Location Trail

In [1]:

```
#Import all the required Python packages

import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from mpl_toolkits.basemap import Basemap
import json
import datetime
```

In [2]:

```
#opening json file

with open(r"C:\Users\Lenovo\Desktop\LocationHistory.json.json", 'r') as fh:
    raw = json.loads(fh.read())

#Creating data frame
ld1 = pd.DataFrame(raw)
ld1.head()

ld = pd.DataFrame(raw['locations'])
#print(ld)
ld.head()
```

Out[2]:

	accuracy	activity	altitude	heading	latitudeE7	longitudeE7	timestampMs	velocity
0	8	[{'timestampMs': '1459337601422', 'activity': ...	NaN	NaN	286140644	773315975	1459337584762	NaN
1	6	NaN	NaN	NaN	286141145	773316091	1459337629757	NaN
2	73	[{'timestampMs': '1459417836273', 'activity': ...	NaN	NaN	286140148	773316689	1459417832884	NaN
3	131	[{'timestampMs': '1459417961283', 'activity': ...	NaN	NaN	286139969	773317500	1459417960931	NaN
4	78	[{'timestampMs': '1459418103430', 'activity': ...	NaN	NaN	286140502	773317174	1459418083448	NaN

In [3]:

```
del raw #free up some memory
```

In [4]:

```
# convert to typical units

ld['latitudeE7'] = ld['latitudeE7']/float(1e7)
ld['longitudeE7'] = ld['longitudeE7']/float(1e7)
ld['timestamp'] = ld['timestampMs'].map(lambda x: float(x)/1000) #to seconds
ld['TimeStamp'] = ld.timestamp.map(datetime.datetime.fromtimestamp)

# Rename fields based on the conversions we just did
ld.rename(columns={'latitudeE7':'latitude', 'longitudeE7':'longitude', 'timestampMs':'timestamp'}, inplace=True)
ld = ld[ld.accuracy < 1000] #Ignore locations with accuracy estimates over 1000m
ld.reset_index(drop=True, inplace=True)
ld.drop(columns='heading',inplace=True)
ld.drop(columns='activity',inplace=True)
ld.drop(columns='altitude',inplace=True)
ld.drop(columns='timestamp',inplace=True)
ld.drop(columns='velocity',inplace=True)
ld.drop(columns='verticalAccuracy',inplace=True)

ld.head()
```

Out[4]:

	accuracy	latitude	longitude	TimeStamp
0	8	28.614064	77.331598	2016-03-30 17:03:04.762
1	6	28.614114	77.331609	2016-03-30 17:03:49.757
2	73	28.614015	77.331669	2016-03-31 15:20:32.884
3	131	28.613997	77.331750	2016-03-31 15:22:40.931
4	78	28.614050	77.331717	2016-03-31 15:24:43.448

In [5]:

```
#making a single column of latutude and longitute
ld['LatLong'] = ld[['latitude', 'longitude']].apply(tuple, axis=1)
ld.head()
```

Out[5]:

	accuracy	latitude	longitude	TimeStamp	LatLong
0	8	28.614064	77.331598	2016-03-30 17:03:04.762	(28.6140644, 77.3315975)
1	6	28.614114	77.331609	2016-03-30 17:03:49.757	(28.6141145, 77.3316091)
2	73	28.614015	77.331669	2016-03-31 15:20:32.884	(28.6140148, 77.3316689)
3	131	28.613997	77.331750	2016-03-31 15:22:40.931	(28.6139969, 77.33175)
4	78	28.614050	77.331717	2016-03-31 15:24:43.448	(28.6140502, 77.3317174)

In [6]:

```
# A Python library for offline reverse geocoding.
# Reverse Geocoder takes a latitude / longitude coordinate and returns the nearest town/city.
# Reverse geocoding is the process of converting geographic coordinates into a human-readable address

import reverse_geocoder as rg
results = rg.search(list(ld['LatLong']))
ld['country'] = [r['cc'] for r in results]
ld['city'] = [r['admin1'] for r in results]
ld['district'] = [r['admin2'] for r in results]
ld.head()
```

```
Loading formatted geocoded file...
```

