# Geographical Feature Development Document

Fengyang Lin

07/28/2020

# app/lib/ctgov.py

Modify the return line

|  |
| --- |
| def get\_nct\_list\_from\_zip(input\_zip, mile\_range=50): ... # fengyang: also return the nearby zipcode list(change the return line) return [temp\_return\_list, zip\_list] |

Add a new function get\_nct\_location\_latlng

|  |
| --- |
| # fengyang: add functions to get location latlng import pickle nct\_locs\_dict = pickle.load(open('../dquest-flask/app/resources/nct\_latlng\_dict.pkl', 'rb')) def get\_nct\_location\_latlng(nct\_id):  nct\_loc\_latlng = []  # if no valid latlng, Object.keys(nct\_loc\_latlng).length == 0, else 2  if nct\_id in nct\_locs\_dict.keys():  # if loc\_index <= len(nct\_locs\_dict[nct\_id])-1:  nct\_loc\_latlng = nct\_locs\_dict[nct\_id]  # [{lat: 40.8476284, lng: -73.8360251}]  return nct\_loc\_latlng |

# app/lib/question\_info\_entropy.py

Add a new function find\_all\_locations\_on\_current\_page

|  |
| --- |
| # fengyang: find all locations on current page import ctgov as ctgov def find\_all\_locations\_on\_current\_page(working\_nct\_id\_list, npag):  '''  find nct locations on current page by connecting to knowledgebase  :param working\_nct\_id\_list:  :param npag: page number of result to return  :return: this\_page\_loc\_flat\_list is a list of location latlng dict [{'lat':, 'lng': }, ...]  nct\_id\_this\_page\_list is a list of nct\_id corresponding to the location in this\_page\_loc\_flat\_list  '''  this\_page\_loc\_flat\_list = []   working\_nct\_id\_0 = [(record[1], record[2]) for record in working\_nct\_id\_list if record[3] == 0]  srt\_working\_nct\_id\_0 = sorted(working\_nct\_id\_0, key=lambda x: x[1], reverse=False)  start\_idx = (npag - 1) \* 20  end\_idx = min(len(srt\_working\_nct\_id\_0), npag \* 20)  nct\_id\_this\_page = [srt[0] for srt in srt\_working\_nct\_id\_0[start\_idx:end\_idx]]  nct\_id\_this\_page = list(set(nct\_id\_this\_page))  nct\_id\_rank = {}  for srt in srt\_working\_nct\_id\_0[start\_idx:end\_idx]:  nct\_id\_rank[srt[0]] = srt[1]   if len(nct\_id\_this\_page) == 0:  this\_page\_loc\_flat\_list = []  nct\_id\_this\_page\_list = []  else:  nct\_id\_this\_page = [str(x) for x in nct\_id\_this\_page]   this\_page\_loc\_latlng\_list = []  nct\_id\_this\_page\_list = []  for nct\_id in nct\_id\_this\_page:  trial\_loc\_latlng\_list = ctgov.get\_nct\_location\_latlng(nct\_id)  this\_page\_loc\_latlng\_list.append(trial\_loc\_latlng\_list)  nct\_id\_this\_page\_list = nct\_id\_this\_page\_list + [nct\_id]\*len(trial\_loc\_latlng\_list)   # flat list of lists into one single list  this\_page\_loc\_flat\_list = [item for sublist in this\_page\_loc\_latlng\_list for item in sublist]   return [this\_page\_loc\_flat\_list, nct\_id\_this\_page\_list] |

Add 2 functions haversine

find\_nct\_loc\_within\_range

|  |
| --- |
| # fengyang: functions to find all location within search range  from math import radians, cos, sin, asin, sqrt import numpy as np  def haversine(lon1, lat1, lon2, lat2):  """  Calculate the great circle distance between two points  on the earth (specified in decimal degrees)  """  # convert decimal degrees to radians  lon1, lat1, lon2, lat2 = map(radians, [lon1, lat1, lon2, lat2])   # haversine formula  dlon = lon2 - lon1  dlat = lat2 - lat1  a = sin(dlat/2)\*\*2 + cos(lat1) \* cos(lat2) \* sin(dlon/2)\*\*2  c = 2 \* asin(sqrt(a))  # r = 6371 # Radius of earth in kilometers. Use 3956 for miles  r = 3956  return c \* r  def find\_nct\_loc\_within\_range(loc\_latlng\_list, nct\_id\_list, input\_zipcode\_latlng, input\_miles):  '''  :param loc\_latlng\_list: [{'lat': , 'lon': }]  :param input\_zipcode\_latlng: {'lat': , 'lon': }  :param input\_miles: number  :return:  '''  lat1 = input\_zipcode\_latlng['lat']  lng1 = input\_zipcode\_latlng['lng']  # allow distance error +3  distance\_list = [haversine(lng1, lat1, loc['lng'], loc['lat'])+3 if loc is not None else 999999  for loc in loc\_latlng\_list]  filter\_index = np.where(np.array(distance\_list) <= input\_miles)[0]  # loc\_within\_range = np.array(loc\_latlng\_list)[np.where(np.array(distance\_list) <= input\_miles)[0]]  loc\_within\_range = np.array(loc\_latlng\_list)[filter\_index]  loc\_within\_range = list(loc\_within\_range)  nct\_within\_range = np.array(nct\_id\_list)[filter\_index]  nct\_within\_range = list(nct\_within\_range)   # get trial title  conn = general\_pool\_criteria.connection()  cur = conn.cursor()  sql = '''  select nct\_id, official\_title  from dbo.aact\_trial\_info  '''  cur.execute(sql)  trial\_title = cur.fetchall()  conn.close()  cur.close()  nct\_id\_title = {}  if len(trial\_title) > 0:  for r in trial\_title:  if len(r) >=2:  nct\_id\_title[r[0]] = r[1]  nct\_title\_within\_range = [nct\_id\_title[nct\_id] for nct\_id in nct\_within\_range]   return [loc\_within\_range, nct\_within\_range, nct\_title\_within\_range] |

