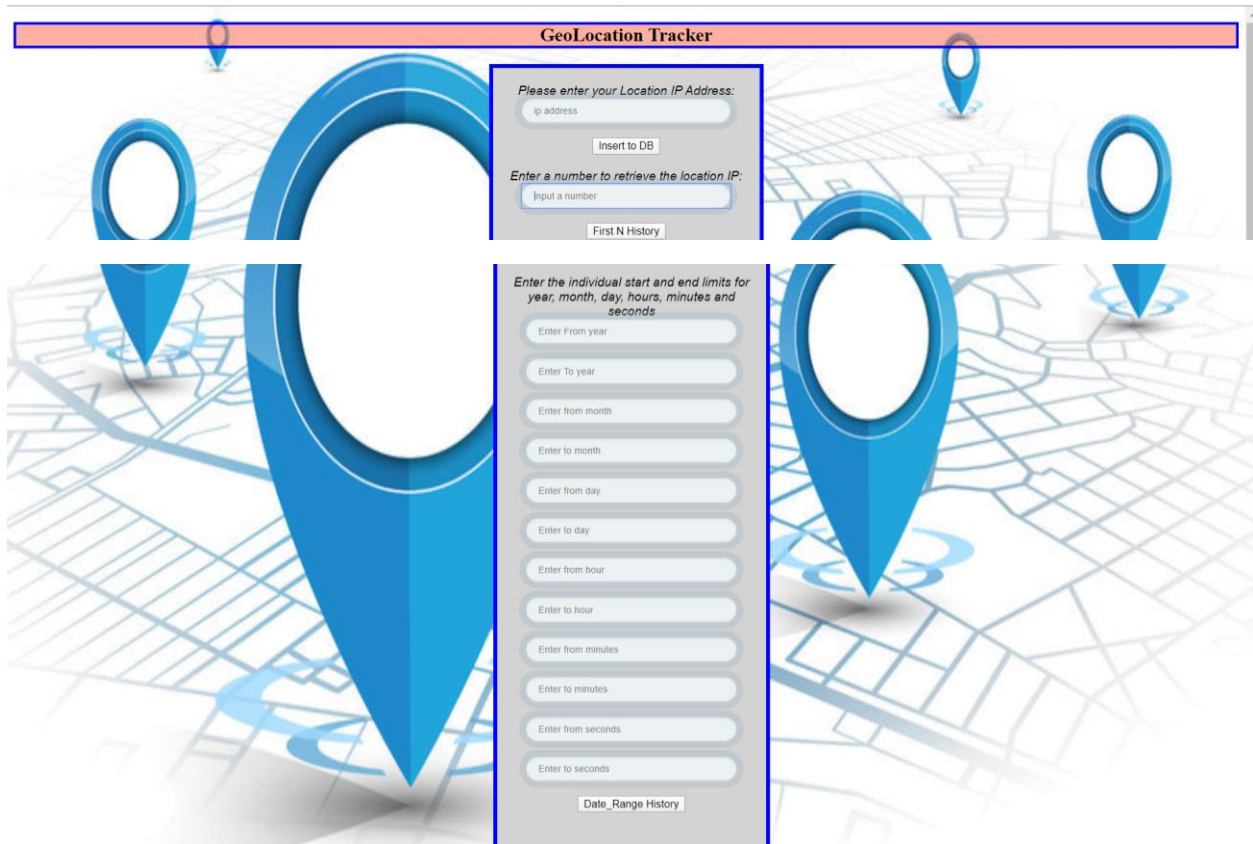


## Document for Geo Location Tracker

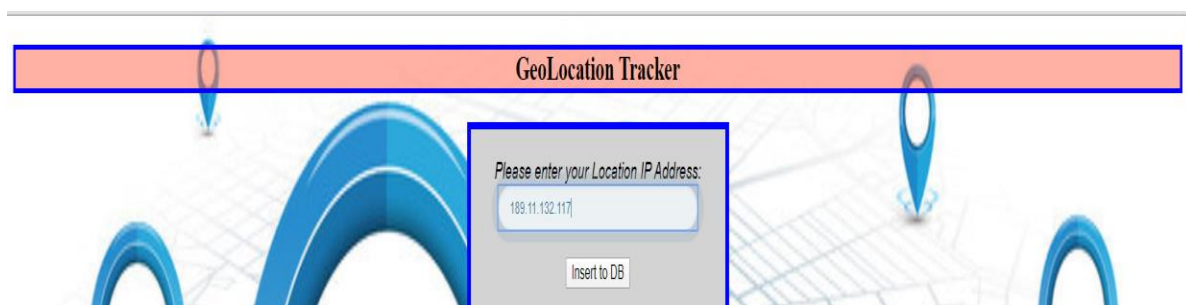
Initial homepage for the geo location tracker



The image shows the initial homepage of the GeoLocation Tracker application. The background features a stylized map with several large blue location pins. A central panel contains the following elements:

- A red header bar with the text "GeoLocation Tracker".
- A section titled "Please enter your Location IP Address:" with a text input field labeled "ip address" and an "Insert to DB" button.
- A section titled "Enter a number to retrieve the location IP:" with a text input field labeled "Input a number" and a "First N History" button.
- A section titled "Enter the individual start and end limits for year, month, day, hours, minutes and seconds" with multiple input fields for each unit (year, month, day, hour, minute, second) and a "Date\_Range History" button.