Out[6]:

	accuracy	latitude	longitude	TimeStamp	LatLong	country	city	district
0	8	28.614064	77.331598	2016-03-30 17:03:04.762	(28.6140644, 77.3315975)	IN	Uttar Pradesh	Gautam Buddha Nagar
1	6	28.614114	77.331609	2016-03-30 17:03:49.757	(28.6141145, 77.3316091)	IN	Uttar Pradesh	Gautam Buddha Nagar
2	73	28.614015	77.331669	2016-03-31 15:20:32.884	(28.6140148, 77.3316689)	IN	Uttar Pradesh	Gautam Buddha Nagar
3	131	28.613997	77.331750	2016-03-31 15:22:40.931	(28.6139969, 77.33175)	IN	Uttar Pradesh	Gautam Buddha Nagar
4	78	28.614050	77.331717	2016-03-31 15:24:43.448	(28.6140502, 77.3317174)	IN	Uttar Pradesh	Gautam Buddha Nagar

In [7]:

```
# -> Geopy makes it easy for Python developers to locate the coordinates of ad
dresses, cities, countries, and landmarks across
# the globe using third-party geocoders and other data sources.
# ->This package provides the ratelimiter module, which ensures that an operat
ion will not be executed more than a given
# number of times on a given period.

import geopy
from geopy.geocoders import Nominatim
from geopy.extra.rate_limiter import RateLimiter

# taking small data
subld = ld.head(20).copy(deep=True)
print(subld)
```

```
    accuracy   latitude  longitude               TimeStamp  \
0          8  28.614064  77.331598 2016-03-30 17:03:04.762   
1          6  28.614114  77.331609 2016-03-30 17:03:49.757   
2         73  28.614015  77.331669 2016-03-31 15:20:32.884   
3        131  28.613997  77.331750 2016-03-31 15:22:40.931   
4         78  28.614050  77.331717 2016-03-31 15:24:43.448   
5         27  28.613989  77.331677 2016-03-31 15:26:25.358   
6         74  28.614006  77.331693 2016-03-31 15:28:30.734   
7         69  28.613949  77.331686 2016-03-31 15:30:30.957   
8         73  28.613976  77.331728 2016-03-31 15:34:23.943   
9         72  28.613995  77.331667 2016-03-31 15:37:46.083   
10        59  28.614026  77.331655 2016-03-31 15:41:47.193   
11        38  28.614034  77.331693 2016-03-31 15:46:58.838   
12        20  28.614017  77.331626 2016-03-31 15:52:36.328   
13        53  28.613996  77.331732 2016-03-31 15:57:05.369   
14        36  28.614013  77.331702 2016-03-31 16:02:57.928   
15        29  28.614059  77.331718 2016-03-31 16:17:59.499   
16        57  28.614023  77.331678 2016-03-31 16:53:09.211   
17        54  28.614032  77.331706 2016-03-31 17:09:09.635   
18        50  28.614042  77.331762 2016-03-31 17:09:59.544   
19        17  28.614103  77.331564 2016-04-04 16:54:49.358   

                   LatLong country           city             district  
0   (28.6140644, 77.3315975)      IN  Uttar Pradesh  Gautam Buddha Nagar  
1   (28.6141145, 77.3316091)      IN  Uttar Pradesh  Gautam Buddha Nagar  
2   (28.6140148, 77.3316689)      IN  Uttar Pradesh  Gautam Buddha Nagar  
3     (28.6139969, 77.33175)      IN  Uttar Pradesh  Gautam Buddha Nagar  
4   (28.6140502, 77.3317174)      IN  Uttar Pradesh  Gautam Buddha Nagar  
5    (28.613989, 77.3316772)      IN  Uttar Pradesh  Gautam Buddha Nagar  
6   (28.6140059, 77.3316934)      IN  Uttar Pradesh  Gautam Buddha Nagar  
7   (28.6139493, 77.3316858)      IN  Uttar Pradesh  Gautam Buddha Nagar  
8   (28.6139763, 77.3317277)      IN  Uttar Pradesh  Gautam Buddha Nagar  
9   (28.6139953, 77.3316675)      IN  Uttar Pradesh  Gautam Buddha Nagar  
10  (28.6140257, 77.3316547)      IN  Uttar Pradesh  Gautam Buddha Nagar  
11  (28.6140336, 77.3316926)      IN  Uttar Pradesh  Gautam Buddha Nagar  
12   (28.614017, 77.3316262)      IN  Uttar Pradesh  Gautam Buddha Nagar  
13  (28.6139959, 77.3317317)      IN  Uttar Pradesh  Gautam Buddha Nagar  
14  (28.6140133, 77.3317016)      IN  Uttar Pradesh  Gautam Buddha Nagar  
15  (28.6140595, 77.3317181)      IN  Uttar Pradesh  Gautam Buddha Nagar  
16  (28.6140225, 77.3316776)      IN  Uttar Pradesh  Gautam Buddha Nagar  
17  (28.6140324, 77.3317059)      IN  Uttar Pradesh  Gautam Buddha Nagar  
18  (28.6140419, 77.3317621)      IN  Uttar Pradesh  Gautam Buddha Nagar  
19  (28.6141035, 77.3315638)      IN  Uttar Pradesh  Gautam Buddha Nagar  
```