# app/views.py

## Modify a few lines (in red)

|  |
| --- |
| # small function to make sure input zip code is valid and found in list @app.route('/\_check\_homepage\_parameters') def check\_homepage\_parameters():  ...   print('keyword in views: ' + keyword\_search)   # trying zipcode search  try:  # fengyang: after change the output of get\_nct\_list\_from\_zip function  nearby\_nct = ctgov.get\_nct\_list\_from\_zip(locn, miles)[0]  print('length of nearby nct: ' + str(len(nearby\_nct)))  # session['nearby\_nct\_list'] = nearby\_nct  valid\_zip = True  print('assigned new variable')  # fengyang: return nearby zip list  nearby\_zip\_list = ctgov.get\_nct\_list\_from\_zip(locn, miles)[1]   except:  nearby\_nct = []  valid\_zip = False   ...   # fengyang: add nearby\_zip\_list at return output  return jsonify(valid\_zip=valid\_zip, nearby\_length=len(nearby\_nct), keyword\_length=len(klist), type\_length=len(tlist),  working\_nct\_length=len(zanct), nearby\_zip\_list = nearby\_zip\_list) |

## Add lines to return latlng info

|  |
| --- |
| # function to query aact\_trial\_info table and return data for trial information modal @app.route('/\_retrieve\_modal\_data') def retrieve\_modal\_data():  nct\_id = request.args.get('nctid')   # in case of error, returns string 'Error' in location 1  trial\_data = ctgov.query\_trial\_info\_for\_modal(nct\_id)   # simple post-processing on facility and facility contact information  fc = []  print(trial\_data[15])  if trial\_data[15] is not None: # what would be returned if no columns had values  for item in trial\_data[15].split('|'):  fc.append(item.split('; '))   # splitting central contact information  cc = []  print(trial\_data[17])  if trial\_data[17] is not None:  cc = (trial\_data[17].split('; '))   # fengyang: add location latlng info  trial\_loc\_latlng\_list = ctgov.get\_nct\_location\_latlng(nct\_id)  # if nct\_id is not in the pickle file, will return an empty list []   return jsonify(official\_title=trial\_data[0], study\_type=trial\_data[1], primary\_purpose=trial\_data[2],  study\_description=trial\_data[3], gender=trial\_data[4], minimum\_age=trial\_data[5],  maximum\_age=trial\_data[6], healthy\_volunteers=trial\_data[7], phase=trial\_data[8],  allocation=trial\_data[9], intervention\_model=trial\_data[10], observation\_model=trial\_data[11],  masking=trial\_data[12], outcome\_measure=trial\_data[13], outcome\_description=trial\_data[14],  facilities\_and\_contacts=fc, intervention\_names=trial\_data[16], central\_contact=cc,  # fengyang  location\_latlng=trial\_loc\_latlng\_list) |

## Add a few lines to return latlng for nearby latlng for nearby markers

|  |
| --- |
| # load nct details. @app.route('/\_find\_nct\_by\_page', methods=['POST']) def find\_nct\_by\_page():  requestion\_dict = request.get\_json()  working\_nct\_id\_list = requestion\_dict['working\_nct\_id\_list']  npag = requestion\_dict['npag']  nct\_details\_for\_this\_page = qst.find\_nct\_details(working\_nct\_id\_list, npag)  print('find\_nct\_by\_page: ' + str(working\_nct\_id\_list))  size\_of\_active\_trials = qst.find\_size\_of\_active\_trials(working\_nct\_id\_list)  print('size\_of\_active\_trials: ' + str(size\_of\_active\_trials))  # fengyang  nct\_loc\_for\_this\_page = qst.find\_all\_locations\_on\_current\_page(working\_nct\_id\_list, npag)[0]  nct\_id\_rep\_for\_this\_page = qst.find\_all\_locations\_on\_current\_page(working\_nct\_id\_list, npag)[1]  # print('len\_of\_all\_marker: '+ str(len(nct\_loc\_for\_this\_page)))  # nct\_log\_nearby  input\_latlng = requestion\_dict['input\_zipcode\_latlng']  input\_miles = requestion\_dict['input\_miles']  nct\_loc\_nearby\_for\_this\_page = qst.find\_nct\_loc\_within\_range(nct\_loc\_for\_this\_page, nct\_id\_rep\_for\_this\_page,input\_latlng, input\_miles)[0]  nct\_id\_nearby\_for\_this\_page = qst.find\_nct\_loc\_within\_range(nct\_loc\_for\_this\_page, nct\_id\_rep\_for\_this\_page,input\_latlng, input\_miles)[1]  nct\_title\_nearby\_for\_this\_page = \  qst.find\_nct\_loc\_within\_range(nct\_loc\_for\_this\_page, nct\_id\_rep\_for\_this\_page, input\_latlng, input\_miles)[2]   return jsonify(working\_nct\_id\_list=working\_nct\_id\_list, npag=npag,  nct\_details\_for\_this\_page=nct\_details\_for\_this\_page, size\_of\_active\_trials=size\_of\_active\_trials,  # fengyang  nct\_loc\_for\_this\_page = nct\_loc\_for\_this\_page,  nct\_loc\_nearby\_for\_this\_page = nct\_loc\_nearby\_for\_this\_page,  nct\_id\_nearby\_for\_this\_page = nct\_id\_nearby\_for\_this\_page,  nct\_title\_nearby\_for\_this\_page=nct\_title\_nearby\_for\_this\_page,  ) |

