

# Music Memory Lane - MWG8 - 1260588

## Contents:

[Intro.](#)

[Data extraction and preparation](#)

[Application development.](#)

[Final result and thoughts](#)

## Music Memory Lane

An exploration of Last.fm data and Google Maps location history.

**Hold shift and click and drag** to explore points on that region. In this demo there is no data outside of New Zealand.

### Activities detected:

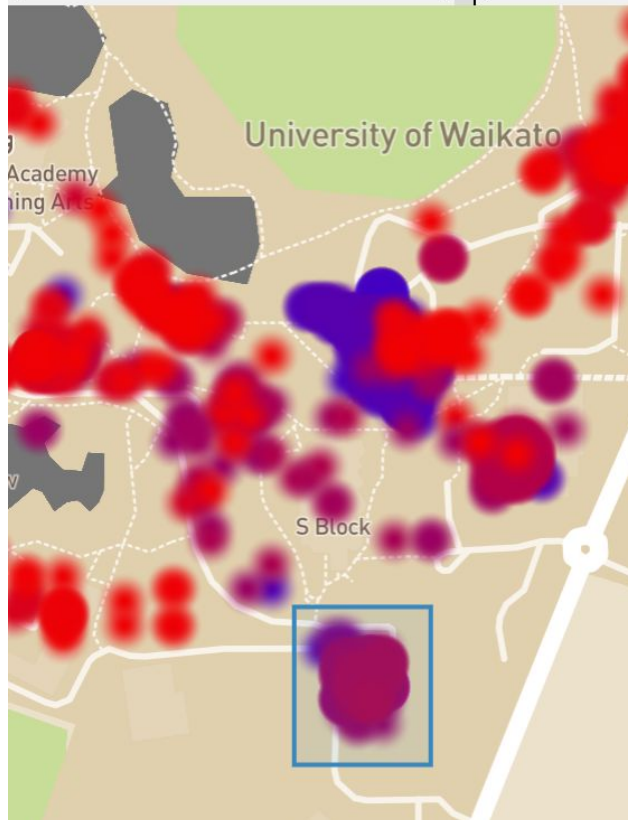
STILL: 1032  
TILTING: 98  
UNKNOWN: 24  
IN\_VEHICLE: 16  
ON\_FOOT: 7  
ON\_BICYCLE: 1

### Top days listened:

8-Jun-17: 146  
11-May-17: 87  
30-Mar-17: 76  
5-Jun-17: 73  
10-May-17: 61  
23-Feb-17: 61  
6-May-17: 54  
15-Jun-17: 51  
28-Mar-17: 41  
6-Jun-17: 38  
25-May-17: 37

### Top Artists listened:

Zammuto: 66  
death's dynamic shroud.wmv: 61  
Chance the Rapper: 50  
The Knife: 49  
Gorillaz: 48  
James Blake: 48  
Moderat: 48  
Death Grips: 43  
Girls Who Care: 36  
Fennec: 33  
My Bloody Valentine: 33



### 1178 tracks:

31-Jul-15: Björk - Cocoon  
31-Jul-15: Björk - It's Not Up to You  
5-Aug-15: The Avalanches - Since I Left You  
5-Aug-15: The Avalanches - Stay Another Season  
23-Mar-16: The Sugarhill Gang - Fireworks  
23-Mar-16: The Sugarhill Gang - Last Day of School  
23-Mar-16: The Sugarhill Gang - Sugar Hill Groove  
23-Mar-16: The Sugarhill Gang - My Little Playmate  
23-Mar-16: Lorn - Me Or You  
23-Mar-16: Lorn - Megaman Rmx  
24-Mar-16: Lorn - Metropolis  
24-Mar-16: Lorn - More Human Than Human (Lorn Remix)  
24-Mar-16: Lorn - Murphy Bed  
24-Mar-16: Lorn - my flower  
24-Mar-16: Lorn - My Single Dancefloor  
17-Sep-16: Lekebergstrion - B1 Go Tell It on the Mountain  
17-Sep-16: Fennec - OXOXO  
17-Sep-16: Fennec - Rearview Glare  
2-Oct-16: How to Dress Well - The Ruins  
2-Oct-16: How to Dress Well - Burning Up

