Talking Network

Hardik Khurana, Vinayakk Garg

GOAL

- The aim is to create a utility webapp to facilitate studying and analysation of pipeline diagrams.

- We’re building a voice based application to receive feedback about pipeline parameters.

Process:

1. Fetching of data and data preprocessing.
2. Conversion of data to appropriate format.
3. Building the layout of webapp.
4. Projection of the data on the webmap.
5. Connection of events to Google Voice API.
6. Deploying the app on the local server.

Fetching of data and data preprocessing:

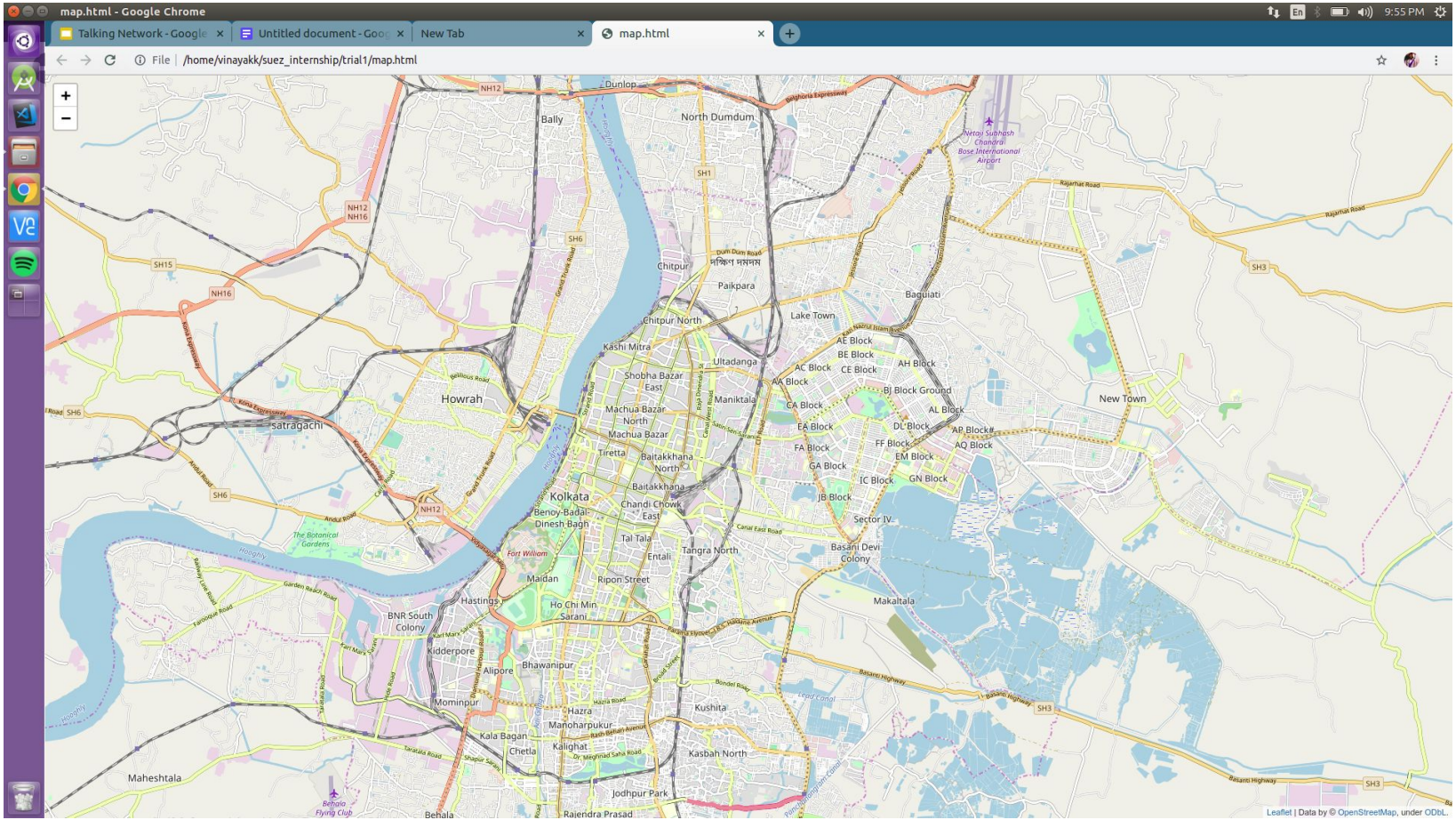
The data provided to us was related to the Existing Pipeline network in one of the wards in Kolkata.

