

[medium.com](https://medium.com)

# Tools I recommend for building Geospatial Web Applications

*Ramiz Sami*

7-8 minutes



Being a freelance WebGIS developer, it is very frequent for me to decide what tools I'll need for my next project. Now after a few years of experience, this decision has become a bit easier because of prior knowledge and I thought I should share it with all those who want to build a Geospatial Webapp.

Before you read further, I want you to know two things:

1. All the tools that I'll discuss in this blog are free and/or opensource, won't talk about paid and/or proprietary tools.
2. These tools can be used to build a small to medium scale web application. They will be able to work at scale with a lot of users but might or might not be able to handle very large geospatial datasets. If you want to handle large geospatial datasets in your web app, you might want to read "[How to work with BIG Geospatial Data](#)".

The most important part of a Geospatial web Application is the frontend where all the data is visualized which is the main focus in about 90% of geospatial web apps.

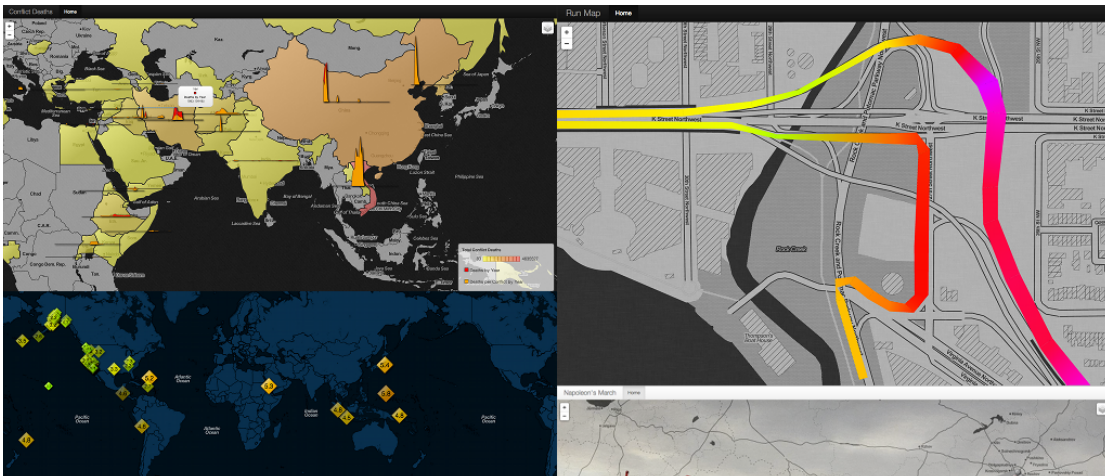
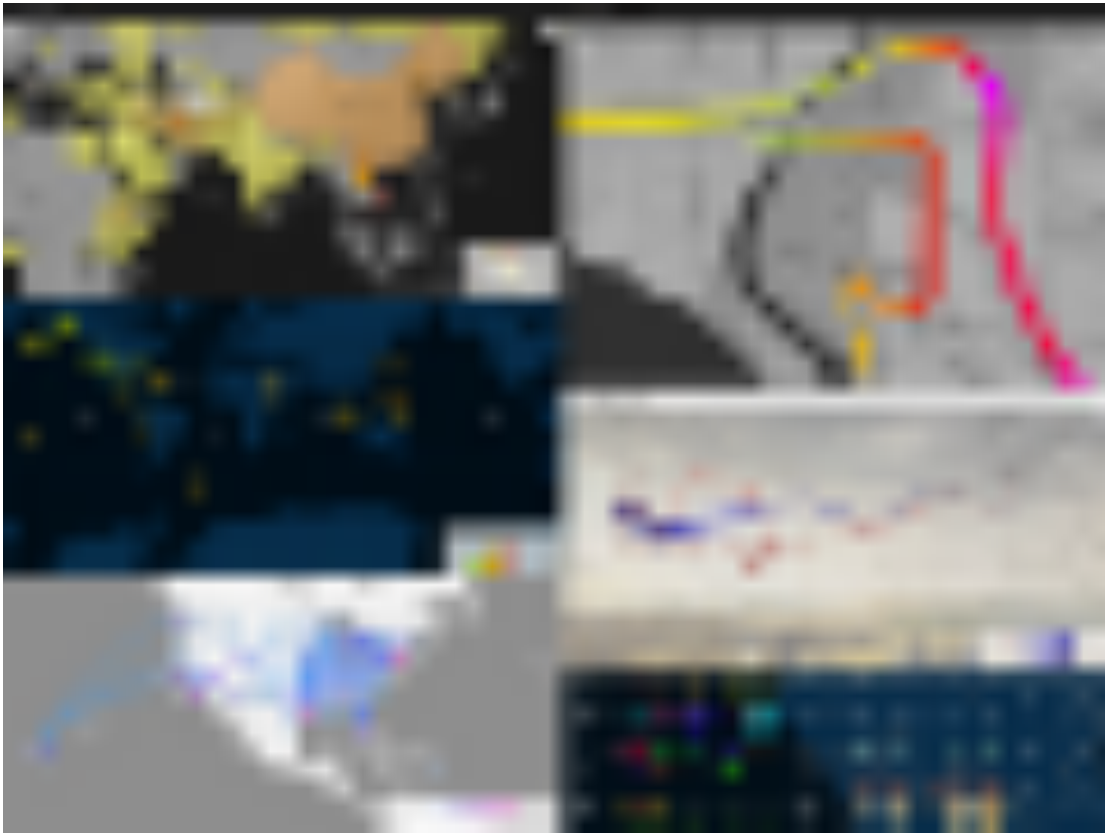
## Leaflet