In [8]:

```
# getting location using longitude and latitude
geolocator = Nominatim(user_agent="geoapiExercises")
lald = "28.614064, 77.331598"
print("Latitude and Longitude:",lald)
location = geolocator.reverse("28.614064, 77.331598")
print(location.address)
```

```
Latitude and Longitude: 28.614064, 77.331598
Gharoli Dairy, Samaspur, Mayur Vihar Tehsil, East Delhi, Delhi, 201301, India
```

In [9]:

```
geolocator = Nominatim(user_agent="geoapiExercises")
rgeocode = RateLimiter(geolocator.reverse, min_delay_seconds=0.001)
subld['address'] = subld['LatLong'].apply(rgeocode)
subld.head()
```

Out[9]:

	accuracy	latitude	longitude	TimeStamp	LatLong	country	city	district	addre
0	8	28.614064	77.331598	2016-03-30 17:03:04.762	(28.6140644, 77.3315975)	IN	Uttar Pradesh	Gautam Buddha Nagar	(Gha Da Samasp Ma Vi Tehsil
1	6	28.614114	77.331609	2016-03-30 17:03:49.757	(28.6141145, 77.3316091)	IN	Uttar Pradesh	Gautam Buddha Nagar	(Gha Da Samasp Ma Vi Tehsil
2	73	28.614015	77.331669	2016-03-31 15:20:32.884	(28.6140148, 77.3316689)	IN	Uttar Pradesh	Gautam Buddha Nagar	(Gha Da Samasp Ma Vi Tehsil
3	131	28.613997	77.331750	2016-03-31 15:22:40.931	(28.6139969, 77.33175)	IN	Uttar Pradesh	Gautam Buddha Nagar	(Gha Da Ma Vihar Samasp Ma
4	78	28.614050	77.331717	2016-03-31 15:24:43.448	(28.6140502, 77.3317174)	IN	Uttar Pradesh	Gautam Buddha Nagar	(Gha Da Samasp Ma Vi Tehsil

In [10]:
```
# creating a dataframe of two requires columns
loc=subld.loc[:,['latitude', 'longitude']]
print(loc)
```

```
     latitude  longitude
0   28.614064  77.331598
1   28.614114  77.331609
2   28.614015  77.331669
3   28.613997  77.331750
4   28.614050  77.331717
5   28.613989  77.331677
6   28.614006  77.331693
7   28.613949  77.331686
8   28.613976  77.331728
9   28.613995  77.331667
10  28.614026  77.331655
11  28.614034  77.331693
12  28.614017  77.331626
13  28.613996  77.331732
14  28.614013  77.331702
15  28.614059  77.331718
16  28.614023  77.331678
17  28.614032  77.331706
18  28.614042  77.331762
19  28.614103  77.331564
```

