải Dương' : { lat: 20.9375, lon: 106.3206},  'Hậu Giang' : { lat: 9.7779, lon: 105.5928},  'Hòa Bình' : { lat: 20.6764, lon: 105.3397},  'Hưng Yên' : { lat: 20.6463, lon: 106.0519},  'Khánh Hòa' : { lat: 12.2388, lon: 109.1963},  'Kiên Giang' : { lat: 10.1377, lon: 105.1119},  'Kon Tum' : { lat: 14.3505, lon: 107.9619},  'Lai Châu' : { lat: 22.3964, lon: 103.4583},  'Lâm Đồng' : { lat: 11.9325, lon: 108.4306},  'Lạng Sơn' : { lat: 21.8500, lon: 106.7500},  'Lào Cai' : { lat: 22.4167, lon: 103.9833},  'Long An' : { lat: 10.7400, lon: 106.2500},  'Nam Định' : { lat: 20.4330, lon: 106.1771},  'Nghệ An' : { lat: 19.6258, lon: 105.6121},  'Ninh Bình' : { lat: 20.2525, lon: 105.9750},  'Ninh Thuận' : { lat: 11.7666, lon: 108.3574},  'Phú Thọ' : { lat: 21.4241, lon: 105.2351},  'Phú Yên' : { lat: 13.1113, lon: 109.1126},  'Quảng Bình' : { lat: 17.5367, lon: 106.2405},  'Quảng Nam' : { lat: 15.8700, lon: 108.3500},  'Quảng Ngãi' : { lat: 15.1219, lon: 108.8000},  'Quảng Ninh' : { lat: 21.0167, lon: 107.2500},  'Quảng Trị' : { lat: 16.6667, lon: 107.2500},  'Sóc Trăng' : { lat: 9.5984, lon: 105.9781},  'Sơn La' : { lat: 21.1667, lon: 104.1667},  'Tây Ninh' : { lat: 11.3029, lon: 106.0996},  'Thái Bình' : { lat: 20.4425, lon: 106.3378},  'Thái Nguyên' : { lat: 21.5928, lon: 105.8442},  'Thanh Hóa' : { lat: 19.8000, lon: 105.7667},  'Thừa Thiên Huế' : { lat: 16.4667, lon: 107.6000},  'Tiền Giang' : { lat: 10.3749, lon: 106.3460},  'Trà Vinh' : { lat: 9.9345, lon: 106.3460},  'Tuyên Quang' : { lat: 21.8167, lon: 105.2167},  'Vĩnh Long' : { lat: 10.2500, lon: 105.9667},  'Vĩnh Phúc' : { lat: 21.3086, lon: 105.6044},  'Yên Bái' : { lat: 21.7000, lon: 104.8500}  }; |
| --- |

## 4.2. Xử lý và vẽ dữ liệu dạng đường

### Tìm đến phần dưới đây và thêm phần in đậm như bên dưới

| $('#Provinces\_List').on('click', 'li', function(event) {  var target = $(this).text();  var pointRadioButton = $('#Point');  var lineRadioButton = $('#Line');  var polygonRadioButton = $('#Polygon');  if (polygonRadioButton.is(':checked')) {  renderPolygon(target);  } **else if (lineRadioButton.is(':checked')) {**  **renderLine(target);**  **}** else if (pointRadioButton.is(':checked')) {  renderPoint(target);  }  });  $('input[name="GIS\_Type"]').change(function() {  var selectedOption = $(this).val(); // Di chuyển khai báo biến vào đây  $('label.btn').removeClass('active');  $(this).closest('label.btn').addClass('active');  **$('#AreaInputGroup').addClass('d-none');**  **$('#LineInputGroup').addClass('d-none');**  **if (selectedOption === 'Polygon') {**  **$('#AreaInputGroup').removeClass('d-none');**  **} else if (selectedOption === 'Line') {**  **$('#LineInputGroup').removeClass('d-none');**  **}**  });  mymap.on('click', function(e) {  if ($('#Polygon').is(':checked')) {  provincesLayerGroup.clearLayers();  roadsLayerGroup.clearLayers();  getData(e.latlng.lat, e.latlng.lng);  } **else if ($('#Line').is(':checked')) {**  **provincesLayerGroup.clearLayers();**  **LineHighlightLayerGroup.clearLayers();**  **highlightLine\_click(e.latlng.lat, e.latlng.lng);**  **}**  }); |
| --- |