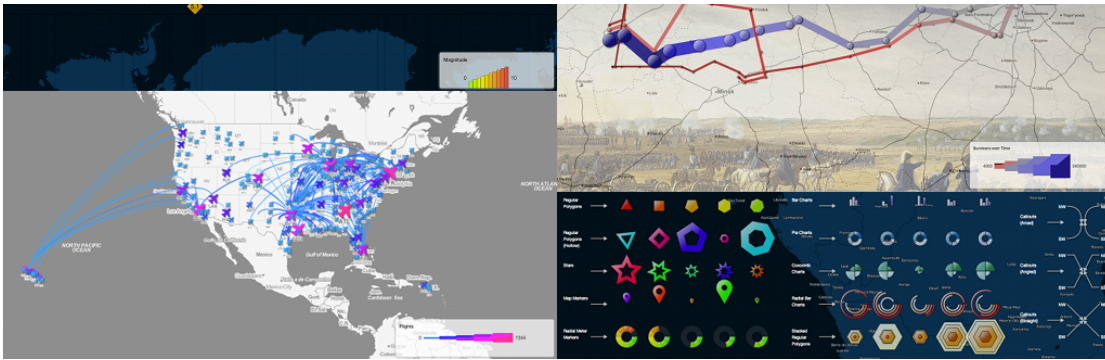
For the frontend, my first priority is [Leaflet](#) which is a lightweight javascript library for creating web maps.





The reason why I recommend it is because it is very simple to use and the map looks pretty modern. There is also a huge number of Leaflet [plugins](#) developed by the opensource community. One plugin that I'd particularly like to mention here is the [Leaflet Data Visualization Framework](#).





## Leaflet Data Visualization framework

Some of the prominent sites/portals that use Leaflet include [OpenStreetMap](#), [Craigslist](#), [USGS](#), [Windy](#), and [Data.gov](#), etc.

## Openlayers

My next recommendation for web mapping library after Leaflet is [Openlayers](#). You should use it when you need to do some advanced data manipulation and/or display on the web map.

Openlayers provides more functionalities as compared to Leaflet but on the downside, the learning curve of Openlayers is steeper.