In [11]:
```
sub_ld= ld.head(6).copy(deep=True)
print(sub_ld)
```

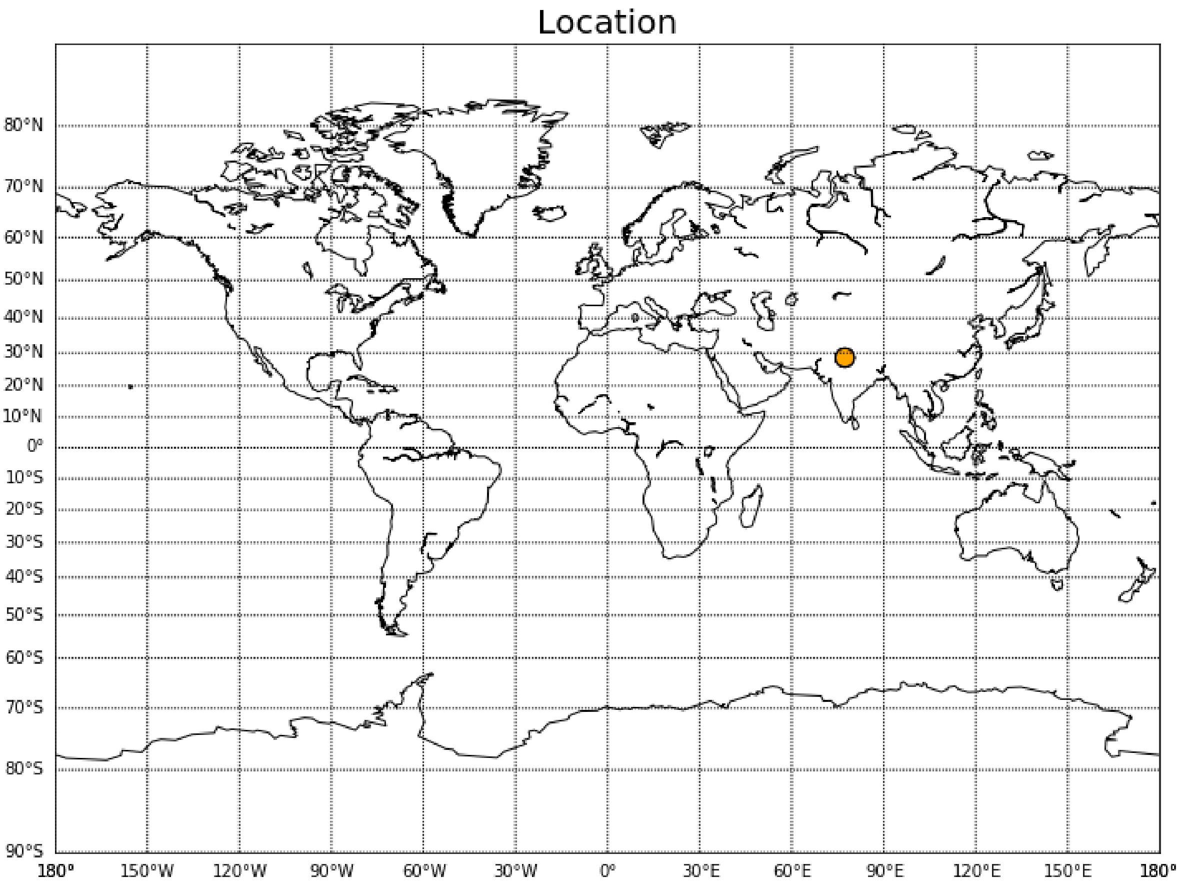
```
   accuracy   latitude  longitude               TimeStamp  \
0         8  28.614064  77.331598 2016-03-30 17:03:04.762
1         6  28.614114  77.331609 2016-03-30 17:03:49.757
2        73  28.614015  77.331669 2016-03-31 15:20:32.884
3       131  28.613997  77.331750 2016-03-31 15:22:40.931
4        78  28.614050  77.331717 2016-03-31 15:24:43.448
5        27  28.613989  77.331677 2016-03-31 15:26:25.358

                   LatLong country           city             district
0  (28.6140644, 77.3315975)      IN  Uttar Pradesh  Gautam Buddha Nagar
1  (28.6141145, 77.3316091)      IN  Uttar Pradesh  Gautam Buddha Nagar
2  (28.6140148, 77.3316689)      IN  Uttar Pradesh  Gautam Buddha Nagar
3    (28.6139969, 77.33175)      IN  Uttar Pradesh  Gautam Buddha Nagar
4  (28.6140502, 77.3317174)      IN  Uttar Pradesh  Gautam Buddha Nagar
5   (28.613989, 77.3316772)      IN  Uttar Pradesh  Gautam Buddha Nagar
```

