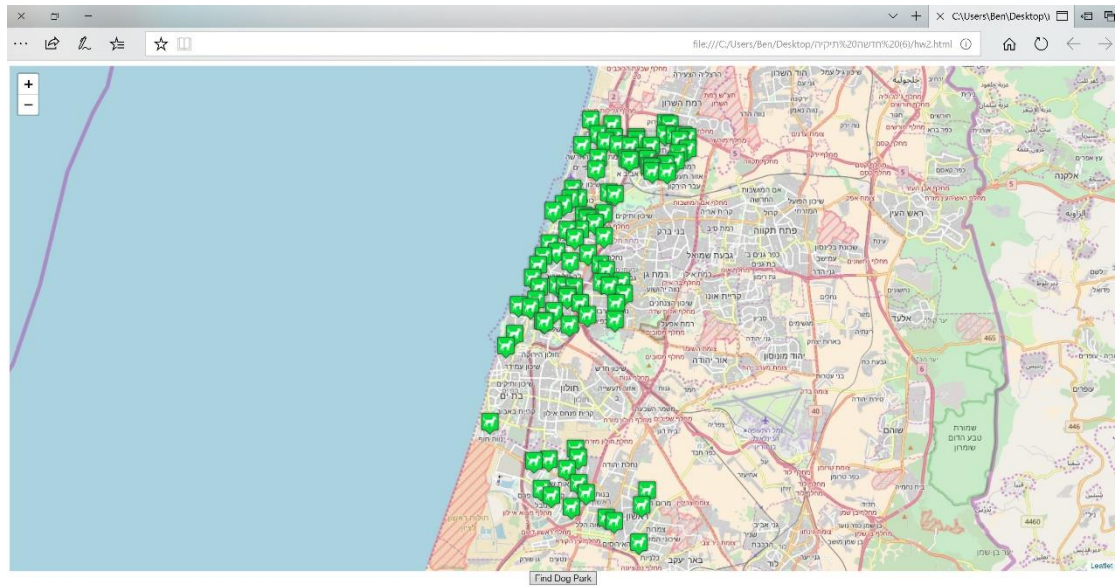
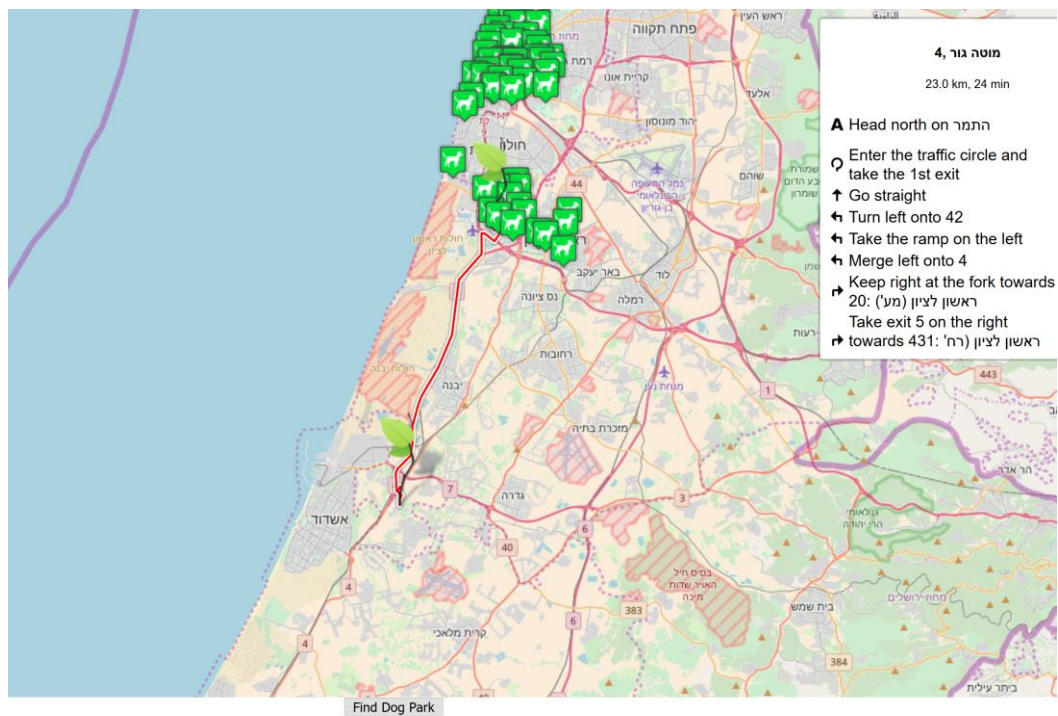
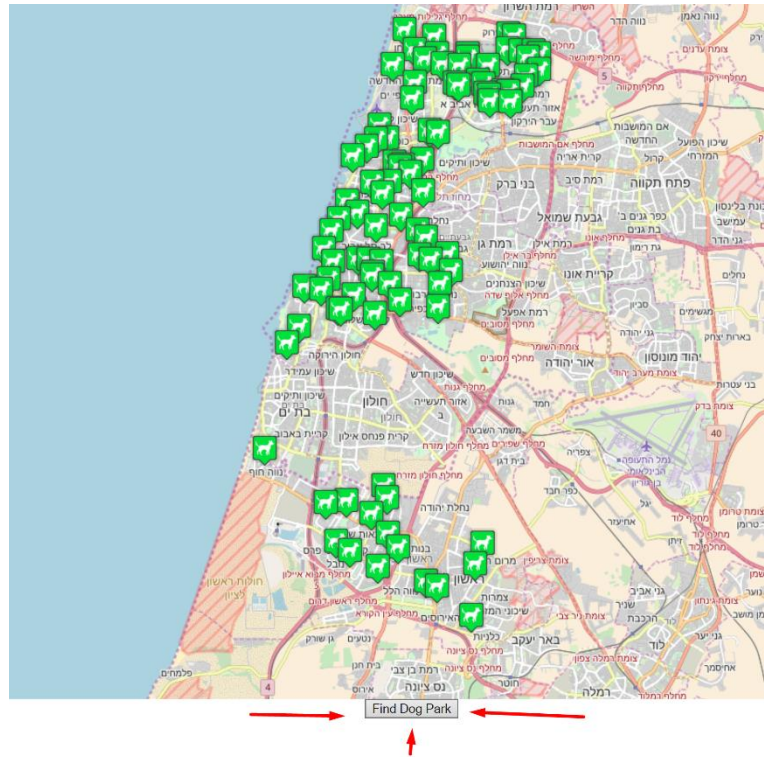