Openlayers

## GeoExt

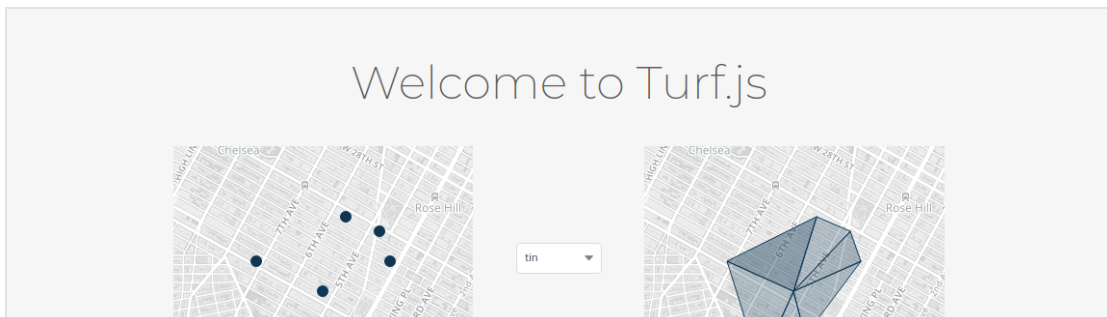
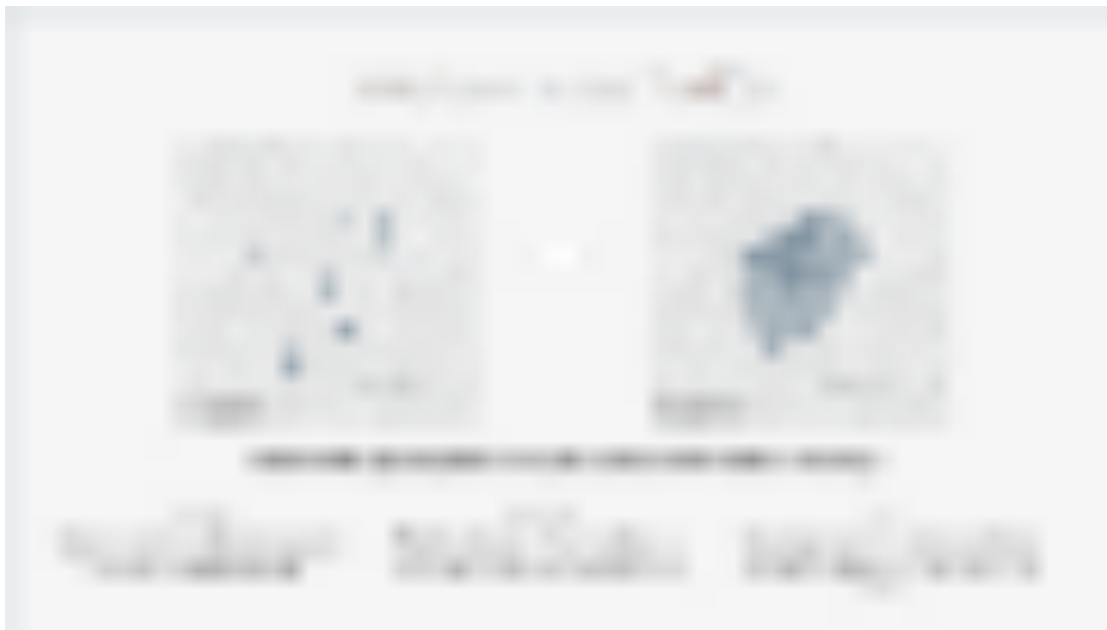
[GeoExt](#) is a complete and very powerful Javascript framework used for building 'desktop-like' geospatial interfaces on the web. It combines the mapping functionality of [OpenLayers](#) with the user interface of the [ExtJS](#) library and lets you create great Geospatial interfaces.

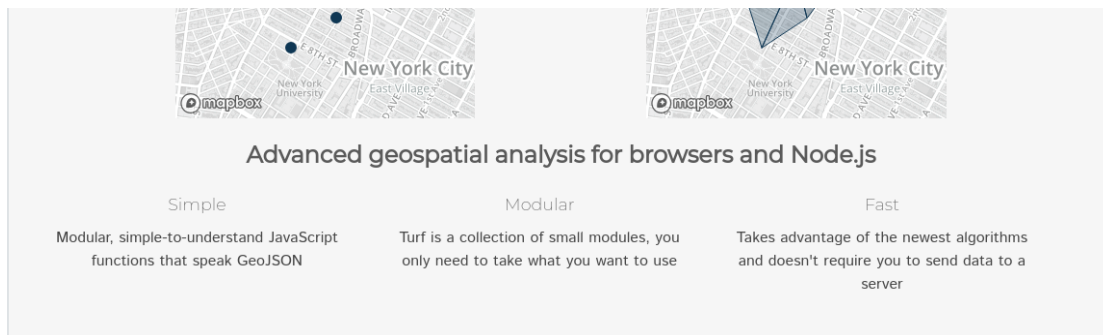




## Turf.js

Another very useful library for the frontend is [Turf.js](#). It is a Javascript library so you can also use it on the backend as well if you are using Node.js. It allows a lot of manipulation of geospatial data which comes handy in some cases if you can do it on the frontend.