# Intro.

I am an avid listener of music, over the last 7 years I've listened to 96,000+ songs with an average of 36 per day. I know this because every song I listen to is tracked in [last.fm](https://last.fm). I have a profile on it which I can see my top artists, albums, tracks and more from any year or any date range. You can look for yourself if you want, at [last.fm/user/CaptainNikGreen](https://last.fm/user/CaptainNikGreen).

My second dataset comes from Google Maps, Android devices often check into Google and store your current location and the activity you were doing. There are 8 possible activities, Google tries to figure out which one you are currently doing from the devices velocity and orientation.

## Constant Summary

int	<a href="#">IN_VEHICLE</a>	The device is in a vehicle, such as a car.
int	<a href="#">ON_BICYCLE</a>	The device is on a bicycle.
int	<a href="#">ON_FOOT</a>	The device is on a user who is walking or running.
int	<a href="#">RUNNING</a>	The device is on a user who is running.
int	<a href="#">STILL</a>	The device is still (not moving).
int	<a href="#">TILTING</a>	The device angle relative to gravity changed significantly.
int	<a href="#">UNKNOWN</a>	Unable to detect the current activity.
int	<a href="#">WALKING</a>	The device is on a user who is walking.

More details at [Google's API documentation](#)

When I saw the project assignment details I knew it would be great to do something with these two datasets available to me. This project is done solo by me.

# Data extraction and preparation

## Data sources:

I got all my listens from 26 May 2020 to 2013 from <https://lastfm.ghan.nl/export/> (13mb CSV)

I got my location data from 26 May back to 2014 [Google Takeout](#) (150mb JSON)

Once I had my two files from Last fm and Google Maps, immediately I thought of a problem with my data. There won't be a 1:1 relationship with each location data point and each track listen.

My movement is something being tracked 24/7 where I'm not listening to music 24/7 either. Also since the timestamps are measured in seconds, they won't exactly line up with the same timestamp as my music. I need a bit of slack to accept the location for where a scrobble was made. I decided to accept the last known location in the last 6 hours per song listen.

The Google Maps data came in a 150mb JSON file, that had way more details than I needed. Each data point has a confidence value for every Activity and also multiple latitude and longitude points for each single data point, to show a path travelled in that point. I knew I had to slim this down to work with only what I needed.

Here's a single JSON entry on the right, which I then cleaned up in Excel using Power Query Editor. I basically stripped out all the other activity information that didn't have the highest confidence interval. Also I wasn't concerned with the accuracy field either, since I wasn't sure I could include it in the application. In hindsight, I wonder if I could've used the accuracy field to change the size of a data point on the application map to represent the possible area it could've been in.

After a lot of cleaning up in Excel the final CSV was **11mb, down from 150mb**. Which is much easier for Python parsing.

The Google Maps timestamp value includes milliseconds. I truncated it to match with Last FM's timestamp value.

```
{
  "locations" : [ {
    "timestampMs" : "1400836128607",
    "latitudeE7" : -377427719,
    "longitudeE7" : 1761333490,
    "accuracy" : 20,
    "activity" : [ {
      "timestampMs" : "1400835903281",
      "activity" : [ {
        "type" : "STILL",
        "confidence" : 100
      } ]
    } ],
    "timestampMs" : "1400836086963",
    "activity" : [ {
      "type" : "UNKNOWN",
      "confidence" : 44
    }, {
      "type" : "STILL",
      "confidence" : 33
    }, {
      "type" : "IN_VEHICLE",
      "confidence" : 18
    }, {
      "type" : "ON_BICYCLE",
      "confidence" : 5
    } ],
    "timestampMs" : "1400836147558",
    "activity" : [ {
      "type" : "STILL",
      "confidence" : 85
    }, {
      "type" : "IN_VEHICLE",
      "confidence" : 10
    }, {
      "type" : "UNKNOWN",
      "confidence" : 5
    } ]
  } ]
}
```