### Tính diện tích và khoảng cách

| // Kiểm tra và hạn chế nhập liệu cho ô nhập diện tích:  $("#SearchArea").click(function() {  var inputValue = $("#AreaInput").val();  searchArea(inputValue);  });  $("#SearchDistance").click(function() {  var inputValue = $("#DistanceInput").val();  searchDistance(inputValue);  }); |
| --- |

### Hàm renderLine

| function renderLine(target) {  provincesLayerGroup.clearLayers();  roadsLayerGroup.clearLayers();  var provinceCoordinates = coordinatesVietnamProvinces[target];  $.ajax({  url: '/get-line',  type: 'GET',  data: {  lat: provinceCoordinates.lat,  lon: provinceCoordinates.lon  },  success: function(response) {  response.forEach(function(item) {  var Roads = JSON.parse(item.road);  console.log(Roads);  const roadCoordinates = Roads.coordinates;  roadCoordinates.forEach(function(coords) {  // Khai báo một mảng mới để lưu trữ tọa độ của polyline  var polylineCoords = [];  // Lặp qua từng cặp tọa độ trong tập hợp coords  for (var coord of coords) {  // Đảo vị trí tọa độ (dữ liệu trả về dạng [long, lat] nhưng Leaflet cần [lat, long])  polylineCoords.push([coord[1], coord[0]]);  }  // Tạo một polyline từ tọa độ và thêm nó vào roadsLayerGroup  L.polyline(polylineCoords, {  color: 'green',  weight: 3  }).addTo(roadsLayerGroup);  });  });  },  error: function(xhr) {  console.log('Error:', xhr);  }  });  } |
| --- |

### Xử lý higlight

| // Hàm xử lý highlight khi click vào đường nào đó  function highlightLine\_click(lat, lng) {  $.ajax({  url: '/get-highlightLine',  type: 'GET',  data: {  lat: lat,  lon: lng  },  success: function(response) {  response.forEach(function(item) {  var Roads = JSON.parse(item.road);  console.log(Roads);  const roadCoordinates = Roads.coordinates;  roadCoordinates.forEach(function(coords) {  // Khai báo một mảng mới để lưu trữ tọa độ của polyline  var polylineCoords = [];  // Lặp qua từng cặp tọa độ trong tập hợp coords  for (var coord of coords) {  // Đảo vị trí tọa độ (dữ liệu trả về dạng [long, lat] nhưng Leaflet cần [lat, long])  polylineCoords.push([coord[1], coord[0]]);  }  L.polyline(polylineCoords, {  color: 'yellow',  weight: 4  }).addTo(LineHighlightLayerGroup);  });  });  },  error: function(xhr) {  console.log('Error:', xhr);  }  });  } |
| --- |

### Xử lý hight khoảng cách

| // Xử lý tô màu đường có khoảng cách >= input  function searchDistance(inputValue) {  provincesLayerGroup.clearLayers();  LineHighlightLayerGroup.clearLayers();  roadsLayerGroup.clearLayers();  $.ajax({  url: '/get-distance',  type: 'GET',  data: {  distance: inputValue,  },  success: function(response) {  response.forEach(function(item) {  var Roads = JSON.parse(item.road);  console.log(Roads);  const roadCoordinates = Roads.coordinates;  roadCoordinates.forEach(function(coords) {  // Khai báo một mảng mới để lưu trữ tọa độ của polyline  var polylineCoords = [];  // Lặp qua từng cặp tọa độ trong tập hợp coords  for (var coord of coords) {  // Đảo vị trí tọa độ (dữ liệu trả về dạng [long, lat] nhưng Leaflet cần [lat, long])  polylineCoords.push([coord[1], coord[0]]);  }  L.polyline(polylineCoords, {  color: 'blue',  weight: 4  }).addTo(roadsLayerGroup);  });  });  },  error: function(xhr) {  console.log('Error:', xhr);  }  });  } |
| --- |