## Turf.js

In a geospatial web app, a lot is going on in the backend. Most of the data manipulation and querying is done there. So it is important to choose the backend wisely.

For building a geospatial web app, I'll always recommend Python for the backend. The reason for that is because Python is very popular in the GIS community and hence a lot of powerful geospatial libraries are available in Python. These python libraries include [Shapely](#), [GeoPandas](#), [GDAL](#), [PyProj](#) and many others. Building the backend in Python enables you to use these libraries in your backend.

## GeoDjango

The framework I always use for the backend of a Geospatial web application is [Django](#). And I've got a lot of reasons for it. First of all, it is the most popular web framework in Python so there's a lot of community support available. Secondly, being a Python framework, it allows you to integrate all those powerful Geospatial libraries (we talked about above) in your backend.

The best part of it is [GeoDjango](#) which is a sub-framework of



Django. You can just integrate it into your Django project by adding `django.contrib.gis` in your installed apps.

It lets you store [geospatial data](#) types, [support](#) different geospatial databases, create geospatial forms, make geospatial queries, have a geospatially enabled admin panel, create [geospatial lookups](#), serialize Geojson objects, and do a lot more. It also comes with [GDAL](#) and [GEOS](#) APIs integrated with it.



Another great thing about Geodjango is the [Django-rest-framework-gis](#) which is a geospatial addon for the [Django](#)



<input type="checkbox"/>		Spearfish archeological sites	sf:archsites	sf	✓	EPSG:26713
<input type="checkbox"/>		Spearfish bug locations	sf:bugsites	sf	✓	EPSG:26713
<input type="checkbox"/>		Spearfish restricted areas	sf:restricted	sf	✓	EPSG:26713
<input type="checkbox"/>		Spearfish roads	sf:roads	sf	✓	EPSG:26713
<input type="checkbox"/>		Spearfish streams	sf:streams	sf	✓	EPSG:26713
<input type="checkbox"/>		Spearfish elevation	sf:sfDEM	sfDEM	✓	EPSG:26713
<input type="checkbox"/>		World rectangle	tiger:giant_polygon	nyc	✓	EPSG:4326
<input type="checkbox"/>		Manhattan (NY) points of interest	tiger:poi	nyc	✓	EPSG:4326
<input type="checkbox"/>		Manhattan (NY) landmarks	tiger:poly_landmarks	nyc	✓	EPSG:4326
<input type="checkbox"/>		Manhattan (NY) roads	tiger:tiger_roads	nyc	✓	EPSG:4326
<input type="checkbox"/>		USA Population	topp:states	states_shapefile	✓	EPSG:4326
<input type="checkbox"/>		Tasmania cities	topp:tasmania_cities	taz_shapes	✓	EPSG:4326
<input type="checkbox"/>		Tasmania roads	topp:tasmania_roads	taz_shapes	✓	EPSG:4326
<input type="checkbox"/>		Tasmania state boundaries	topp:tasmania_state_boundaries	taz_shapes	✓	EPSG:4326
<input type="checkbox"/>		Tasmania water bodies	topp:tasmania_water_bodies	taz_shapes	✓	EPSG:4326

<< < 1 > >> Results 1 to 19 (out of 19 items)

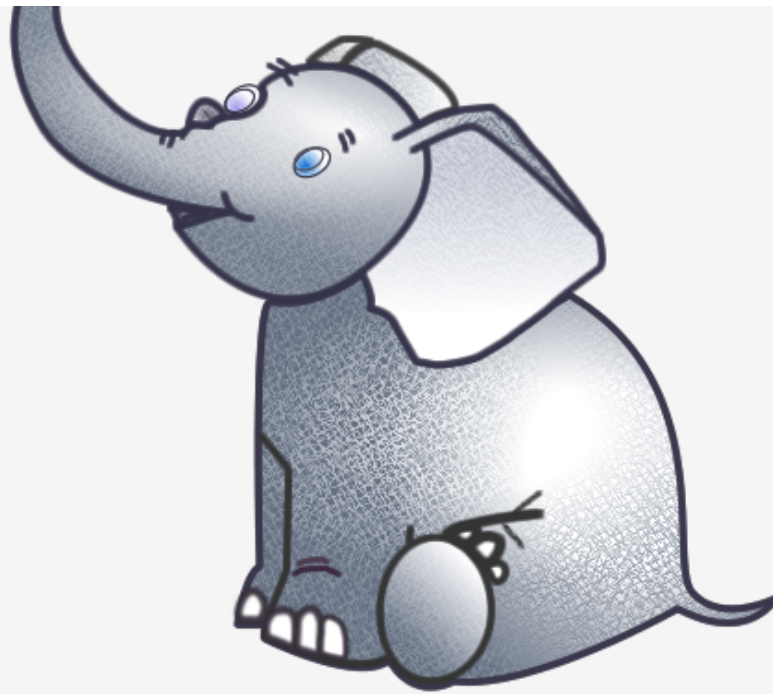
Geoserver is built in Java which runs on Apache tomcat. Using its UI, you can publish your geospatial data as web services without needing to code even one line. It supports both Raster and Vector data and is quite helpful when the main goal of your web app is to display data.