	ABC	ABC	ABC	ABC	
	123	123	123	123	
	Name	Value.timestampMs	Value.latitudeE7	Value.longitudeE7	Value.accuracy
1	locations	1400836128607	-377427719	1761333490	20
2	locations	1400836190078	-377427623	1761333464	20
3	locations	1400836251046	-377427589	1761333600	20
4	locations	1400836343055	-377427786	1761333440	20
5	locations	1400836404105	-377428111	1761333498	32
6	locations	1400836475592	-377427952	1761333372	33
7	locations	1400836535550	-377427755	1761333239	20
8	locations	1400836596536	-377427594	1761333244	20
9	locations	1400836657646	-377427567	1761333309	20
10	locations	1400836743132	-377427562	1761333464	20
11	locations	1400836804182	-377427629	1761333341	32
12	locations	1400836868017	-377428072	1761333469	32



Once I had the data all sorted out, I then needed to create a new CSV that has both song data & location data so I could feed that into the application I'd make.

I created a Python script myself to scrape through both CSVs and merge them appropriately. The logic essentially was:

```
OPEN location-csv
OPEN last-fm-tracks-csv

LOOP through last-fm-tracks-csv
  READ timestamp value
  LOOP through location-csv
    Find closest timestamp within 6 hours
  PRINT timestamp, latitude & longitude, track information.
END LOOP
```

Here are some example rows in my final CSV:

	A	B	C	D	E	F	G	H
1	timestamp	latitude	longitude	activity	date	artist	track	
2	1400839244	-37.7427824	176.1333391	STILL	23-May-14	Björk	Harm of Will	
3	1400839485	-37.7427951	176.1333376	STILL	23-May-14	Danny Br	25 Bucks	
4	1400839886	-37.7428223	176.1333852	STILL	23-May-14	Gotye	Making Mirrors	

My final results had 39,831 entries of data. Down from 95,521 total songs listened.

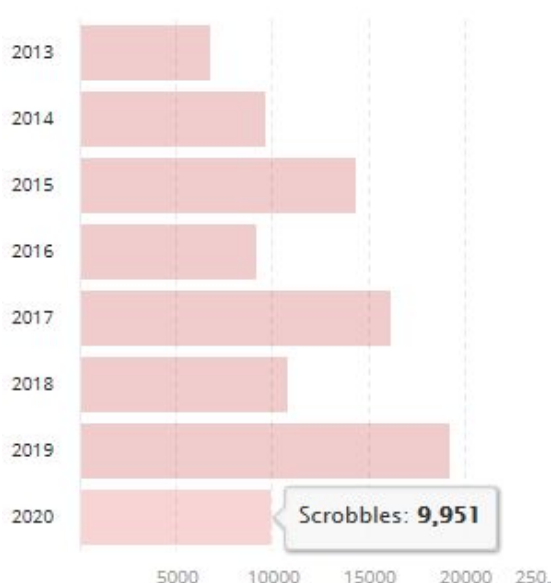
There are a few reasons I can think of that would mean skipped data:

- My song listens are tracked back to 2013, where my location data was only tracked since 2014. Missing out on a year's worth of songs.
- There was a period of late 2019 to now where my location tracking was disabled. Meaning my final results would be missing data from all of this year. I regret turning it off now as this meant I missed my data from my trip to Vietnam last year in November.
- Other periods Google Maps didn't track, either through error or me switching phones and not having location tracking going straight away.