# app/static/js/scripts.js

## Add functions and variables for map

|  |
| --- |
| // fengyang: map start var map; var geocoder; var infowindow; var marker\_list = []; var input\_miles\_list = []; var input\_latlng = [];  function initMap() {  map = new google.maps.Map(document.getElementById('map'), {  zoom: 9,  center: {lat: 40.8412426, lng: -73.9431593},  mapTypeControl: false  });  geocoder = new google.maps.Geocoder();  infowindow = new google.maps.InfoWindow({maxWidth: 200});  document.getElementById('submit').addEventListener('click', function() {  geocodeAddress(geocoder, map);  }); } |

|  |
| --- |
| function geocodeAddress(geocoder, resultsMap) {  var address = document.getElementById('address').value;  geocoder.geocode({'address': address}, function(results, status) {  if (status === 'OK') {  resultsMap.setCenter(results[0].geometry.location);  change\_previous\_marker\_icon(marker\_list);  let marker = new google.maps.Marker({  map: resultsMap,  position: results[0].geometry.location  });  marker\_list.push(marker);  } else {  alert('Geocode was not successful for the following reason: ' + status);  }  }); }   function add\_marker\_with\_address(geocoder, resultsMap, address, info\_content='', icon\_url = '') {  // <!--var address = document.getElementById('address').value;-->  geocoder.geocode({'address': address}, function(results, status) {  if (status === 'OK') {  resultsMap.setCenter(results[0].geometry.location);  // change\_previous\_marker\_icon(marker\_list);  let marker = new google.maps.Marker({  map: resultsMap,  position: results[0].geometry.location  });  if (icon\_url != ''){  marker.setIcon(icon\_url);}   // add infowindow  marker.addListener('click', function() {  infowindow.setContent(info\_content);  infowindow.open(resultsMap, marker);  });  marker\_list.push(marker);  } else {  alert('Geocode was not successful for the following reason: ' + status);  console.log('Geocode was not successful for the following reason: ' + status);  }  });  } // have to store it in a global variable, geocode is an asynchronous call  function geocode\_marker\_with\_address(geocoder, resultsMap, address, info\_content='', icon\_url = '') {  // functions similar to geocode marker, but only for input value on the index page  // let latlng\_dict\_return = [];  geocoder.geocode({'address': address}, function(results, status) {  if (status === 'OK') {  resultsMap.setCenter(results[0].geometry.location);  // change\_previous\_marker\_icon(marker\_list);  let marker = new google.maps.Marker({  map: resultsMap,  position: results[0].geometry.location  });  if (icon\_url != ''){  marker.setIcon(icon\_url);}  marker.addListener('click', function() {  infowindow.setContent(info\_content);  infowindow.open(resultsMap, marker);  });  marker\_list.push(marker);   latlng\_dict = {  'lat': results[0].geometry.location.lat(),  'lng': results[0].geometry.location.lng()  };  input\_latlng.push(latlng\_dict);  } else {  alert('Geocode was not successful for the following reason: ' + status);  console.log('Geocode was not successful for the following reason: ' + status);  }  });  }  function add\_marker\_with\_latlng(resultsMap, latlng, info\_content='', icon\_url = '',storeID = '',  icon\_size = '',  list\_to\_push = marker\_list) {  resultsMap.setCenter(latlng); //if setCenter or not  // change\_previous\_marker\_icon(marker\_list);  let marker = new google.maps.Marker({  map: resultsMap,  position: latlng,  storeId: storeID,  });  if (icon\_url != ''){  if(icon\_size != ''){  marker.setIcon({  url: icon\_url,  scaledSize: new google.maps.Size(icon\_size, icon\_size)})  } else{  marker.setIcon(icon\_url);  }  }   // add infowindow  // let marker\_index = marker\_list.length;  // marker content not define yet  if (info\_content != ''){  marker.addListener('click', function() {  infowindow.setContent(info\_content);  infowindow.open(resultsMap, marker);  });  }  list\_to\_push.push(marker);  }  function add\_10\_markers\_each\_time(to\_add\_marker\_list, to\_add\_info\_list, time\_sleep){  var num\_add\_times = Math.floor(to\_add\_marker\_list.length/10);  if (to\_add\_marker\_list.length % 10 != 0){num\_add\_times = num\_add\_times + 1};   for (let k = 0; k < num\_add\_times; k++) {  let current\_marker\_list = to\_add\_marker\_list.slice(k\*10, k\*10+10);  let current\_info\_list = to\_add\_info\_list.slice(k\*10, k\*10+10);  setTimeout(function(){  for (let j = 0; j < current\_marker\_list.length; j++) {  add\_marker\_with\_address(geocoder, map, current\_marker\_list[j], current\_info\_list[j]);  console.log('num\_add\_list\_' +k +'\_'+ current\_marker\_list.length);  }},  time\_sleep\*k)  }  }  function change\_previous\_marker\_icon(exist\_marker\_list){  for (var i = 0; i < exist\_marker\_list.length; i++) {  let marker = exist\_marker\_list[i];  // marker.setIcon("http://maps.google.com/mapfiles/ms/icons/green-dot.png");  marker.setIcon({  url: "http://maps.google.com/mapfiles/ms/icons/lightblue.png",  scaledSize: new google.maps.Size(20, 20)});  }  } function selectMarker(storeId, status) {  let i, len, marker;   // Find the correct marker to change based on the storeId.  for (i = 0, len = marker\_list.length; i < len; i++) {  if (marker\_list[i].storeId == storeId) {  marker = marker\_list[i];  if (status == "start") {  marker.setIcon({  url:"http://maps.google.com/mapfiles/ms/icons/ltblue-dot.png"  });  } else {  marker.setIcon({  url: "http://maps.google.com/mapfiles/ms/icons/lightblue.png",  scaledSize: new google.maps.Size(20, 20)});  }  }  } }  function clear\_marker(){  // fengyang: clear all markers before  if (marker\_list.length > 1){  marker\_list.slice(1).forEach(function(marker) {  marker.setMap(null);  });  marker\_list = marker\_list.slice(0, 1);  } }  // fengyang: map end |