Your Geospatial data needs to be stored in a Geospatial database and we have different options for that like PostGIS, Mysql Spatial, Oracle Spatial, MongoDB, Spatialite, etc.

Among these, I always prefer [PostGIS](#) which is the spatial extension of PostgreSQL. I like it because of the fact that it is the most mature geospatial database out there and supports more Geospatial operations and functions as compared to its competitors.







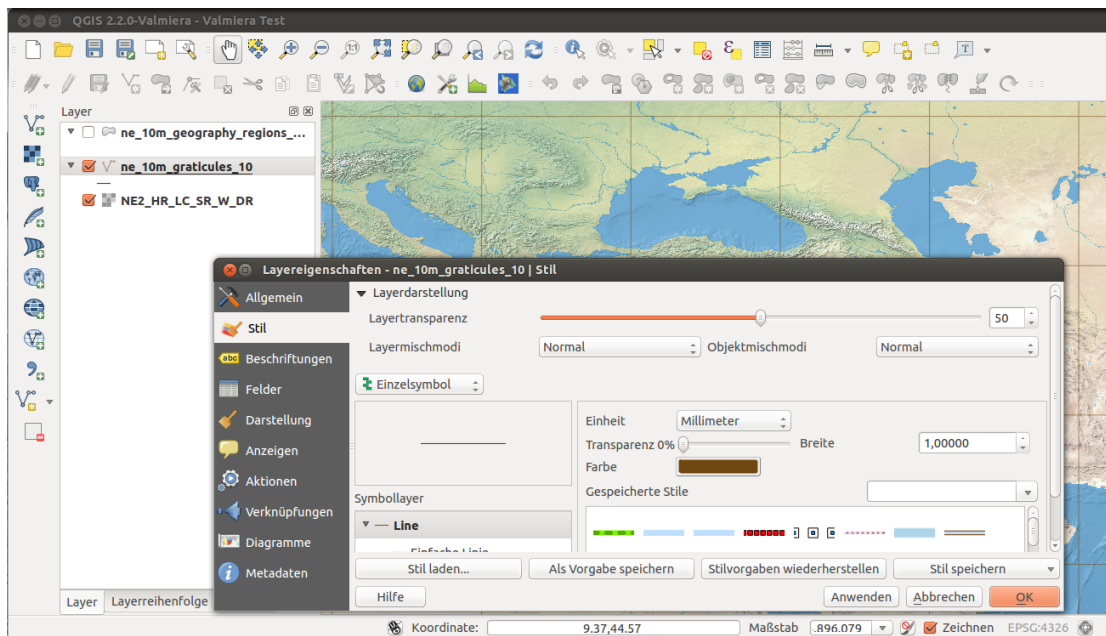
## Geospatial Objects for PostgreSQL

Many times there's a need to preprocess the geospatial data before being able to ingest it in the web application. So I'll talk about the tools I use for this.