For context here are my yearly listening amounts:

More can be found at [CaptainNikGreen's Music Profile | Last.fm](#)

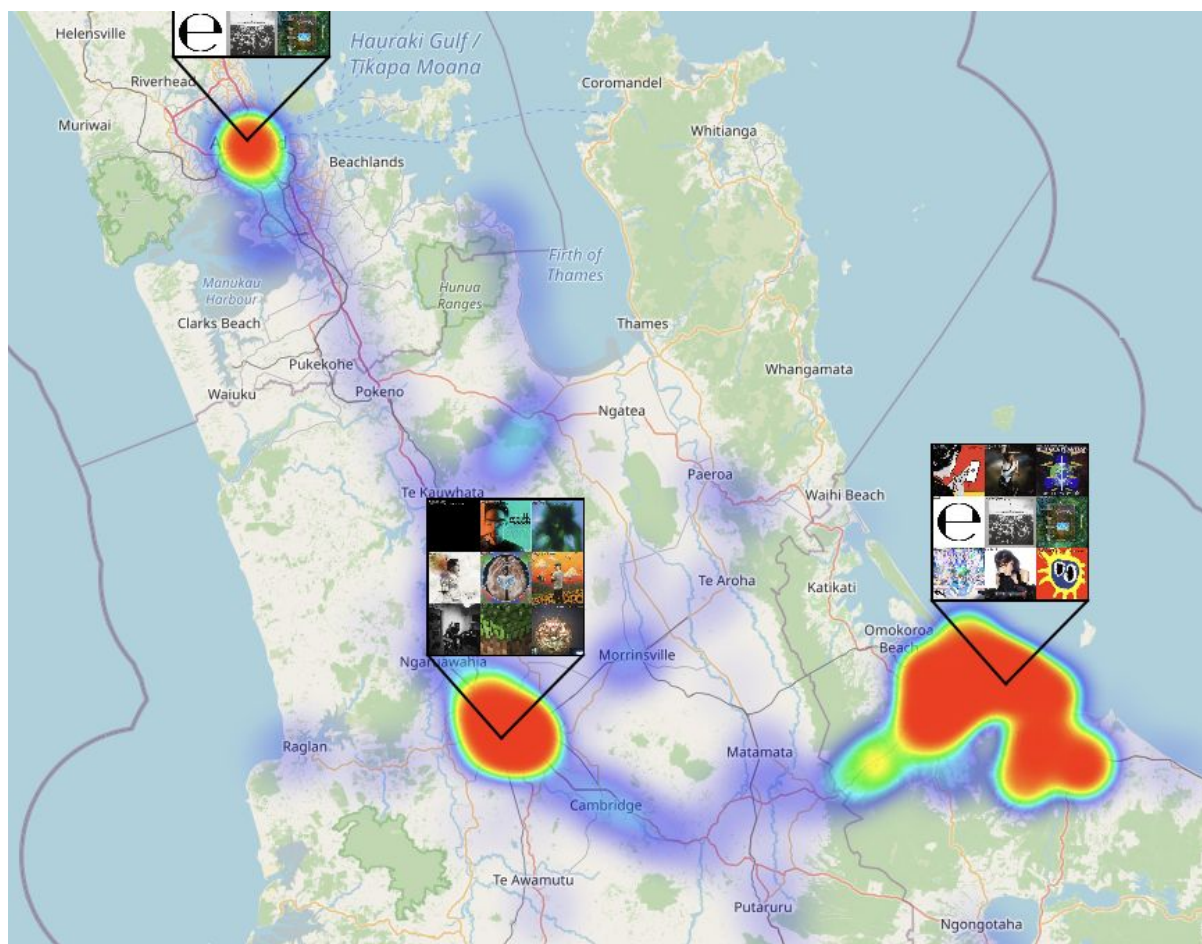
Listening History



# Application development.

Once I had my final CSV, with music data and location data per line. I knew I wanted to have these detailed over a Google Maps instance so the user can zoom in and see individual listens.