## Find\_search\_results: Modify and add a few lines(in orange)

|  |
| --- |
| function find\_search\_results(working\_nct\_id\_list, np,  // fengyang: add para  input\_zipcode\_latlng = input\_latlng[0], input\_miles = input\_miles\_list[0]) {   formData = {  'working\_nct\_id\_list': working\_nct\_id\_list,  'npag': np,  'input\_zipcode\_latlng': input\_zipcode\_latlng,  'input\_miles': input\_miles  }  $.blockUI({  message: '<div class="ui segment"><div class="ui active dimmer">Loading...<div class="ui text loader"></div></div></div>',  css: {  border: 'none',  '-webkit-border-radius': '40px',  '-moz-border-radius': '40px',  opacity: .5,  },  });  $.ajax({  headers: {  'Accept': 'application/json',  'Content-Type': 'application/json'  },  type: 'POST',  url: $SCRIPT\_ROOT + '/\_find\_nct\_by\_page',  async: false,  data: JSON.stringify(formData),  dataType: "json",  success: function (data) { // format query  // sout = '<p class="recap"> Left <span class="drecap">' + data.size\_of\_active\_trials + '</span> clinical trials for: <span id="qlabel" class="drecap">' + data.q + '<span></p>';  show\_search\_results(data.working\_nct\_id\_list, data.npag, data.nct\_details\_for\_this\_page, data.size\_of\_active\_trials);  //fengyang: all markers to all location  let nearby\_loc\_latlng = data.nct\_loc\_nearby\_for\_this\_page;  let nearby\_loc\_nct\_id = data.nct\_id\_nearby\_for\_this\_page;  let nearby\_loc\_nct\_title= data.nct\_title\_nearby\_for\_this\_page;    let all\_loc\_latlng = data.nct\_loc\_for\_this\_page;  if (nearby\_loc\_latlng.length>0){  //if there are latlng records in pickle  for (let j = 0; j < nearby\_loc\_latlng.length; j++){  let this\_marker\_latlng = nearby\_loc\_latlng[j];  let this\_marker\_id = nearby\_loc\_nct\_id[j];  let info\_content = '<div>';  info\_content = info\_content + '<p>' + '<strong>' + 'Official Title' +'</strong>';  info\_content = info\_content + '<br />'+ nearby\_loc\_nct\_title[j] + '</p>';  info\_content = info\_content + '<p>'+ '<strong>' + 'NCT ID Number'+'</strong>';  info\_content = info\_content + '<br />' + this\_marker\_id + '</p>';  info\_content = info\_content + '</div>';  add\_marker\_with\_latlng(map, this\_marker\_latlng,info\_content,  icon\_url = 'http://maps.google.com/mapfiles/ms/icons/lightblue.png',  this\_marker\_id, icon\_size = 20,  );  }  add\_marker\_with\_latlng(map, input\_zipcode\_latlng, 'Your input zip code',  'http://maps.google.com/mapfiles/kml/pal4/icon57.png');  //add\_marker\_with\_address(geocoder, map, '10032',  // 'Your input zip code',  // 'http://maps.google.com/mapfiles/kml/pal4/icon57.png');  }   },  error: function (XMLHttpRequest, textStatus, errorThrown) {  console.log(errorThrown);  }  }); } |

## Show\_search\_results: Add clear\_markers() when page changes

|  |
| --- |
| function show\_search\_results(working\_nct\_id\_list, npag, nct\_details\_for\_this\_page) {  ...  // previous  if (np > 1) {  $('#rprev').unbind('click');  $('#rprev').bind('click', function () {  //fengyang: clear marker when page changes  clear\_marker();  ... } else {  $('#rprev').unbind('click')  }  // next  pmax = Math.ceil(parseInt(size\_of\_active\_trials) / 20);  if (np + 1 <= pmax) {  $('#rnext').unbind('click');  $('#rnext').bind('click', function () {  //fengyang: clear marker when page changes  clear\_marker();   find\_search\_results(working\_nct\_id\_list, parseInt(npag) + 1);   $(document).scrollTop(0);  });  } else {  $('#rnext').unbind('click')  }     $("#search\_results").children('.list').html(sout);   } |

## Similar to Find\_Search\_resutls and Show\_search\_results:

|  |
| --- |
| function find\_results(working\_nct\_id\_list, np,  input\_zipcode\_latlng = input\_latlng[0], input\_miles = input\_miles\_list[0]) {  // fengyang: should add functions to add all markers in this page  formData = {  'working\_nct\_id\_list': working\_nct\_id\_list,  'npag': np,  //fengyang: add params  'input\_zipcode\_latlng': input\_zipcode\_latlng,  'input\_miles': input\_miles,  }  $.blockUI({  message: '<div class="ui segment"><div class="ui active dimmer">Loading...<div class="ui text loader"></div></div></div>',  css: {  border: 'none',  '-webkit-border-radius': '40px',  '-moz-border-radius': '40px',  opacity: .5,  },  });  $.ajax({  headers: {  'Accept': 'application/json',  'Content-Type': 'application/json'  },  type: 'POST',  url: $SCRIPT\_ROOT + '/\_find\_nct\_by\_page',  data: JSON.stringify(formData),  dataType: "json",  success: function (data) { // format query  // sout = '<p class="recap"> Left <span class="drecap">' + data.size\_of\_active\_trials + '</span> clinical trials for: <span id="qlabel" class="drecap">' + data.q + '<span></p>';  // $('#filter\_header\_results').html(sout)  filter\_n = data.size\_of\_active\_trials  $("#filter\_n").html(filter\_n);  show\_qfilter\_results(data.working\_nct\_id\_list, data.npag, data.nct\_details\_for\_this\_page, data.size\_of\_active\_trials);   //fengyang  let nearby\_loc\_latlng = data.nct\_loc\_nearby\_for\_this\_page;  let nearby\_loc\_nct\_id = data.nct\_id\_nearby\_for\_this\_page;  let nearby\_loc\_nct\_title= data.nct\_title\_nearby\_for\_this\_page;  console.log('num nearby location:'+nearby\_loc\_latlng.length);  let all\_loc\_latlng = data.nct\_loc\_for\_this\_page;   if (nearby\_loc\_latlng.length>0){  //if there are latlng records in pickle  for (let j = 0; j < nearby\_loc\_latlng.length; j++){  let this\_marker\_latlng = nearby\_loc\_latlng[j];  let this\_marker\_id = nearby\_loc\_nct\_id[j];  let info\_content = '<div>';  info\_content = info\_content + '<p>' + '<strong>' + 'Official Title' +'</strong>';  info\_content = info\_content + '<br />'+ nearby\_loc\_nct\_title[j] + '</p>';  info\_content = info\_content + '<p>'+ '<strong>' + 'NCT ID Number'+'</strong>';  info\_content = info\_content + '<br />' + this\_marker\_id + '</p>';  info\_content = info\_content + '</div>';  // let this\_marker\_content = nearby\_loc\_nct\_title[j];  // add\_marker\_with\_latlng(map, this\_marker\_latlng,this\_marker\_content,  add\_marker\_with\_latlng(map, this\_marker\_latlng,info\_content,  icon\_url = 'http://maps.google.com/mapfiles/ms/icons/lightblue.png',  this\_marker\_id, icon\_size = 20,  );  }  add\_marker\_with\_latlng(map, input\_zipcode\_latlng, 'Your input zip code',  'http://maps.google.com/mapfiles/kml/pal4/icon57.png')  //add\_marker\_with\_address(geocoder, map, '10032',  // 'Your input zip code',  // 'http://maps.google.com/mapfiles/kml/pal4/icon57.png');  }  },  error: function (XMLHttpRequest, textStatus, errorThrown) {  console.log(errorThrown);  }  }); } |

|  |
| --- |
| function show\_qfilter\_results(working\_nct\_id\_list, npag, nct\_details\_for\_this\_page, size\_of\_active\_trials,  ) {  sout = result\_content(nct\_details\_for\_this\_page);  // sout += '<tr><td colspan="3"><p id="nav\_search">'  np = parseInt(npag);  ffirst = (np - 1) \* 20 + 1;  flast = np \* 20;  $('#ffirst').html(ffirst);  $('#flast').html(flast);  // previous  if (np > 1) {  $('#fprev').unbind('click');  $('#fprev').bind('click', function () {  //fengyang: clear markers before page changes  clear\_marker();  find\_results(working\_nct\_id\_list, parseInt(npag) - 1);  $(document).scrollTop(0);  });  } else {  $('#fprev').unbind('click')  }  // next  pmax = Math.ceil(parseInt(size\_of\_active\_trials) / 20);  if (np + 1 <= pmax) {  $('#fnext').unbind('click');  $('#fnext').bind('click', function () {  //fengyang: clear markers before page changes  clear\_marker();  // add para find\_results  find\_results(working\_nct\_id\_list, parseInt(npag) + 1);  $(document).scrollTop(0);  });  } else {  $('#fnext').unbind('click')  }   $("#filter\_results").children('.list').html(sout); } |

## Generate\_trial\_info\_modal: Add markers when click on the trial item

|  |
| --- |
| // function to handle modal display for trial information function generate\_trial\_info\_modal(nct\_id) {  console.log('clicked-on NCT ID: ' + nct\_id);   $.getJSON($SCRIPT\_ROOT + '/\_retrieve\_modal\_data', {  nctid: nct\_id  },  function (data) {  official\_title=data.official\_title;  study\_type = data.study\_type;  primary\_purpose = data.primary\_purpose;  study\_description = data.study\_description;  gender = data.gender;  minimum\_age = data.minimum\_age;  maximum\_age = data.maximum\_age;  healthy\_volunteers = data.healthy\_volunteers;  phase = data.phase;  allocation = data.allocation;  intervention\_model = data.intervention\_model;  observation\_model = data.observation\_model;  masking = data.masking;  outcome\_measure = data.outcome\_measure;  outcome\_description = data.outcome\_description;  fac\_and\_con = data.facilities\_and\_contacts;  interventions = data.intervention\_names;  central\_contact = data.central\_contact;  // fengyang: add latlng list  location\_latlng\_list = data.location\_latlng;  })   .done(function () {  // modifying top of the modal information  if (official\_title != null) {document.getElementById("modal\_official\_title").innerHTML = official\_title};  if (nct\_id != null) {document.getElementById("modal\_nct\_id").innerHTML = nct\_id};  if (study\_type != null) {document.getElementById("modal\_study\_type").innerHTML = study\_type};  if (primary\_purpose != null) {document.getElementById("modal\_primary\_purpose").innerHTML = primary\_purpose};  if (study\_description != null) {document.getElementById("modal\_study\_description").innerHTML = study\_description};  if (gender != null) {document.getElementById("modal\_gender").innerHTML = gender};  if (minimum\_age != null) {document.getElementById("modal\_min\_age").innerHTML = minimum\_age};  if (maximum\_age != null) {document.getElementById("modal\_max\_age").innerHTML = maximum\_age};  if (healthy\_volunteers != null) {document.getElementById("modal\_healthy\_volunteers").innerHTML = healthy\_volunteers};   // make some small adjustments based on study type  if (study\_type == 'Interventional') {  if (allocation != null) {document.getElementById("modal\_allocation").innerHTML = allocation};  if (masking != null) {document.getElementById("modal\_masking").innerHTML = masking};  if (interventions != null) {document.getElementById("modal\_intervention").innerHTML = interventions};  document.getElementById("modal\_int\_obs\_model\_title").innerHTML = "Intervention Model";  if (intervention\_model != null) {document.getElementById("modal\_int\_obs\_model\_description").innerHTML = intervention\_model};  } else {  document.getElementById("modal\_allocation").style.display = "none";  document.getElementById("modal\_masking").style.display = "none";  document.getElementById("modal\_allocation\_title").style.display = "none";  document.getElementById("modal\_masking\_title").style.display = "none";  if (study\_type == 'Observational') {  document.getElementById("modal\_intervention\_title").style.display = "none";  document.getElementById("modal\_intervention").style.display = "none";  document.getElementById("modal\_int\_obs\_model\_title").innerHTML = "Observation Model";  if (observation\_model != null) {  document.getElementById("modal\_int\_obs\_model\_description").innerHTML = observation\_model  };  } else {  document.getElementById("modal\_int\_obs\_model\_title").innerHTML = "Expanded Access";  document.getElementById("modal\_int\_obs\_model\_description").innerHTML = "Expanded Drug Access";  if (interventions != null) {document.getElementById("modal\_intervention").innerHTML = interventions};  }  }  if (phase != null) {document.getElementById("modal\_phase").innerHTML = phase};  if (outcome\_measure != null) {document.getElementById("modal\_primary\_outcome\_measure").innerHTML = outcome\_measure};  if (outcome\_description != null) {document.getElementById("modal\_primary\_outcome\_description").innerHTML = outcome\_description};   // managing central contact information  console.log(central\_contact);  if (central\_contact.length == 3) {  [phone, email] = sort\_email\_phone(central\_contact[1], central\_contact[2]);  document.getElementById("modal\_cc\_name").innerHTML = 'Contact Name: ' + central\_contact[0];  document.getElementById("modal\_cc\_phone").innerHTML = 'Contact Phone: ' + phone;  document.getElementById("modal\_cc\_email").innerHTML = 'Contact Email: ' + email;  } else {  document.getElementById("modal\_cc\_name").innerHTML = 'Contact Name: ' + central\_contact[0];  document.getElementById("modal\_cc\_phone").innerHTML = 'Contact Phone: ' + central\_contact[1];  }   // managing recruiting locations and associated contacts  for (var i = 0; i < fac\_and\_con.length; i++) {  console.log('working on :' + fac\_and\_con[i]);  new\_div = build\_location\_div(fac\_and\_con[i]);  $(new\_div).appendTo('#tabs-lo');  }    //fengyang: change all previous markers color except the input one  // add marker to locations nearby  change\_previous\_marker\_icon(marker\_list.slice(1));  let num\_add\_marker = 0;   let marker\_list\_current\_tab = [];  let info\_list\_current\_tab = [];  let marker\_index\_list\_current\_tab = [];  for (let i = 0; i < fac\_and\_con.length; i++) {  //fengyang add location map when click  let fac\_zip = fac\_and\_con[i][3].substring(0, 5);  let fac\_info = fac\_and\_con[i]  // check if zip is nearby/within the boundary  // need to add a var to import the nearby zipcode list: nearby\_zip\_code  if(nearby\_zip\_list.includes(fac\_zip)){  // console.log('marker by click');  num\_add\_marker = num\_add\_marker+1;  // create info for infowindow: official\_title, nct\_id  let info\_content = '<div>';  info\_content = info\_content + '<p>' + '<strong>' + 'Official Title' +'</strong>';  info\_content = info\_content + '<br />'+ official\_title + '</p>';  info\_content = info\_content + '<p>'+ '<strong>' + 'NCT ID Number'+'</strong>';  info\_content = info\_content + '<br />' + nct\_id + '</p>';  info\_content = info\_content + '<p>'+ '<strong>' + 'Location: '+ fac\_info[0] + '</strong>'+'<br />';  info\_content = info\_content + fac\_info[1] + ', ' + fac\_info[2] + ' ' + fac\_info[3] + '</p>';  info\_content = info\_content + '</div>';  marker\_list\_current\_tab.push(fac\_zip); //fengyang: do not use zip code to geocode anymore  info\_list\_current\_tab.push(info\_content);  marker\_index\_list\_current\_tab.push(i);  }  }  // fengyang new: location\_latlng\_list  if (info\_list\_current\_tab.length>0){  if(location\_latlng\_list.length>0){//if there are latlng records in pickle  for (let j = 0; j < info\_list\_current\_tab.length; j++){  let this\_marker\_latlng\_index = marker\_index\_list\_current\_tab[j];  let this\_marker\_latlng = location\_latlng\_list[this\_marker\_latlng\_index];  add\_marker\_with\_latlng(map, this\_marker\_latlng,  info\_content=info\_list\_current\_tab[j])  }  } else {  setTimeout(function(){  add\_10\_markers\_each\_time(marker\_list\_current\_tab, info\_list\_current\_tab, 10000);}, 0);  console.log('Find latlng by Geocode API.')  }  }    // add\_marker\_with\_latlng(resultsMap, latlng, info\_content='', icon\_url = '')   //setTimeout(function(){  //add\_10\_markers\_each\_time(marker\_list\_current\_tab, info\_list\_current\_tab, 10000);}, 0);   document.getElementById("modal\_ctgov\_link").href = "http://clinicaltrials.gov/ct2/show/" + nct\_id;   var modal = document.getElementById("myModal");  // modal.style.display = "block"; //fengyang: block style  }) } |

## Document

|  |
| --- |
| $(document).ready(function () {  semantiUIInit();   ...  $('#search\_button').bind('click',  function () {  $.blockUI({  message: '<div class="ui segment"><div class="ui active dimmer">Loading...<div class="ui text loader"></div></div></div>',  css: {  border: 'none',  '-webkit-border-radius': '40px',  '-moz-border-radius': '40px',  opacity: .5,  },  });   // input\_term = $('#first\_focus').val();  let input\_locn = $('#location\_terms').val();  let input\_range = $('#nearby\_miles').val();  let trial\_type = $('#trial\_type').val();  let active\_restriction = $('#active\_trial\_restriction').is(':checked');  let keyword\_search = $('#keyword\_search\_terms').val();  // fengyang  input\_miles\_list.push(parseInt(input\_range));   var vz = false;  $.getJSON($SCRIPT\_ROOT + '/\_check\_homepage\_parameters', {  locn: input\_locn,  miles: input\_range,  keyword: keyword\_search,  trial\_type: trial\_type,  active\_restriction: active\_restriction  },  function (data) {  vz = data.valid\_zip;  zct = data.nearby\_length;  kct = data.keyword\_length;  tct = data.type\_length;  wct = data.working\_nct\_length;  //fengyang: add zipcode  // nearby\_zip\_list = data.nearby\_zip\_list;  // $('#nearby\_zip\_list').val(data.nearby\_zip\_list);  // console.log("search\_button2");  nearby\_zip\_list = data.nearby\_zip\_list;  //console.log(nearby\_zip\_list);  //console.log(data.nearby\_zip\_list);  // nearby\_zip\_code = data.nearby\_zip\_code;  })   .done(function() {  console.log('is zip valid: ' + vz);  console.log('nearby length: ' + zct);  console.log('klength: ' + kct);  console.log('tlength: ' + tct);  console.log('wlength: ' + wct);  // nearby\_zip\_list = data.nearby\_zip\_list；   if (wct > 0) {  search("regular");  $('#filter\_results\_container').hide();  $('#question\_container').hide();  $('#hm\_blurb\_container').hide();  $('#logo\_div').hide();  $('#acknow\_div').hide();  $('#ctgov\_data\_div').hide();  $('#results\_container').show();  $('#multiquestions\_container').show();  $('#search\_form\_container').hide();  $('#dash\_board').hide();  $('#search\_results\_container').show();  //fengyang  let input\_zip = $('#location\_terms').val();  geocode\_marker\_with\_address(geocoder, map, input\_zip.toString(),  'Your input zip code: '+ input\_zip,  'http://maps.google.com/mapfiles/kml/pal4/icon57.png');   //add\_marker\_with\_address(geocoder, map, input\_zip.toString(),  // 'Your input zip code: '+ input\_zip,  // 'http://maps.google.com/mapfiles/kml/pal4/icon57.png');    $(document).scrollTop(0);  } else {...  });   });   $('#first\_focus')...   …  var modal = document.getElementById("myModal");  var span = document.getElementsByClassName("close")[0];   $(document).on("click", "a.trial\_listing\_header" , function() {  input\_nct = $(this).attr('name');  console.log('registered click: ' + input\_nct)  generate\_trial\_info\_modal(input\_nct);  return false;  });   $(document).on("mouseover", "a.trial\_listing\_header" , function() {  input\_nct = $(this).attr('name');  // console.log('registered click: ' + input\_nct)  selectMarker(input\_nct, 'start');  return false;  });  $(document).on("mouseout", "a.trial\_listing\_header" , function() {  input\_nct = $(this).attr('name');  // console.log('registered click: ' + input\_nct)  selectMarker(input\_nct, 'stop');  return false;  });  ////sout += 'onmouseover="selectMarker(' + 'nct[k][0]' + ',\'start\');" onmouseout="selectMarker(' + nct[k][0] + ',\'stop\');'; //">+ ';  // When the user clicks on <span> (x), close the modal  span.onclick = function() {  modal.style.display = "none";  // $('#tabs-lo').empty();  // $('.location\_block').remove();  reset\_info\_modal();  } ...  }); |
|  |

# app/templates/results.html

## Comment all codes for The Modal

|  |
| --- |
| <!-- The Modal --> <!-- fengyang myModal block comment-->  <!--<div id="myModal" class="modal">   &lt;!&ndash; Modal content &ndash;&gt;  &lt;!&ndash; <div class="modal-content"> fengyang&ndash;&gt;  <div class="modal-content2">  <span class="close">&times;</span>  <h4>  Official Title  </h4>  <p class="modal\_p" id="modal\_official\_title">  No information available in ClinicalTrials.gov  </p>  <h4>  NCT ID Number  </h4>  <p class="modal\_p" id="modal\_nct\_id">  No information available in ClinicalTrials.gov  </p>  <h4>  Study Type  </h4>  <p class="modal\_p" id="modal\_study\_type">  No information available in ClinicalTrials.gov  </p>  <h4>  Primary Purpose  </h4>  <p class="modal\_p" id="modal\_primary\_purpose">  No information available in ClinicalTrials.gov  </p>  <h4>  Study Description  </h4>  <p class="modal\_p" id="modal\_study\_description">  No information available in ClinicalTrials.gov  </p>   <div id="tabs">  <ul>  <li><a href="#tabs-el">Eligibility</a></li>  <li><a href="#tabs-sd">Study Design</a></li>  <li><a href="#tabs-lo">Contact Info and Locations</a></li>  <li><a href="#tabs-ai">Additional Resources</a></li>  </ul>  <div id="tabs-el">  <dl class-="dl-horizontal">  <dt>Patient Gender</dt>  <dd id="modal\_gender">  No information available in ClinicalTrials.gov  </dd>  <dt>Minimum Age</dt>  <dd id="modal\_min\_age">  There is no minimum age!  </dd>  <dt>Maximum Age</dt>  <dd id="modal\_max\_age">  There is no maximum age!  </dd>  <dt>Allows Participants Who Are COVID-negative</dt>  <dd id="modal\_healthy\_volunteers">  No information available in ClinicalTrials.gov  </dd>  </dl>  </div>  <div id="tabs-sd">  <dl class-="dl-horizontal">  <dt id="modal\_intervention\_title">Target Intervention</dt>  <dd id="modal\_intervention">  No information available in ClinicalTrials.gov  </dd>  <dt>Study Phase</dt>  <dd id="modal\_phase">  No information available in ClinicalTrials.gov  </dd>  <dt id="modal\_allocation\_title">Allocation</dt>  <dd id="modal\_allocation">  No information available in ClinicalTrials.gov  </dd>  <dt id="modal\_int\_obs\_model\_title">Intervention Model</dt>  <dd id="modal\_int\_obs\_model\_description">  No information available in ClinicalTrials.gov  </dd>  <dt id="modal\_masking\_title">Masking</dt>  <dd id="modal\_masking">  No information available in ClinicalTrials.gov  </dd>  <dt>Primary Outcome Measure</dt>  <dd id="modal\_primary\_outcome\_measure">  No information available in ClinicalTrials.gov  </dd>  <dt>Primary Outcome Description</dt>  <dd id="modal\_primary\_outcome\_description">  No information available in ClinicalTrials.gov  </dd>  </dl>  </div>  <div id="tabs-lo">  <strong>Central Contact Information</strong>  <ul style="padding-left:0;">  <li id="modal\_cc\_name">Contact Name: No information available in ClinicalTrials.gov</li>  <li id="modal\_cc\_phone">Contact Phone: No information available in ClinicalTrials.gov</li>  <li id="modal\_cc\_email">Contact Email: No information available in ClinicalTrials.gov</li>  </ul>  </div>  <div id="tabs-ai">  <a href="https://clinicaltrials.gov" id="modal\_ctgov\_link" class="modal\_ctgov\_link" target="\_blank">  Click HERE to see additional trial information on ClinicalTrials.gov</a>  </div>  </div>     </div>  </div>--> |

## Change layout for filter results container

|  |
| --- |
| <div id="filter\_results\_container" class='ui container' style='display: none'>  <h2 id="filter\_results\_summary" class='ui dividing header'>  <div class="ui ignored warning message fluid" id="result\_trial\_count\_display">You are still eligible for <code id='filter\_n' style="color:#007bff;"></code> trials  </div>  <div class="ui ignored negative negative fluid" style="display:none" id='qfilt\_warning\_3'>You are not eligible  for any trials. Try navigation to go back to previous found trials.  </div>  <div class="ui ignored negative negative fluid" style="display:none" id='qfilt\_warning\_4'>No more eligibility  questions. Please review the results.  </div>  </h2>  <div class="collapseContainer" style="border-radius:6px;">  <div class="ui primary ignored button center massive fluid toggle blue" id="show\_trials" show='no'><i  class="icon user"></i> Hide Eligible Trials</div>  <!--fengyang change style-->  <div class="ui grid two compact segments horizontal">  <div class="ui segment six wide column" id='filter\_results' style='display: none; background: white'>    <div class="ui relaxed divided list" style="overflow:auto;"></div>  <div class="ui right menu secondary stackable container">  <a id="fprev" class="icon item"><i class="left chevron icon"></i> Previous Page </a>  <a id="fshow" class="item"> Showing (<span id='ffirst'></span> - <span id='flast'></span>) </a>  <a id="fnext" class="icon item"> Next Page <i class="right chevron icon"></i></a>  </div>  </div>  <div class="ui segment ten wide column" style="overflow:auto;">  <div class="ui top attached label">Map</div> <!--fengyang: change title-->  <div id="floating-panel">  <input id="address" type="textbox" value="New York, NY">  <input id="submit" type="button" value="Search">  </div>  <!--fengyang add map: please change API key later-->  <script async defer  src="https://maps.googleapis.com/maps/api/js?key=AIzaSyAfGrfnJVQH1V0hU3sBKxgydYMH107Jx2M&callback=initMap&language=EN">  </script>  <div id="map"></div>  <div class="ui container" id="go\_back\_container" >  <div class="field">  <div class="ui container" style="font-size:14px;">  <!--<h4> Detailed trial info </h4>--> <!--fengyang-->  <div class="ui top attached label">Detailed trial info</div>  <div class="ui divided selection list" id="question\_tags">  </div>  </div>  </div>  </div>  <div class="modal-content2">  <span class="close">&times;</span>  <h4>  Official Title  </h4>  <p class="modal\_p" id="modal\_official\_title">  No information available in ClinicalTrials.gov  </p>  <h4>  NCT ID Number  </h4>  <p class="modal\_p" id="modal\_nct\_id">  No information available in ClinicalTrials.gov  </p>  <h4>  Study Type  </h4>  <p class="modal\_p" id="modal\_study\_type">  No information available in ClinicalTrials.gov  </p>  <h4>  Primary Purpose  </h4>  <p class="modal\_p" id="modal\_primary\_purpose">  No information available in ClinicalTrials.gov  </p>  <h4>  Study Description  </h4>  <p class="modal\_p" id="modal\_study\_description">  No information available in ClinicalTrials.gov  </p>  <div id="tabs">  <ul>  <li><a href="#tabs-el">Eligibility</a></li>  <li><a href="#tabs-sd">Study Design</a></li>  <li><a href="#tabs-lo">Contact Info and Locations</a></li>  <li><a href="#tabs-ai">Additional Resources</a></li>  </ul>  <div id="tabs-el">  <dl class-="dl-horizontal">  <dt>Patient Gender</dt>  <dd id="modal\_gender">  No information available in ClinicalTrials.gov  </dd>  <dt>Minimum Age</dt>  <dd id="modal\_min\_age">  There is no minimum age!  </dd>  <dt>Maximum Age</dt>  <dd id="modal\_max\_age">  There is no maximum age!  </dd>  <dt>Allows Participants Who Are COVID-negative</dt>  <dd id="modal\_healthy\_volunteers">  No information available in ClinicalTrials.gov  </dd>  </dl>  </div>  <div id="tabs-sd">  <dl class-="dl-horizontal">  <dt id="modal\_intervention\_title">Target Intervention</dt>  <dd id="modal\_intervention">  No information available in ClinicalTrials.gov  </dd>  <dt>Study Phase</dt>  <dd id="modal\_phase">  No information available in ClinicalTrials.gov  </dd>  <dt id="modal\_allocation\_title">Allocation</dt>  <dd id="modal\_allocation">  No information available in ClinicalTrials.gov  </dd>  <dt id="modal\_int\_obs\_model\_title">Intervention Model</dt>  <dd id="modal\_int\_obs\_model\_description">  No information available in ClinicalTrials.gov  </dd>  <dt id="modal\_masking\_title">Masking</dt>  <dd id="modal\_masking">  No information available in ClinicalTrials.gov  </dd>  <dt>Primary Outcome Measure</dt>  <dd id="modal\_primary\_outcome\_measure">  No information available in ClinicalTrials.gov  </dd>  <dt>Primary Outcome Description</dt>  <dd id="modal\_primary\_outcome\_description">  No information available in ClinicalTrials.gov  </dd>  </dl>  </div>  <div id="tabs-lo">  <strong>Central Contact Information</strong>  <ul style="padding-left:0;">  <li id="modal\_cc\_name">Contact Name: No information available in ClinicalTrials.gov</li>  <li id="modal\_cc\_phone">Contact Phone: No information available in ClinicalTrials.gov</li>  <li id="modal\_cc\_email">Contact Email: No information available in ClinicalTrials.gov</li>  </ul>  </div>  <div id="tabs-ai">  <a href="https://clinicaltrials.gov" id="modal\_ctgov\_link" class="modal\_ctgov\_link" target="\_blank">  Click HERE to see additional trial information on ClinicalTrials.gov</a>  </div>  </div>  </div>  <!-- fengyang -->  </div>  </div>  </div>  </div> |

# app/templates/header.html

Add the following styles

|  |
| --- |
| /\*fengyang start\*/  /\*Map\*/ #map {  /\* height: 20%; \*/  height: 350px; } #floating-panel {  /\*position: absolute;\*/  top: 10px;  left: 10%;  z-index: 5;  background-color: #fff;  padding: 5px;  border: 1px solid #999;  text-align: center;  font-family: 'Roboto','sans-serif';  line-height: 30px;  padding-left: 2px;  width: 300px }  /\* Modal Content 2 fengyang\*/ .modal-content2 {  background-color: #fefefe;  margin: auto;  padding: 20px;  border: 1px solid #888;  width: 80%;  max-height: calc(200vh - 210px);  overflow-y: auto; }  /\*fengyang end\*/ |

# app/lib/update\_trial\_latlng.py

Add a new py file to update latlng for locations for each trial.