- We were provided with the ECW file for the map image.
- We were also given the shape files containing details about the existing pipeline network.

Image Files (Problem and Solution)

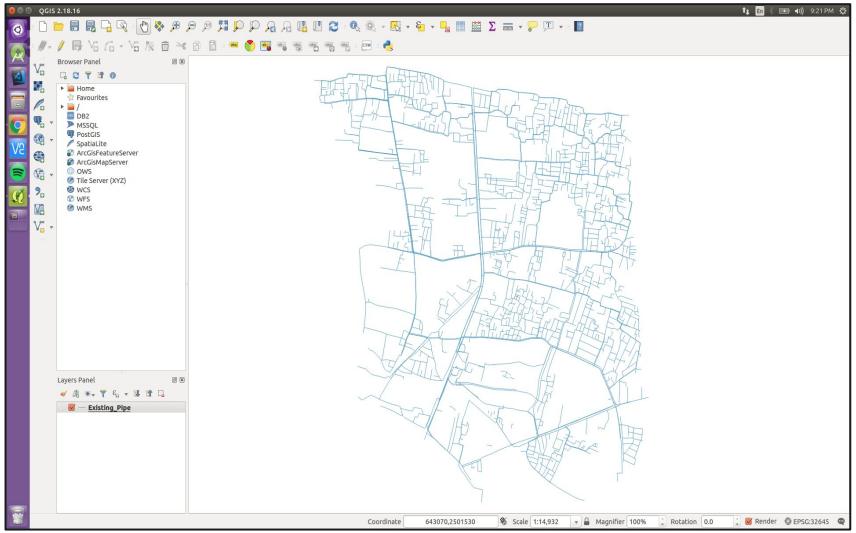
- After extensive research and trial on multiple softwares we concluded that ecw files can be accessed only via the ArcGIS software which was paid and hence presently out of our scope.
- Conversion from ecw to geoJSON was also not possible.
- Since the file could not be accessed, we discussed with the GIS team and looked online for alternative solutions to come across a JavaScript library called Leaflet.JS
- We used geojson.io to determine the central coordinates of the required ward.

Result:


map.html - Google Chrome
Talking Network - Google
Untitled document - Goo
New Tab
map.html
File | /home/vinayakk/suez_internship/trial1/map.html
Leaflet | Data by © OpenStreetMap, under ODbL.

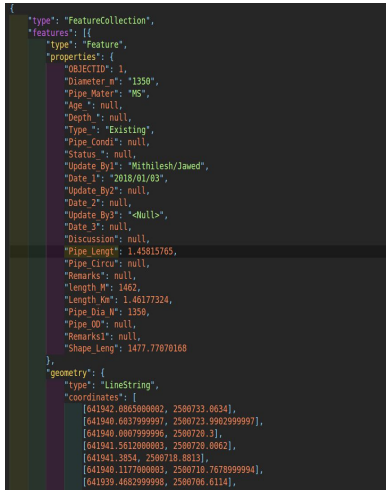
Shape Files (Problem and Solution)

- The shape file contains the actual details about the pipeline network. It is central to our entire project. We were provided the shape files in multiple formats.
- The first step required them to be converted to the GeoJSON format which was easily done using an online tool. However, we then encountered an major problem.
- The coordinates provided didn't make much sense to us and hence we discussed the same with the GIS team to conclude that the coordinates follow the EPSG 32645 (Spatial Reference) format. Generally webmaps are plotted using the EPSG 4326(Global Coordinate System) format (used by Google Maps etc.)
- The first solution we came across was to use a FOSS software called QGIS.



Shape Files (Problem and Solution)

- However, the software didn't really manage to accomplish the task and hence we had to think about another solution for the same.
- We researched on the internet to find the mathematical formula for conversion between EPSG 32645 and EPSG 4326 formats.
- We then developed a Python Script to read the given data and convert it to required EPSG 4326 format. The script also filter out the data required to be provided as the feedback.