First, we will enter the location ip address of any city(189.11.132.117) and press the button "Insert to DB"



The image shows the same GeoLocation Tracker form, but with the IP address "189.11.132.117" entered in the "Please enter your Location IP Address:" field. The "Insert to DB" button is visible below the input field.

Another webpage is displayed which says that the ip address is being added to the database:

Let's have a look at the mongodb now to check whether the new entry has been added to the database or not. The new entry has been added now.

**mongoDB Atlas** All Clusters

CONTEXT  
Project 0

ATLAS  
Clusters  
Data Lake BETA

SECURITY  
Database Access  
Network Access  
Advanced

PROJECT  
Access Management  
Activity Feed  
Alerts 0  
Integrations  
Settings

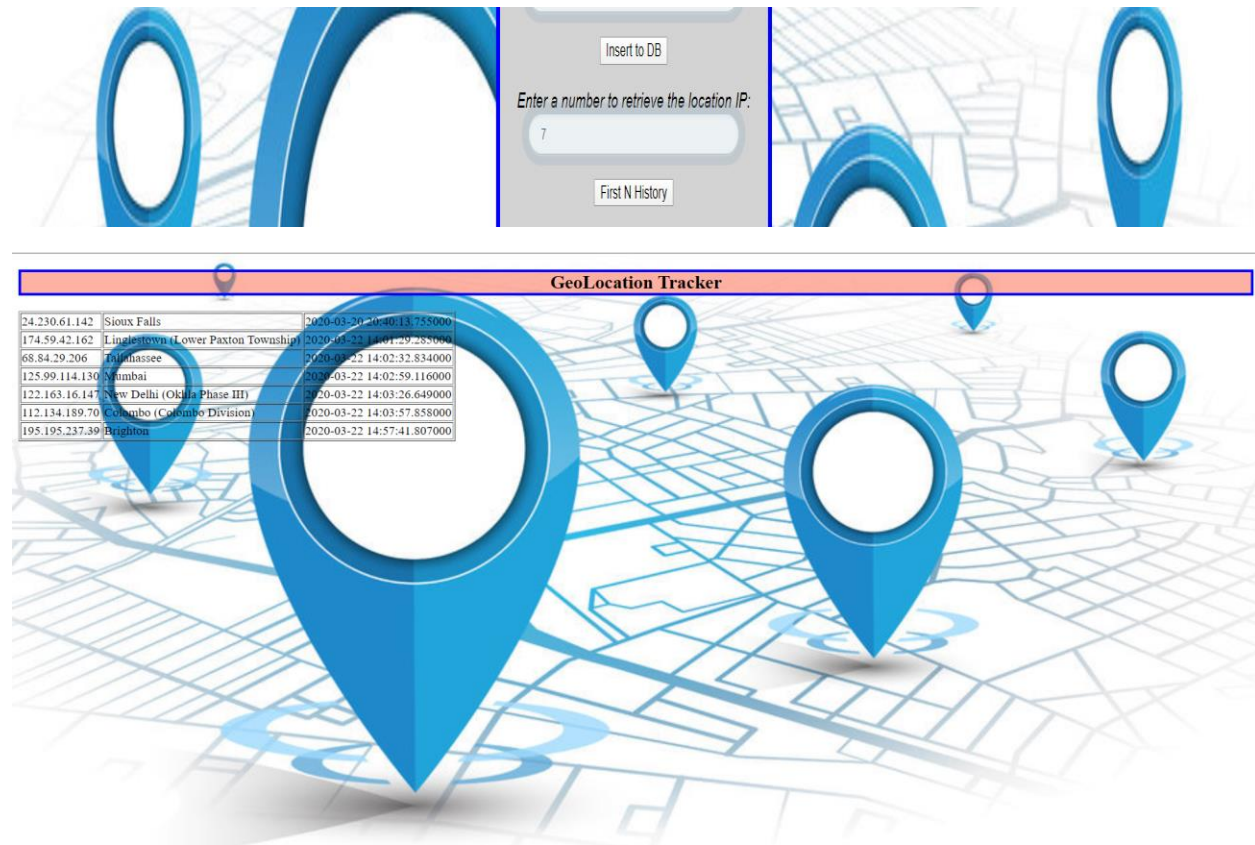
`_id: ObjectId("5e790c2605ca3c1044672588")`  
`IP_Address: "195.9.20.2"`  
`City: "Moscow"`  
`time: 2020-03-23T15:21:10.386+00:00`

`_id: ObjectId("5e7930bf05ca3c1bc0af724c")`  
`IP_Address: "213.143.46.18"`  
`City: "Lavapiés"`  
`time: 2020-03-23T17:57:19.363+00:00`

`_id: ObjectId("5e79792505ca3c1b48efb441")`  
`IP_Address: "189.11.132.117"`  
`City: "São Paulo"`  
`time: 2020-03-23T23:06:13.703+00:00`

System Status: **All Good** Last Login: 72.79.67.168  
©2020 MongoDB, Inc. Status Terms Privacy Atlas Blog Contact Sales

First N location history button is shown to display the last 7 ip address into the database



If we want to specifically trace a particular ip address and its city according to the timestamp then we have to input the start year, end year, start month, end month, start day, end day, start hour, end hour, start minutes, end minutes, start seconds and end seconds. Basically, I am applying a filter based on the actual timestamp the entry has been added into the database.