- בן דן - 200879237
- יותם זייבל - 201307832

תרגיל 2 - Dog Park Finder



דוגמאת שימוש:



הקדמה

במסמך הבא יפורט תהליך בניית תוכנית המציגה למשתמש גינות כלבים ברחבי ת"א וראשל"צ ומאפשר בניית מסלול לגינת הכלבים הקרובה ביותר למיקום המשתמש.

במהלך המסמך אנו נשתמש בכלים הבאים:

1. HTML
2. JS
3. QGIS
4. Leaflet (API)

***מומלץ להפעלה בדפדפן Firefox

שלב 1: הצגת מפה בדפדפן

א. תחילה נוסיף את התיקיות הבאות ב header על מנת שנוכל להשתמש ב-API המפות של Leaflet:

```
<link rel="stylesheet" href="https://unpkg.com/leaflet@1.3.0/dist/leaflet.css"
  integrity="sha512-
Rksm5RenBEKSKFjgI3a41vrjkw4EVPIJ3+OiI65vTjIdo9brlAacEuKOiQ5OFh7cOI1bkDw
LqdLw3Zg0cRJAAQ=="
  crossorigin=""/>

<!-- Make sure you put this AFTER Leaflet's CSS -->
<script src="https://unpkg.com/leaflet@1.3.0/dist/leaflet.js"
  integrity="sha512-
C7BBF9irt5R7hqbUm2uxtODIUVs+IsNu2UULGuZN7gM+k/mmeG4xvIEac01BtQa4YIkU
pp23zZC4wIwuXaPMQA=="
  crossorigin=""></script>
```

ב. כעת נוסיף את ה-div הבא ל-body:

```
<div id="mapid" style="height: 90%; width: 100%;"> </div>
```