In [12]:
```
# finding maximum and minimum
val = (ld.longitude.min(),ld.longitude.max(),ld.latitude.min(), ld.latitude.ma
x())
print(val)
```

```
(76.1028194, 79.530445, 28.2570495, 32.1033445)
```

In [13]:

```
#Basemap is a great tool for creating maps using python in a simple way. It’s a matplotlib extension,
#so it has got all its features to create data visualizations, and adds the geographical projections
#and some datasets to be able to plot coast lines, countries, and so on directly from the library.

from mpl_toolkits.basemap import Basemap
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

fig = plt.figure(figsize=(12,9))

m = Basemap(projection='mill',
           llcrnrlat = -90,
           urcrnrlat = 90,
           llcrnrlon = -180,
           urcrnrlon = 180,
           resolution = 'c')

m.drawcoastlines()

m.drawparallels(np.arange(-90,90,10),labels=[True,False,False,False])
m.drawmeridians(np.arange(-180,180,30),labels=[0,0,0,1])

sites_lat_y = sub_ld['latitude'].tolist()
sites_lon_x = sub_ld['longitude'].tolist()

colors = ['green', 'darkblue', 'yellow', 'red', 'blue', 'orange']

m.scatter(sites_lon_x,sites_lat_y,latlon=True, s=500, c=colors, marker='.', alpha=1, edgecolor='k', linewidth=1, zorder=2)
#m.scatter(-135,60,latlon=True, s=5000, c='blue', marker='^', alpha=1, edgecolor='k', linewidth=1, zorder=1)

plt.title('Location', fontsize=20)

plt.show()
```