My original concept had an element floating over clustered areas on the map with a visual of albums listened in that area.

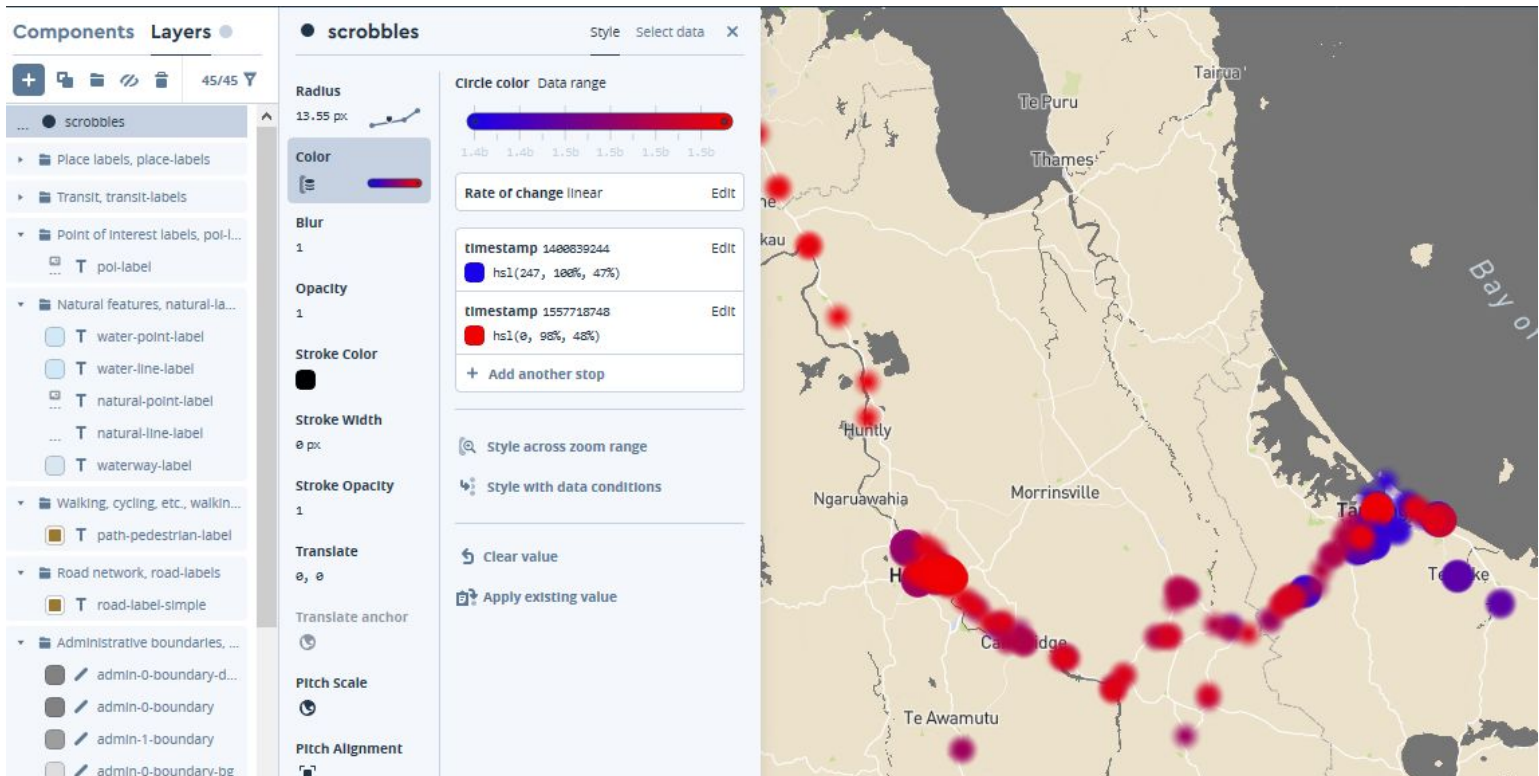


I quickly would realise that in order to get the album art actually in the application, it would require a lookup with official Last FM API to get the album art, then display it in my web application. Or a Google Image search with the right terms - but may return an incorrect result. This could also tax the performance of the application, as I had 39,831 entries to work with. With Last fm API capability I could pull extra information not found in my exported CSV, such as song length - which I could sum into minutes listened to on a location on my map.