### **QGIS**

QGIS is the most extensive free and opensource GIS software. It supports almost all sorts of geospatial operations and caters to almost all of my data preprocessing needs. The

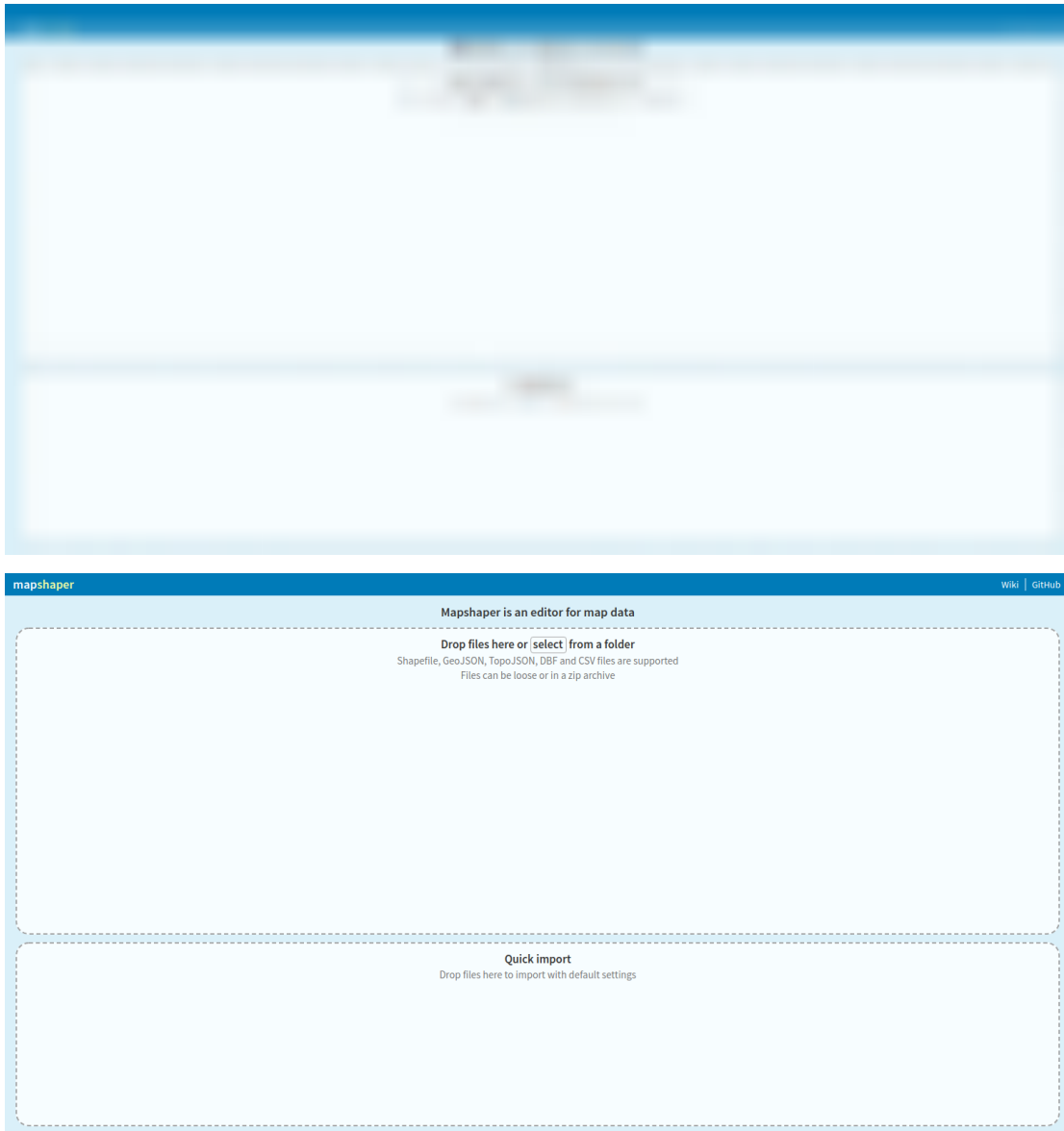
best thing about it is that (unlike Esri GIS) it is cross-platform compatible.



## Mapshaper

Another great tool for data preprocessing is [Mapshaper](#). It is a very simple site that allows for very simple operations like data

format conversion, data simplification, and data visualization.  
It only works with vector data.