ג. נוסיף את הסקריפט הבא, שבעצם יוצר מפה חדשה ובו ניתן להגדיר זום ראשוני, ונקודה ראשונית בה המפה תפתח (בדוגמא הגדרנו מבט ראשוני על תל אביב בזום של 12):

```
<script>
// Center the map to TelAviv, Israel
// Bind the map to the #map HTML element

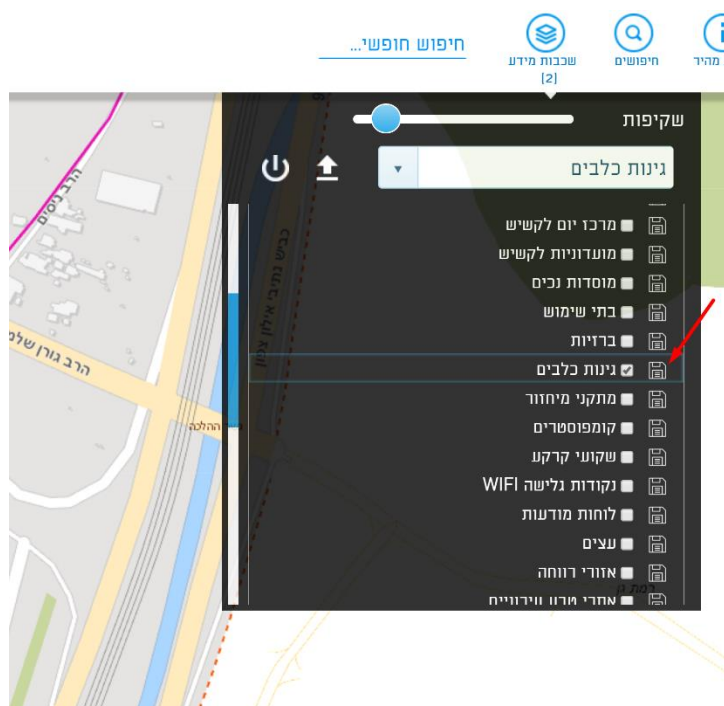
var map = L.map("mapid").setView([32.095, 34.783], 12);

// Get the map layer
L.tileLayer(
'http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png',
{
  maxZoom: 17,
  minZoom: 1
}
).addTo(map);

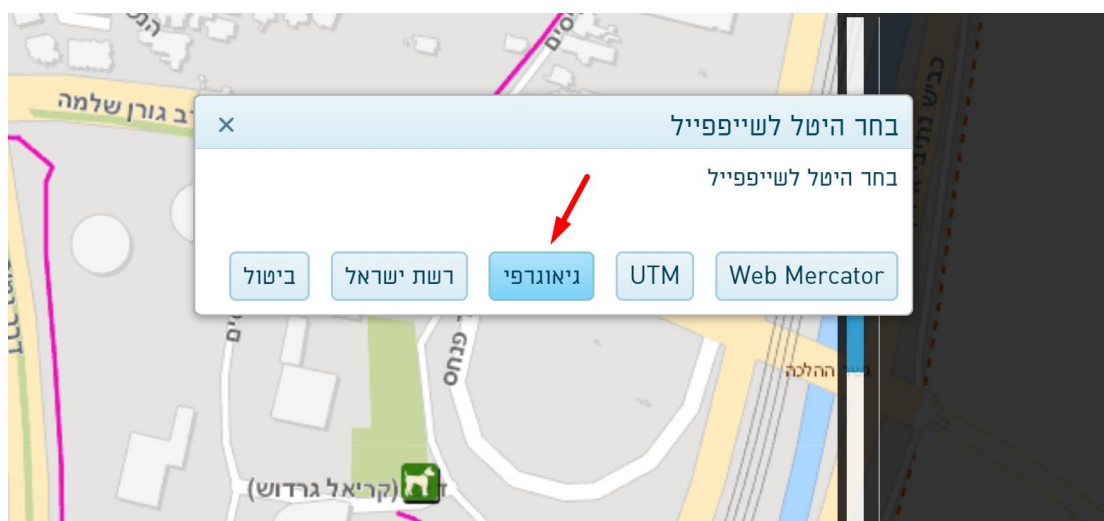
</script>
```