I started out by Google-ing for any tool I could use to create my heatmap for me. This would lead me to find mapbox.com. Which is an online website that lets users create overlays on Google Maps. There's heaps of options for displaying data on top of the map.

I signed up and created a new style for displaying my points of data on the map.

This ended up being only used in my application for the visual, the rest was done in JavaScript myself, more on that later.



You can see this at [Mapbox.com](https://www.mapbox.com)

This website let me upload my 18mb CSV and render it on the map. It also allowed me to do the colour changing based on the timestamp data, with earlier timestamps in blue and more recent ones in red. This way at a glance I could see my trends across new zealand, so I could see the changes in what I listened to years ago - vs what I listen to nowadays.

Once I had the map looking the way I wanted, I could embed this map visual into a local HTML file that I would edit to become my final application.

For my application, I added the ability to draw a square on the map and then query the data points in that square. Results from that square are sorted and categorised and displayed onto the panels around the map (Activities detected, Top days, Top Artists, All results).

This main functionality is broken down as follows:

```

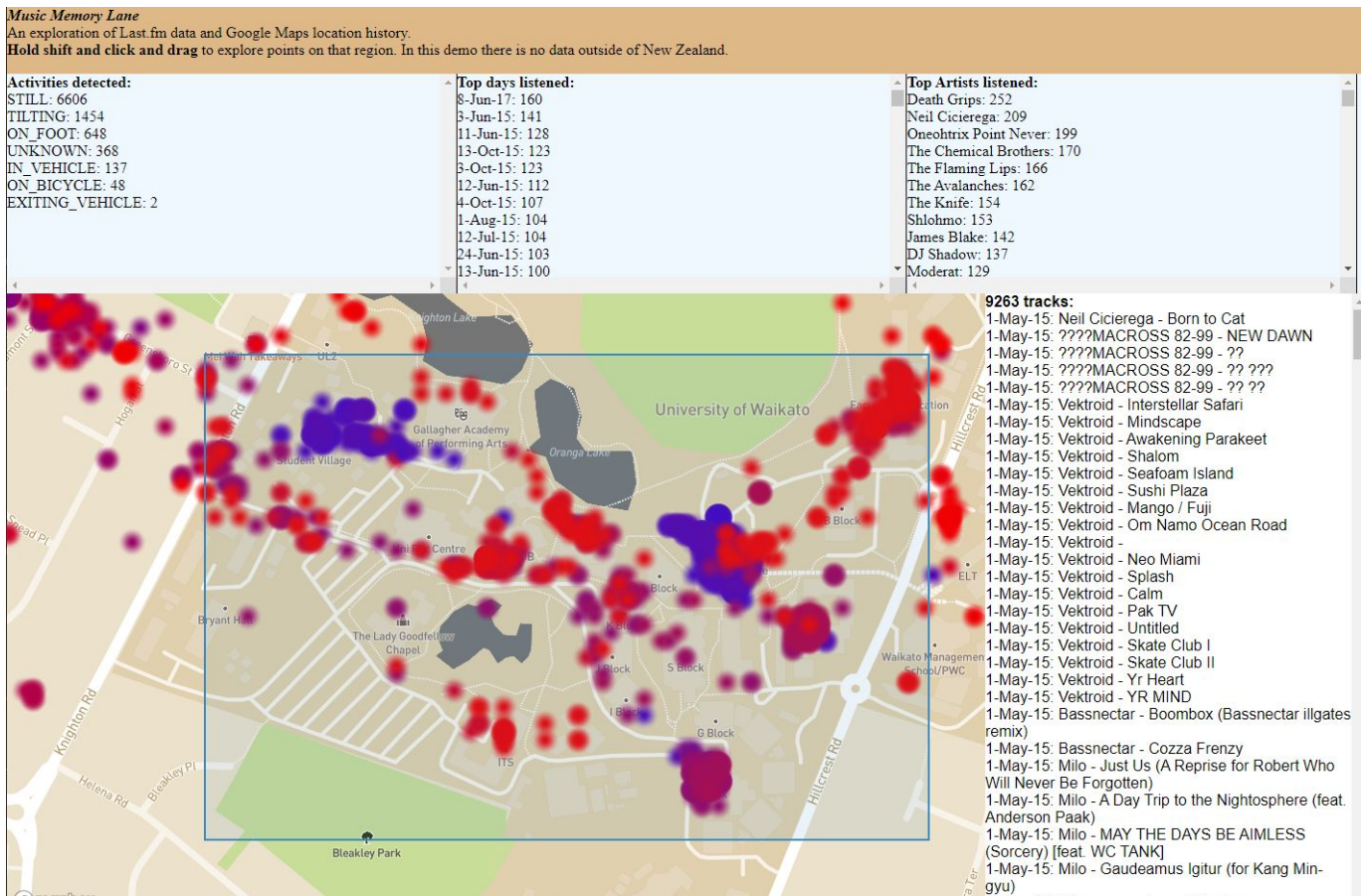
GET lat and long for first point of the square
GET lat and long for second point of the square

//Search through results for points that are in between those two locations
LOOP through master-csv
  IF within latA and latB
    IF within longA and longB
      Add this the results
  --
--
DISPLAY all the results on the web application.