So, for the current example I have put the following values:

Start year: 2020

End year: 2020

Start month: 03(March)

End Month: 03(March)

Start Day: 20(20<sup>th</sup> March)

End Day: 22(22<sup>nd</sup> March)

Start Hour: 20(i.e. 10pm on 20<sup>th</sup> March)

End Hour: 14(i.e. 2pm on 22<sup>nd</sup> March)

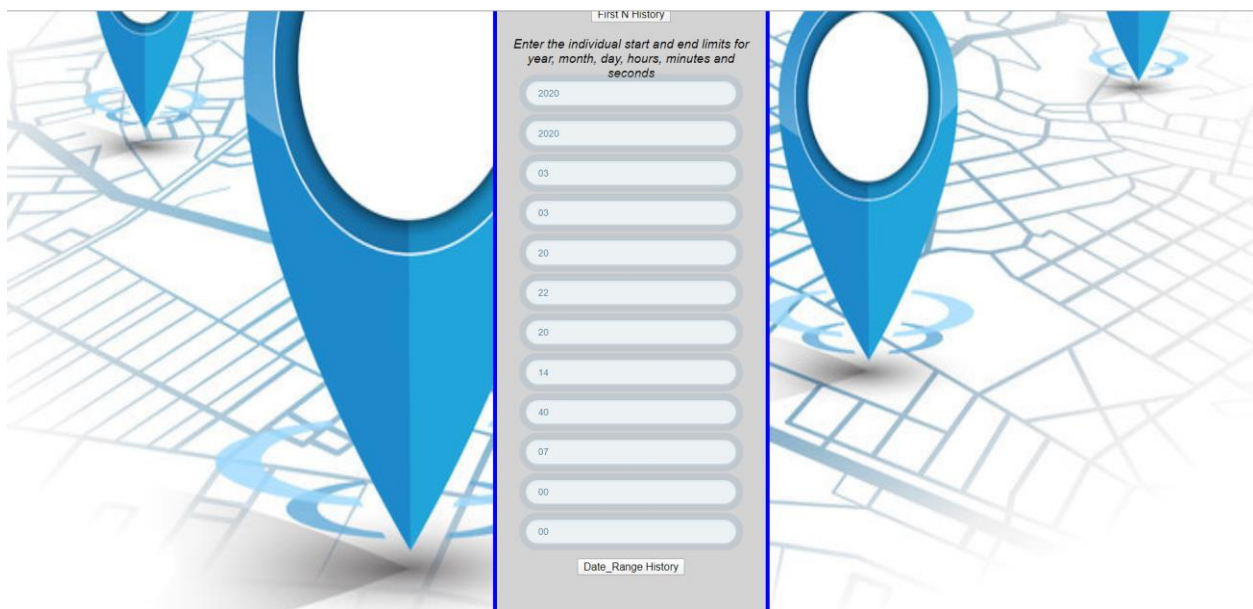
Start Minutes: 40(10:40pm on 20<sup>th</sup> March)

End Minutes: 07(2:07pm on 22<sup>nd</sup> March)

Start Seconds: 00(i.e. start on 20<sup>th</sup> March 10:40:00pm)

End Seconds:00(i.e. end on 22<sup>nd</sup> March 2:07:00pm)

I have given the entire timestamp for the query to be executed and now I will click on the button to get the output screen.



The screenshot shows a web interface for selecting a date range. It features a central vertical column of input fields, each with a blue location pin icon to its left. The background is a light gray map with a grid of streets. The interface includes a 'First N History' button at the top, a 'Date\_Range History' button at the bottom, and a text prompt: 'Enter the individual start and end limits for year, month, day, hours, minutes and seconds'. The input fields are arranged in two columns, with the first column containing fields for year, month, day, hour, minute, and second, and the second column containing fields for year, month, day, hour, minute, and second. The values entered in the fields are: 2020, 03, 20, 22, 20, 14, 40, 07, 00, 00.

Field	Value
Year (Start)	2020
Month (Start)	03
Day (Start)	20
Hour (Start)	22
Minute (Start)	20
Second (Start)	14
Year (End)	40
Month (End)	07
Day (End)	00
Hour (End)	00
Minute (End)	
Second (End)	



This is the python shell output which matches the criteria of

**Start Time: 20<sup>th</sup> March 10:40:00pm and**

**End time: 22<sup>nd</sup> March 2:07:00pm**

```
{'_id': ObjectId('5e75626d05ca3c5108db74da'), 'IP_Address': '24.230.61.142', 'City': 'Sioux Falls', 'time': datetime.datetime(2020, 3, 20, 20, 40, 13, 755000)}
```