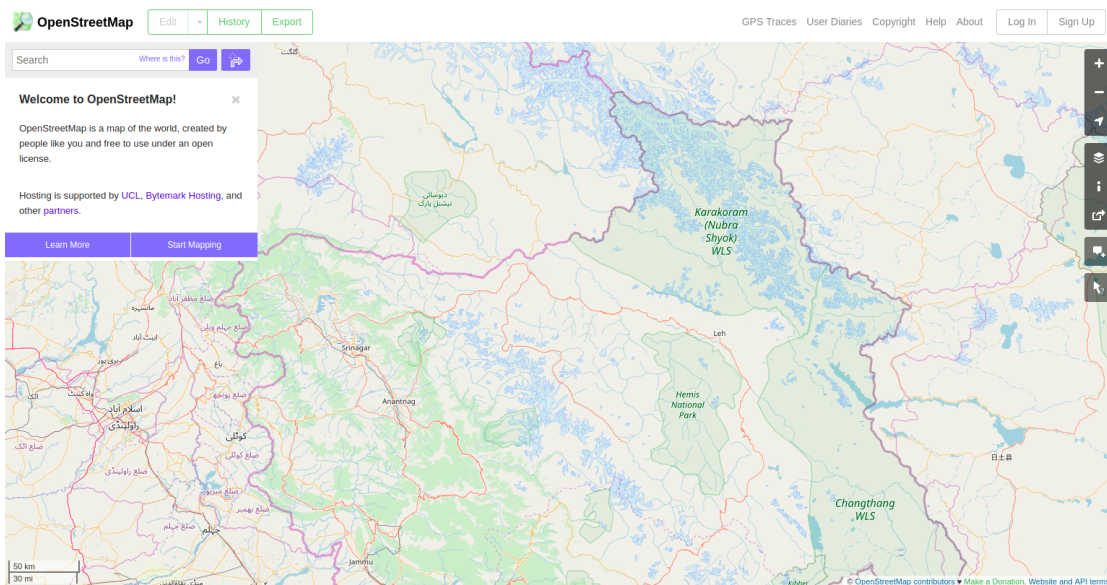
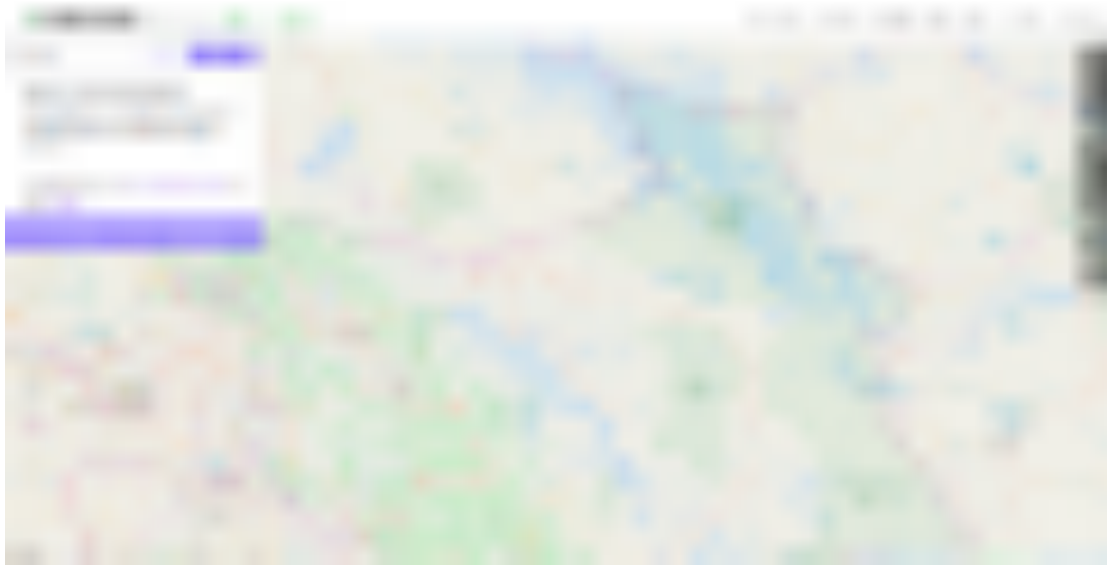


Many times you need to get the data from somewhere to add to your geospatial web app. For this, the two sources I like the most are OSM and Diva-GIS.

## OSM

[OpenStreetMap](#) is an opensource competitor of Google maps

and being opensource, its data is publicly available to download via UI and APIs. It is really helpful in many cases to get data from it. It also serves as Basemap in many web apps I develop. It also provides free routing and geocoding APIs.

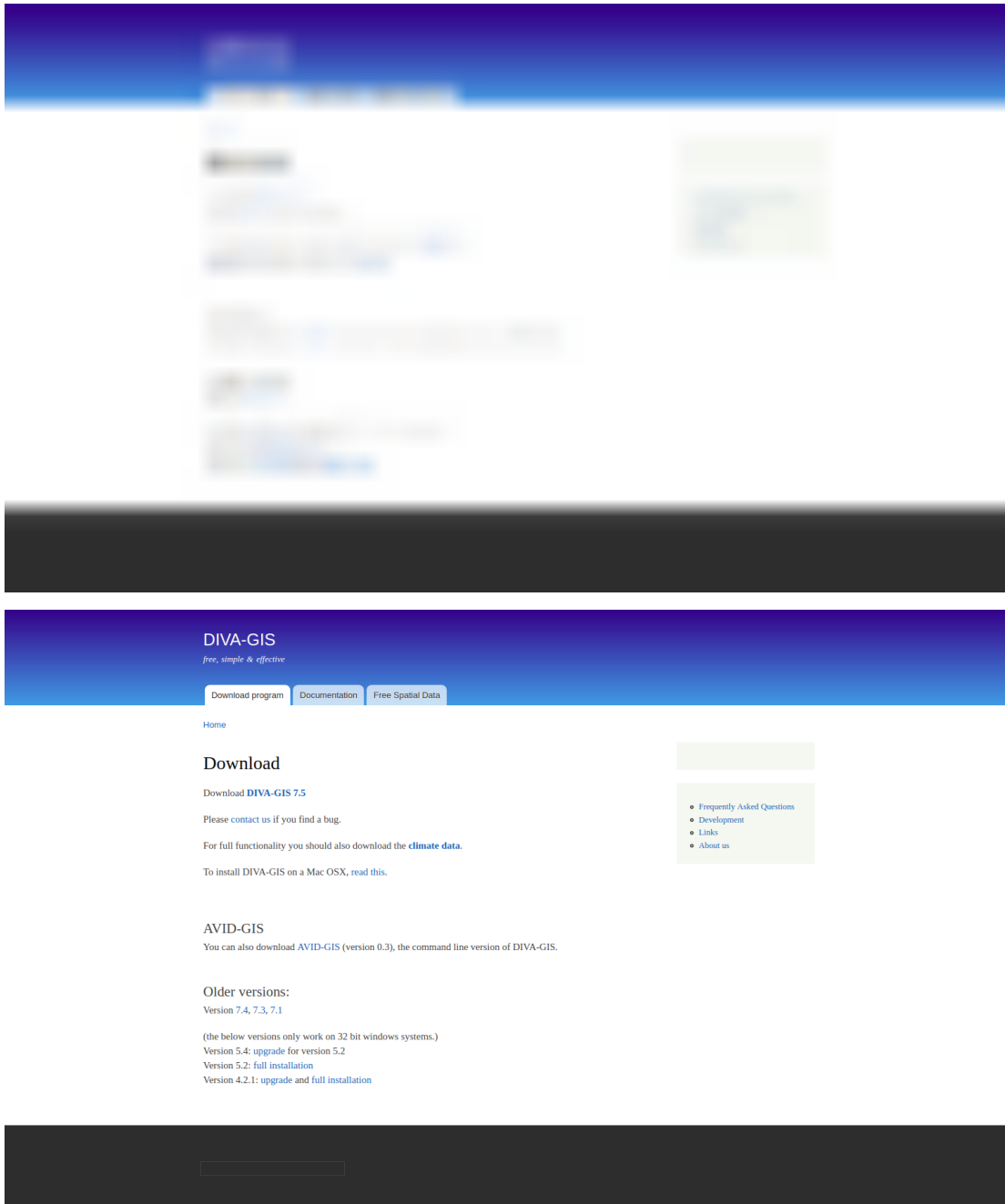


## Diva-GIS

Another very convenient data source that I frequently use is [Diva-GIS](#). It provides free spatial data like different levels of



administrative boundaries, railroads, roads, water, etc. It also provides global datasets about climate, species occurrence, crops, elevation, and others.



There's no fixed answer to the question "which tools you should use for your Geospatial web app?" Every web application is unique and so is the combination of tools and

technologies that fit best for its requirements. There's no hard and fast rule about it and usually, it is pretty flexible.

I am Ramiz Sami. I climb mountains, lift heavy weights and build WebGIS solutions. Feel free to get connected with me on [LinkedIn](#).