```

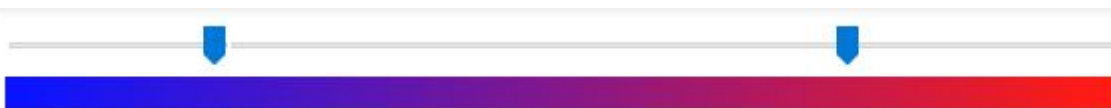


# Final result and thoughts



I'm pleased with the final result. The user can drill in and out of the map and select any area of their choosing. I was happy that I could have a solution that can take in so much data and display it all. Using colour to represent time, you can easily see places I was in the past and see trends there.

One thing I wanted to have would be a time slider so I could change the period shown. This slider could have been matched to the colours used also, so at a glance you can get an idea of the time represented.

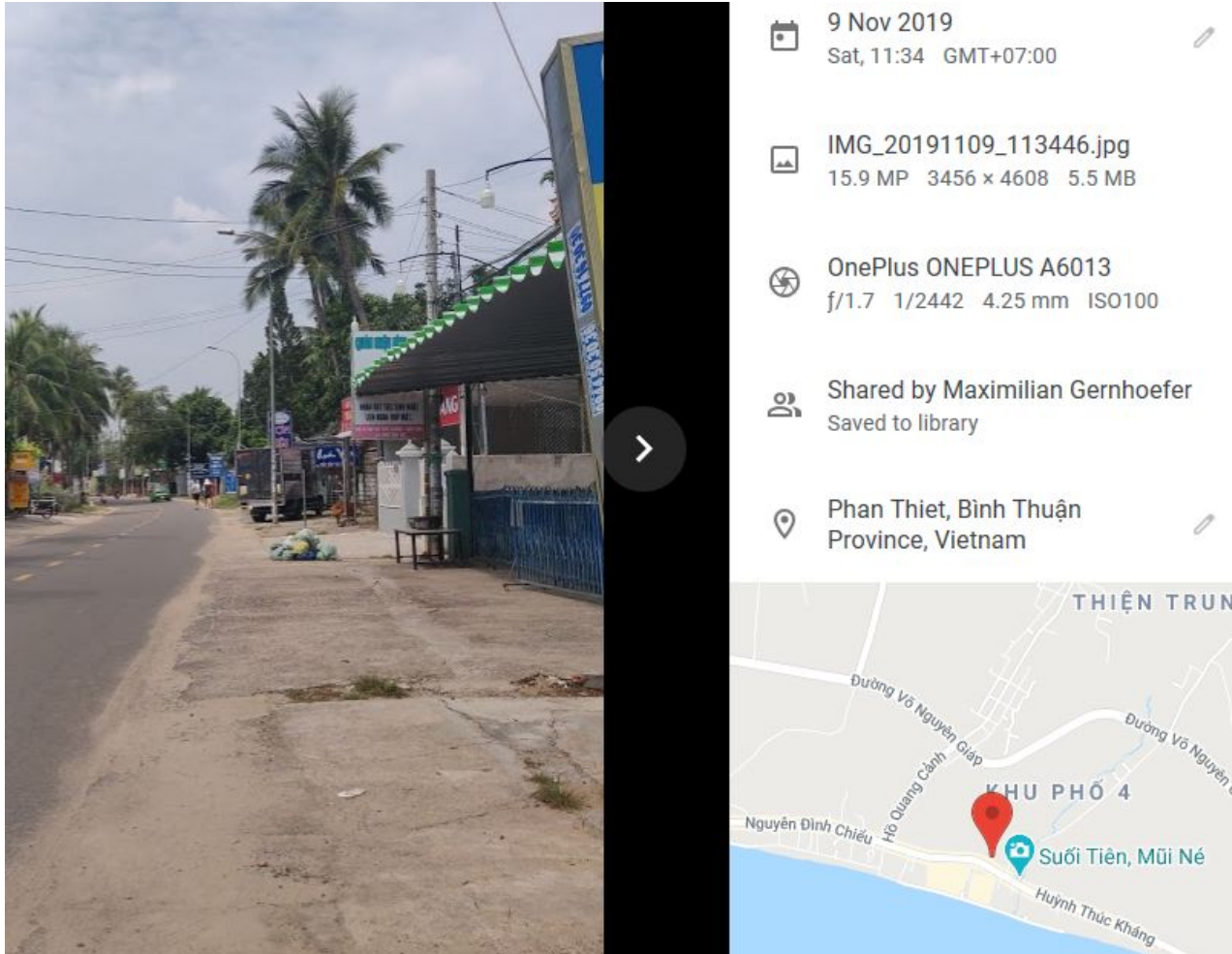


A downside is that this is static data. All coming from the datasets I downloaded originally from Google Maps and Last FM. With enough time/skill a online implementation could've been made using Last FM and Google's API.

Then anyone who uses both Last FM and has data on their Google Timeline could sign into my application and view their own data. This is entirely possible, but beyond the scope of this project.

One of the things I was originally hoping to see was what I was listening to during my trips to India in December 2017 and trip to Vietnam in November 2019. That didn't make it into my application because I think I didn't have my Google location tracking enabled both times.

However, in hindsight, I could have manually got location data from the photos I took on those trips and fed them into my location CSV.



All my images from Vietnam and India have location data in their EXIF metadata. I haven't looked into how I could automate getting that data, as I would rather not do it manually through hundreds of photos. If I had this data included as well, my application would show data outside of New Zealand and make a much more compelling application.

## Resources used.

### Data sources:

<https://lastfm.ghan.nl/export/>

<https://takeout.google.com/settings/takeout?pli=1>

### My Profile

<https://www.last.fm/user/captainnikgreen>

### Mapbox.com

<https://docs.mapbox.com/mapbox-gl-js/example/heatmap-layer/>

<https://docs.mapbox.com/mapbox-gl-js/example/using-box-queryrenderedfeatures/>

### Application implementation

Various Javascript documentation & my own Python and Javascript skills.