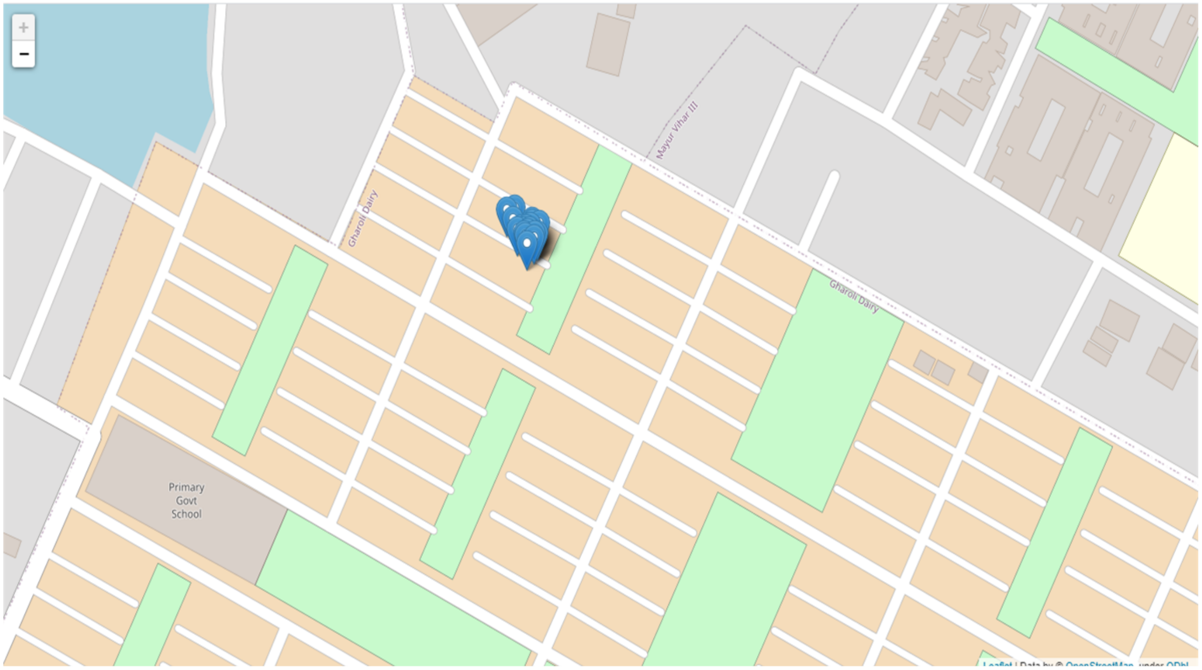
In [14]:

```
#Folium is a Python Library that can allow us to visualize spatial data in an interactive manner
import folium
mf=folium.Map(location=[28.6140644, 77.3315975],zoom_start=12)
mf=folium.Map(location=[28.6139969, 77.33175],zoom_start=12)
loc.apply(lambda row:folium.Marker(location=[row['latitude'], row['longitude']]).add_to(mf), axis=1)
mf.save("mf.html")
```

In [15]:

```
from IPython.display import Image
Image(filename="pic.png")
```

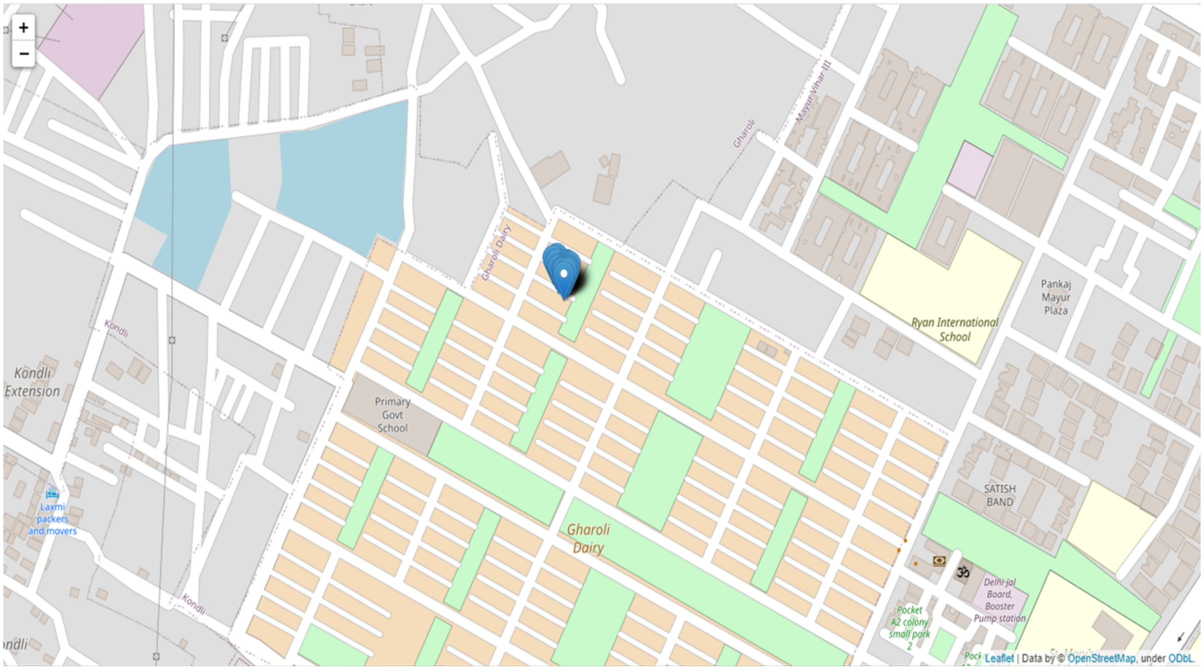
Out[15]:



In [16]:

```
Image(filename="pic1.png")
```

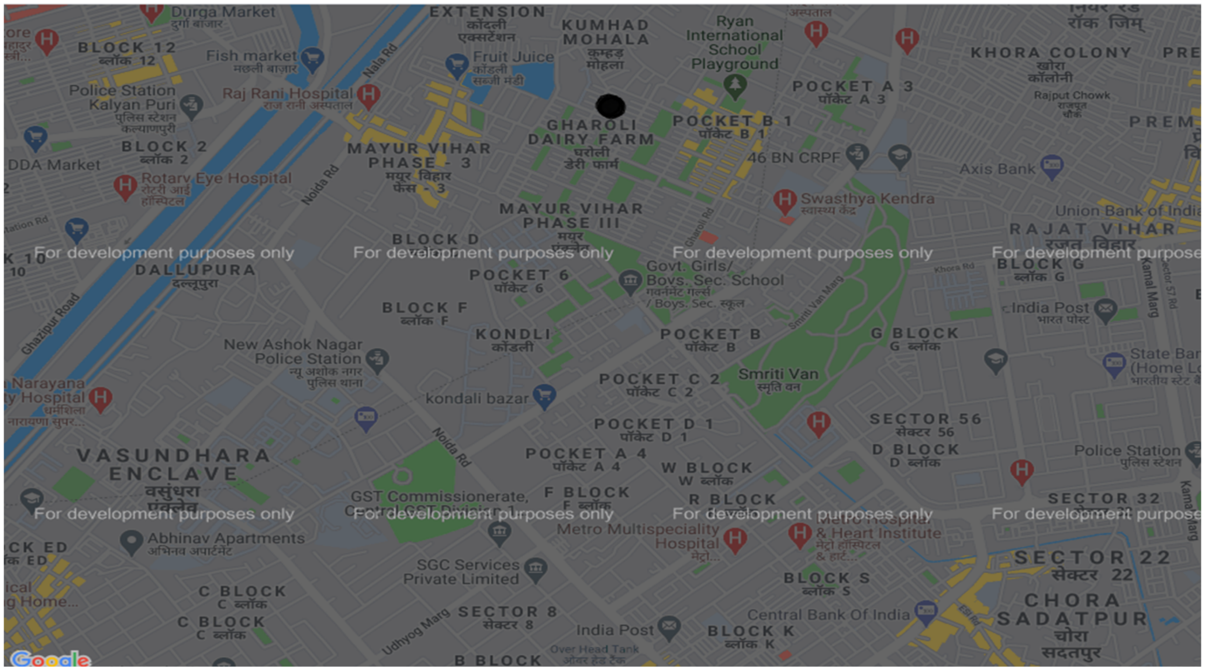
Out[16]:



In [17]:

```
# Python library gmplot allows us to plot data on google maps.
import gmplot

# Place map
gmap = gmplot.GoogleMapPlotter(28.6140644, 77.3315975, 13)

# Scatter points
top_attraction_lats, top_attraction_lons = zip(*[
    (28.614064,  77.331598),
   (28.614114,  77.331609),
   (28.614015, 77.331669),
   (28.613997,  77.331750),
   (28.614050,  77.331717),
   (28.613989, 77.331677),
   (28.614006, 77.331693),
   (28.613949,  77.331686)
    ])

#plotting location
gmap.scatter(top_attraction_lats, top_attraction_lons, '00FFFF', size=40, mark
er=False) #3B0B39

# Marker
hidden_gem_lat, hidden_gem_lon = 28.6140644, 77.3315975
gmap.marker(hidden_gem_lat, hidden_gem_lon, 'cornflowerblue')

# Location where you want to save your file
gmap.draw("my_map.html")

Image(filename="pic3.png")
```

Out[17]:



In [ ]:

In [ ]: