Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ГЕОДЕЗИИ и КАРТОГРАФИИ (МИИГАИК)

Кафедра информационно-измерительных систем



**Курсовая работа**

по дисциплине: **«Геопорталы»**

на тему **«Разработка геопортала городской сети аптек»**

Выполнил:

студент ФПКиФ ИСИТ2-2м

Клинков В. А.

Проверил:

Ст.препод. Груздев С.С.

Москва, 2017

Содержание:

[ВВЕДЕНИЕ. 3](#_Toc501976112)

[Добавление карты с помощью библиотеки leaflet. 4](#_Toc501976113)

[Добавление слоёв карт. 5](#_Toc501976114)

[Добавление инструментов для измерения 7](#_Toc501976115)

[Добавление маркеров. 7](#_Toc501976116)

[Добавление легенды. 10](#_Toc501976117)

[Создание кластеризации. 11](#_Toc501976118)

[Добавление тепловой карты 12](#_Toc501976119)

[Заключение. 13](#_Toc501976120)

[Список литературы. 14](#_Toc501976121)

ВВЕДЕНИЕ.

В данной работе рассмотрена сеть аптек в городе Сергиев Посаде на 2017г, их расположение в зависимости от района, а так же указаны аптеки с круглосуточным графиком работы, а так же аптеки с благоприятными часами работы в вечернее время для людей которые днём работают, и в течении дня не успевают прикупить необходимые для здоровья лекарственные препараты и витамины.

**Целью** данной работы является создание геопортала с помощью библиотеки Leaflet на примере сети аптек в городе Сергиев Посад.

Для достижения поставленной цели, необходимо выполнить комплекс **задач**, таких как:

1. Добавить карту с помощью библиотеки Leaflet
2. Добавить слои других карт
3. Добавить инструменты для измерения
4. Добавить маркеры
5. Добавить точек при помощи файла GeoJSON
6. Создать кластеры
7. Создать тепловую карту

Язык гипертекстовой разметки (HyperTextMarkupLanguage — HTML), основной строительный блок веб-страниц, используется для создания и визуального представления веб-страниц. Он определяет содержание страницы, но не её функциональность.

JavaScript – это язык программирования, который добавляет интерактивность на ваш веб-сайт.

Leaflet — это JavaScript-библиотека для создания браузерных и адаптированных под мобильные устройства интерактивных карт.Для расширения функциональности библиотеки использовали плагины:

* Leaflet-measure – контроль координат, измерение расстояний и площадей.
* Leaflet.MarkerCluster – кластеризация объектов.
* Leaflet.Heat – создание тепловых карт.

Добавление карты с помощью библиотеки leaflet.

Самым первым делом мы создаём HTML документ в котором начинаем прописывать код нашего геопортала начиная с самого начала.

Для начала необходимо скачать библиотеку Leaflet [http://leafletjs.com/download.html]. Далее подключить ее: добавить файл стиля leafletи библиотеку Javascript. Задаём имя вкладки и устанавливаем кодировку для страницы по умолчанию UTF-8.

<title> Сеть городских аптек Сергиев Посада </title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<script src="./bin/leaflet.js"></script>

<link rel="stylesheet" href="./bin/leaflet.css">

Затем мы задаём размер карты и добавляем название геопортала.

<style>

#map{height:500px; width:800px}

h1{text-align: center}

</style>

<body>

<h1>Сеть городских аптек Сергиев Посада </h1>

</body>

Далее нам необходимо настроить карту под себя, для этого необходимо установить центр карты, в моём случае это будет центр Сергиев Посаде, установить min и max приближение и д.р.

var map = L.map('map', {

center: [56.315321, 38.135766],

zoom: 14,

minZoom: 7,

maxZoom: 19,

zoomControl: false,

attributionControl: false,

maxBounds:[[56.932987,34.958496],[54.175297,40.231934]],

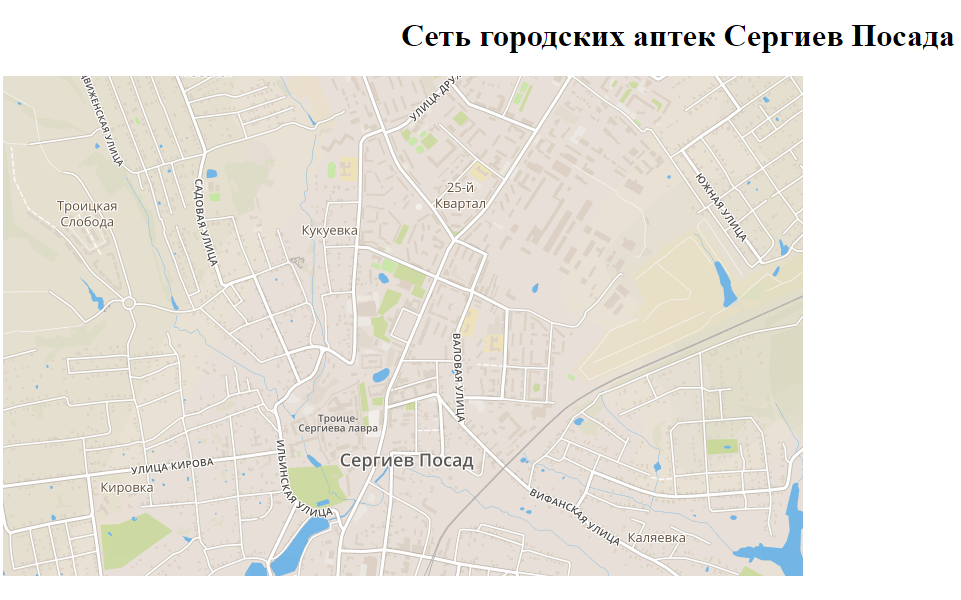
layers: [mbOSM] });

Осталось только добавить саму карту, для примеры мы будем добавлять слой карты Mambox.

var mURL = 'https://api.tiles.mapbox.com/v4/{id}/{z}/{x}/{y}.png?access \_token=pk.eyJ1IjoibWFwYm94IiwiYSI6ImNpejY4NXVycTA2emYycXBndHRqcmZ3N3gifQ.rJcFIG214AriISLbB6B5aw',

var mbOSM = L.tileLayer(mURL,{attribution: mAttr, id:'mapbox.streets'});

В итоге в браузере мы уже можем видеть название нашего геопортала и карту.



Добавление слоёв карт.

Добавим слои карт Google.

var GM = [L.tileLayer('http://{s}.google.com/vt/lyrs=s&x={x}&y={y}&z={z}', {maxZoom: 20, subdomains:['mt0','mt1','mt2','mt3']}), // КартаGoogle.Satellite

L.tileLayer('http://{s}.google.com/vt/lyrs=m&x={x}&y={y}&z={z}', {maxZoom: 20, subdomains:['mt0','mt1','mt2','mt3']}), // КартаGoogle.Hybrid

L.tileLayer('http://{s}.google.com/vt/lyrs=s,h&x={x}&y={y}&z={z}', {maxZoom: 20, subdomains:['mt0','mt1','mt2','mt3']})] // КартаGoogle.Street

var baseLayers = {

"Google Спутник": GM[0],

"Google Карта": GM[1],

"Google Гибрид": GM[2] }

Добавляем слои карт Яндекс.

Для начала добавляем ссылку на карту.

<script src="./bin/Yandex.js"></script>

<script src="http://api-maps.yandex.ru/2.0/?load=package.map&lang=Ru"> </script>

Затем добавляем слой карты Яндекс и слой с Яндекс Пробками.

var ya = new L.Yandex();

var yaTr = new L.Yandex("null",{traffic:true, opacity:1, overlay: true});

var baseLayers = {

"Яндекс Карта": ya,

"Яндекс Пробки": yaTr };

Добавим кадастровую карту.

var WMSCad = L.tileLayer.wms('https://pkk5.rosreestr.ru/arcgis/services/ Cadastre/CadastreWMS/MapServer/WMSServer?',{

layers: '27,28,29,30,31,32',

format: 'image/png',

transparent: true })

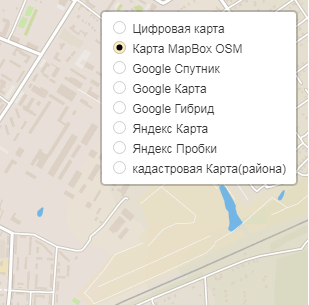
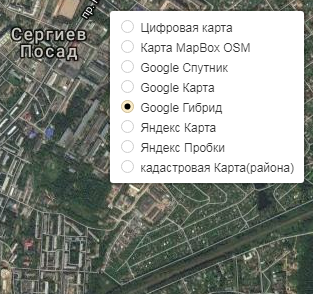
var overlays = {"кадастровая Карта(района)": WMSCad}

В конце прописываем элементы интерфейса для выбора слоёв и элементы интерфейса масштабной линейки.

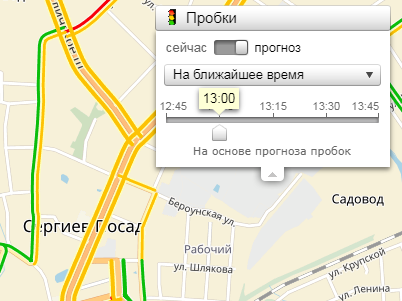
L.control.layers(baseLayers, overlays).addTo(map);

L.control.scale({imperial: false, maxwidth: 200, position: 'bottomright'}) .addTo(map);

По итогу в браузере мы уже можем видеть слои с подключенными нами картами.

Доступные слои. Слой карты Google Гибрид.



Слой карты Яндекс Пробки.

Добавление инструментов для измерения

Для того чтобы измерять расстояние и площадь на карте необходимо скачать плагин **Leaflet-measure**. Этот плагин позволяет контролировать координаты, а также измерять расстояние и площадь.

Первым делом скачиваем и подключаем этот плагин.

<scriptsrc="./bin/leaflet-measure.min.js"></script>

Затем производим его настройку.

var msrControl = new L.Control.Measure({

localization: 'ru',

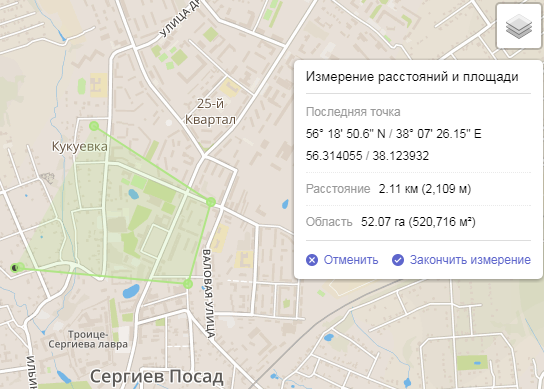
secondaryAreaUnit: 'sqmeters',

primaryLengthUnit: 'kilometers', secondaryLengthUnit: 'meters',

primaryAreaUnit: 'hectares', secondaryAreaUnit: 'sqmeters' });

msrControl.addTo(map);

Данный инструмент на карте будет выглядеть при измерении следующим образом.



Добавление маркеров.

Для добавления маркеров необходимо создать новую папку icons в папке Data. В новую папку загрузить картинки иконок. Я использовал иконки с сайта [[https://www.flaticon.com/](https://vk.com/away.php?to=https%3A%2F%2Fwww.flaticon.com%2F&cc_key=)], а затем их дорабатывал в [Adobe Photoshop](https://ru.wikipedia.org/wiki/Adobe_Photoshop).

Задаём класс эконки.

var Aicn = L.Icon.extend({

options:{iconSize:[32,32],

iconAnchor: [16,32],

popupAnchor:[0,-20]

}})

Потом с помощью массива мы задаём путь к интересующим нас иконкам, в данном случае иконки времени до которого работает аптека.

var AicnT = [new Aicn({iconUrl: './date/icons/19.png'}),

new Aicn({iconUrl: './date/icons/20.png'}),

new Aicn({iconUrl: './date/icons/21.png'}),

new Aicn({iconUrl: './date/icons/22.png'}),

new Aicn({iconUrl: './date/icons/24.png'}),

new Aicn({iconUrl: './date/icons/time.png'})

];

Чтобы все наши метки нам надо подгрузить файл с точками сделанный при помощи QGIS и сохранённый в GeoJSON. Данный файл должен быть заполнен интересующей нас информацией об аптеках.

Для дальнейшего его подключения все данные файла надо заключить в переменную, в моём случае эта переменная будет называться «apt», а сам файл надо будет сохранить с расширением js.

Чтобы начать работу с ним нам надо его подключить к нашему геопорталу.

<script src="./date/Apteki.js"> </script>

Затем подгружаем метки и пишем небольшое условие для того чтобы при нажатии на метку показывалась фотография аптеки и её название.

var apteki2 = L.geoJSON(apt, {pointToLayer: function(feature,latlng){

var vrema = feature.properties.Time;

vrema == '09:00 – 19:00' || vrema == '08:00 – 19:00' ? icn = AicnT[0]:

vrema == '09:00 – 20:00' || vrema == '08:00 – 20:00' ? icn = AicnT[1]:

vrema == '09:00 – 21:00' || vrema == '08:00 – 21:00' || vrema == '10:00 – 21:00' ? icn = AicnT[2]:

vrema == '09:00 – 22:00' || vrema == '08:00 – 22:00' || vrema == '10:00 – 22:00' ? icn = AicnT[3]:

vrema == 'Круглосуточно' ? icn = AicnT[4]:

icn = AicnT[5];

return L.marker(latlng,{icon: icn})

}})

.bindPopup(function(apt) {

var WApteki = '';

if (apt.feature.properties.Name != '(0:)') {

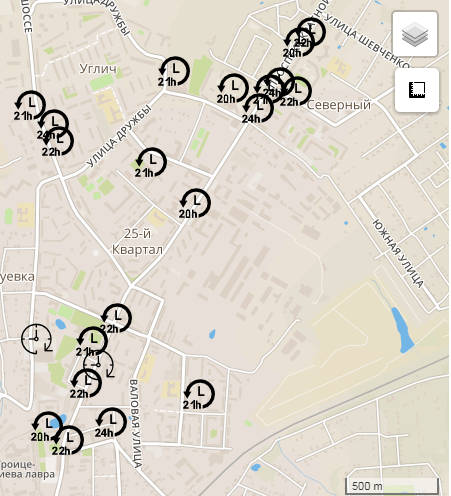
WApteki = '<br><img src ="./date/aptek/' + apt.feature.properties.id + '/' + apt.feature.properties.id + '.jpg" height ="150" width = "267">';}

return '<b> Имя: </b> ' + apt.feature.properties.Name +WApteki });

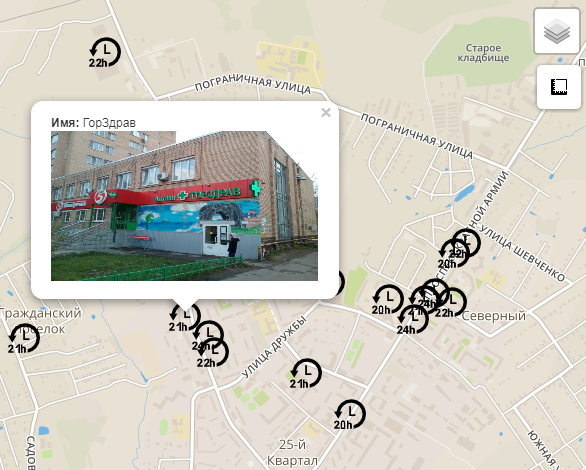
Добавляем наши метки к слою.

var overlays = {"Аптеки в Сергиев Посаде" : apteki2}

Теперь все наши аптеки видны на нашей карте.



А при нажатии можно увидеть название аптеки и её фотографию.



Добавление легенды.

Легенда это вспомогательная информация для пользователя, которая в свою очередь помогает разобраться в описании меток аптек.

Для добавления легенды создам новую переменную и настраиваем её.

var lgnd = L.control({position: 'bottomleft'})

lgnd.onAdd = function (map) {

var div = L.DomUtil.create('div','infolgnd'),

labels = [];

labels.push('<b> Часы работы аптек </b>');

labels.push('<img src ="./date/icons/19.png" height ="15" width = "15">'+'До 19.00');

labels.push('<img src ="./date/icons/20.png" height ="15" width = "15">'+'До 20.00');

labels.push('<img src ="./date/icons/21.png" height ="15" width = "15">'+'До 21.00');

labels.push('<img src ="./date/icons/22.png" height ="15" width = "15">'+'До 22.00');

labels.push('<img src ="./date/icons/24.png" height ="15" width = "15">'+'Круглосуточно');

labels.push('<img src ="./date/icons/time.png" height ="15" width = "15">'+'Другое время');

div.innerHTML = labels.join('<br>');

return div}

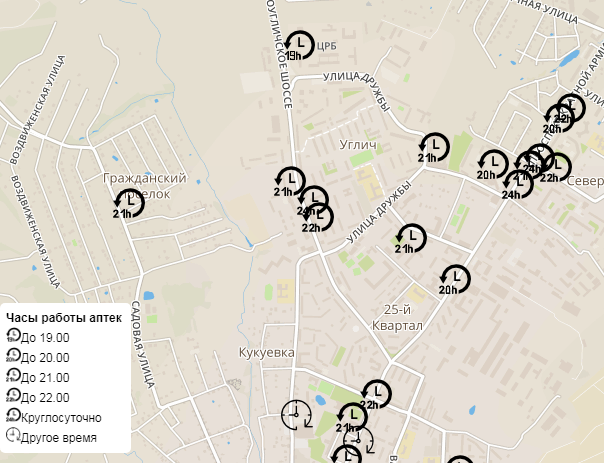
lgnd.addTo(map);

Задаём стиль наше легенды цвета, размер и прочее.

<style>

.infolgnd{background: white; padding: 6px; border-radius: 5px; background:rgba(255,255,255,0,8) }

</style>



Создание кластеризации.

Для создания кластеров необходимо скачать плагин Leaflet.MarkerCluster.

Далее добавляем библиотеку и стили плагина в элемент body:

<scriptsrc="./bin/leaflet.markercluster.js"></script>

<link rel="stylesheet" href="./bin/MarkerCluster.css">

<link rel="stylesheet" href="./bin/MarkerCluster.Default.css">

В элементе script прописываем параметры кластера и добавляем его к карте.

var cc = L.markerClusterGroup({

showCoverageOnHover: false,

//zoomToBoundsOnClick: false,

//spiderflyOnMaxZoom: false,

removeOutsideVisibleBounds: true,

spiderLegPolylineOptions: {weight: 2, color: '#775249', opacity: 1},

//disableCluster...

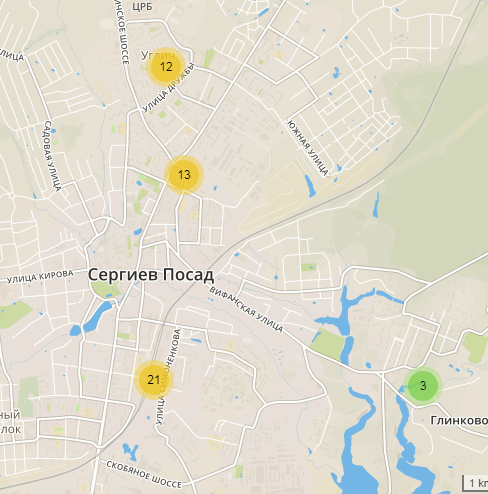
maxClusterRadius: 200 });

cc.addLayer(apteki2);

map.addLayer(cc);

var AptekiK = cc;

В итоге у нас получается кластер всех наших аптек, где можно посмотреть в каком районе находится больше аптек, а в каких значительно меньше.



Добавление тепловой карты

Для работы необходимо скачать плагин для создания тепловых карт leaflet-heat и добавить его в документ.

<scriptsrc="./bin/leaflet-heat.js"></script>

Далее создаем новую переменную и добавляем в массив координаты объектов. Устанавливаем параметры отображения.

var Hmap = L.heatLayer([

[56.339821523624906, 38.124630633855233, 0.5],

……………………………………………………………

[56.316094443561695, 38.132070146981,1]

], {

radius: 70,

blur: 50,

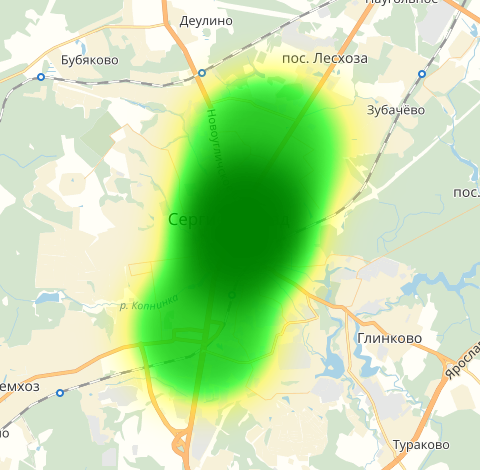
maxZoom: 15,

// minOpacity: 0.2,

gradient: {0.4: 'yellow', 0.65:'lime',1:'green' }

}); // карта плотности аптек

Итог который у нас получился.



Заключение.

У нас получилось создать инфраструктивный геопортал который будет полезен всем и всегда. На которой можно посмотреть все аптеки города, узнать до скольких они работают, посмотреть в какой части города их больше и многое другое.

По завершению создания нашего геопортала, у нас получилось достичь поставленных задач, а именно:

* Добавить карту с помощью библиотеки Leaflet
* Добавить слои других карт
* Добавить инструменты для измерения
* Добавить маркеры
* Добавить точек при помощи файла GeoJSON
* Создать кластеры
* Создать тепловую карту.

Список литературы.

1. Библиотека для интерактивных карт leaflet [http://leafletjs.com/]

2. Учебник по HTML. [htmlbook.ru]

3. Учебники по javascript [https://learn.javascript.ru/], [https://javascript.ru/]

4. Сайт с иконками. [[https://www.flaticon.com/](https://vk.com/away.php?to=https%3A%2F%2Fwww.flaticon.com%2F&cc_key=)]