שלב 2: הוספת שכבת נתונים GEOjson למפה שאותה יצרנו (גינות כלבים)

את שכבת הנתונים שלנו נוריד [מהאתר של עיריית ת"א](#): שירותים > מערכת המפות GIS

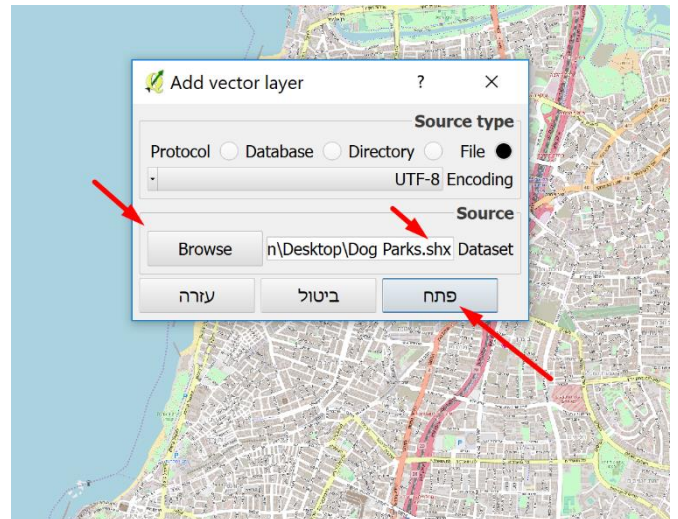
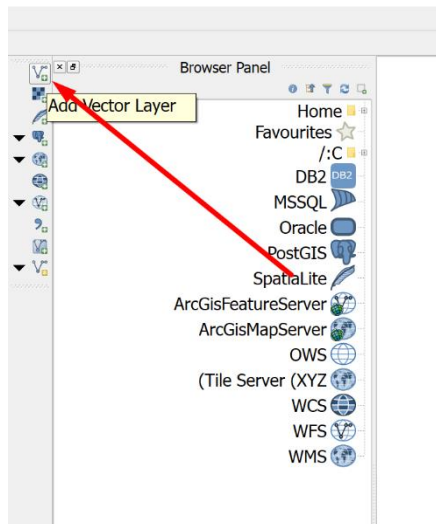


הורדת שכבת הנתונים



כעת נפתח את הקובץ עם סיומת shx בתוכנת QGIS על מנת שנוכל לייצא את הקובץ כ-geojson

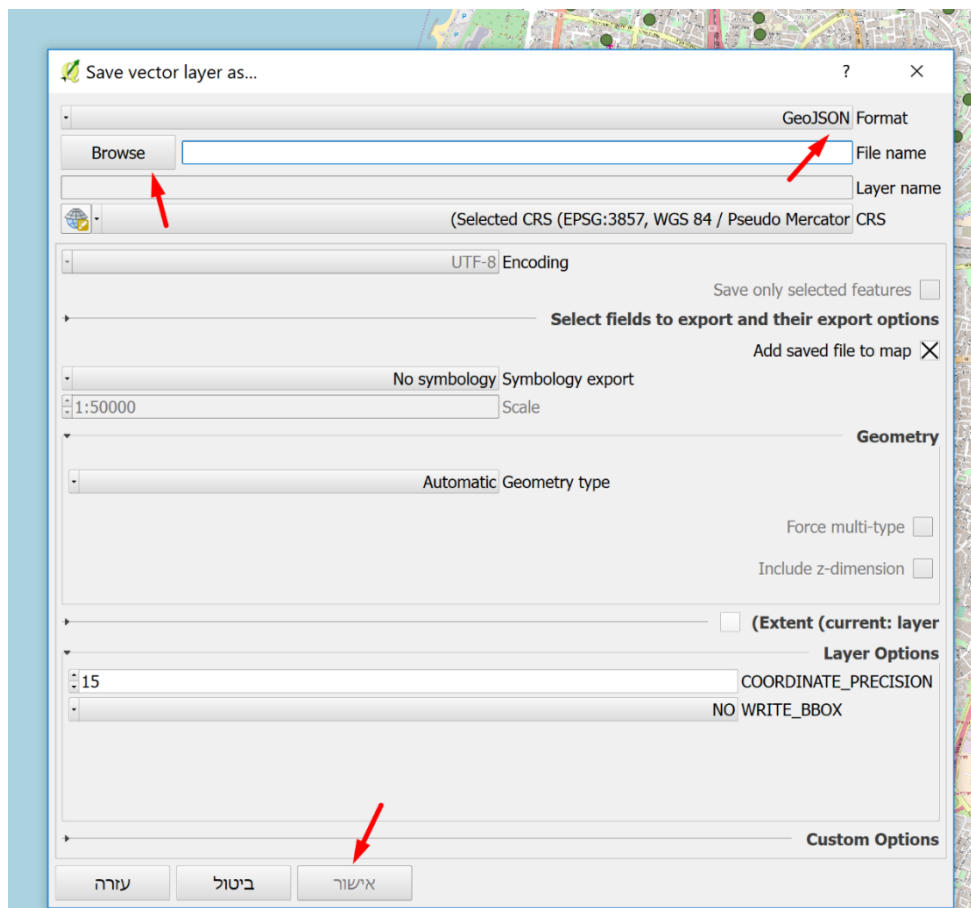
א. פתיחת הקובץ (כמו בתמונות)



כעת נראה כי נוספה לנו שכבה חדשה בפאנל השכבות עם גינות הכלבים, מה שנותר זה לייצא אותה כקובץ JSON על מנת נוכל לטעון למפה שייצרנו.

ב. יצוא קובץ JSON:

נלחץ כפתור ימני על השכבה החדשה ובחר SAVE AS...



שלב 3: טעינת הנתונים על המפה של Leaflet

א. כעת שיש לנו קובץ GeoJson, נוסיף אותו לתיקייה בה שמרנו את קובץ ה-HTML ונוסיף את הקוד הבא:

```
var geojsonLayer = new L.GeoJSON.AJAX("Dog-ParksW84.geojson")
geojsonLayer.addTo(map);
```

הערה: Dog-ParksW84.geojson = שם הקובץ

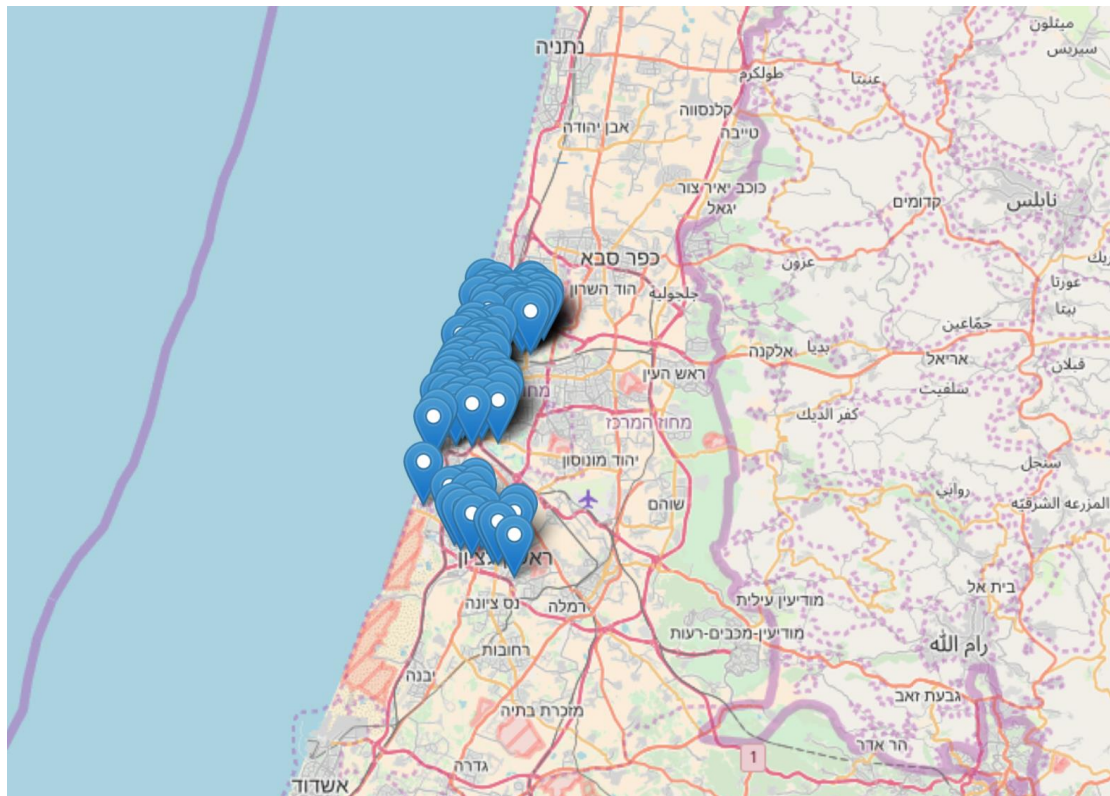
ב. ונוסיף את התיקיות הבאות בheader על מנת שנוכל להשתמש בפונקציית ה-AJAX:

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/leaflet.ajax/2.1.0/leaflet.ajax.min.js"></script>
```

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/geojson/0.5.0/geojson.js">
```

```
</script>
```

כעת קיבלנו מפה עם נקודות המציינות גינות כלבים עליה, כמו בתמונה:



שלב 4: נוסף את אפשרות הניווט לגינה הקרובה

א. הוספת הכפתור שישמש אותנו לביצוע הפעולה:

```
<button onclick="findPark()">Find Dog Park</button>
```

ב. כעת נוסיף את הקוד הבא, אשר יודע לחשב את המרחק מהמיקום הנוכחי של המשתמש לגינת הכלבים הקרובה ביותר עבורו ולהציג מסלול ניווט מתאים:

```
function findPark() {
    if (navigator.geolocation) {
        navigator.geolocation.getCurrentPosition(minDis);
    } else {
        console.log("Geolocation is not supported by this browser.");
    }
}

function distance(lat1, lon1, lat2, lon2, unit) {
    var radlat1 = Math.PI * lat1/180
    var radlat2 = Math.PI * lat2/180
    var theta = lon1-lon2
    var radtheta = Math.PI * theta/180
    var dist = Math.sin(radlat1) * Math.sin(radlat2) + Math.cos(radlat1) * Math.cos(radlat2) * Math.cos(radtheta);
    dist = Math.acos(dist)
    dist = dist * 180/Math.PI
    dist = dist * 60 * 1.1515
    if (unit=="K") { dist = dist * 1.609344 }
    if (unit=="N") { dist = dist * 0.8684 }
    return dist
}

function minDis(MyPos) {
    var abc = L.marker([MyPos.coords.latitude,MyPos.coords.longitude]).addTo(map);

    $.getJSON("Dog-ParksW84.geojson", function(json) {
        var x1=Number(json.features[0].geometry.coordinates[0]);
        var y1=Number(json.features[0].geometry.coordinates[1]);

        //my position
        var x2=MyPos.coords.longitude;
        var y2=MyPos.coords.latitude;

        var min=distance(x1,y1,x2,y2,'K');
        var index=0;
        var minParkIndex=0;

        (json.features).forEach(function (park){

            try {

                var x1=Number(park.geometry["coordinates"][0]);
                var y1=Number(park.geometry["coordinates"][1]);

                var tmp = distance(x1,y1,x2,y2,'K');

            }
            catch(err){}

            if(tmp<min){
                min=tmp;
                minParkIndex=index;
            }
            index++;
        })

        x1=Number(json.features[minParkIndex].geometry["coordinates"][0]);
        y1=Number(json.features[minParkIndex].geometry["coordinates"][1]);
        L.Routing.control({
            waypoints: [
                L.latLng(y2, x2),
                L.latLng(y1, x1)
            ]
        }).addTo(map);

    });
}
```

שלב אחרון: בשלב הזה לחיצה על הכפתור שיצרנו מבצעת קריאה לפונקציה המתאימה שיודעת לחשב המרחקים ולהחזיר את המיקום של הגינה הקרובה ביותר. לאחר מכן נלקח המיקום של המשתמש עם גינת הכלבים הנבחרת ובעזרת API של Leaflet נבנה מסלול ניווט מהמיקום של המשתמש לגינת הכלבים.