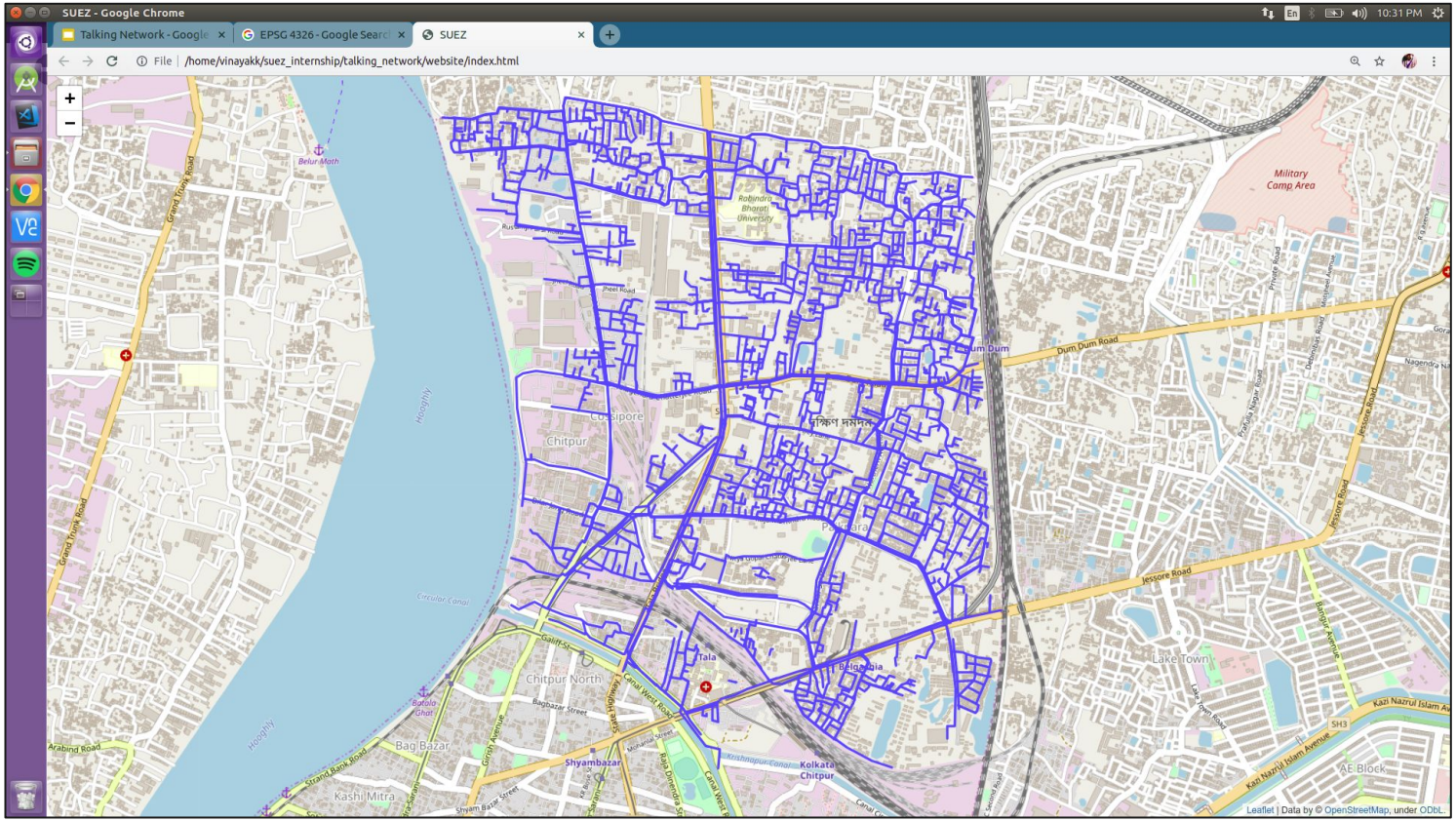


Original GeoJSON File

Converted GeoJSON File

Result:

Once we had all the data in the required format, we then projected it onto the webmap.

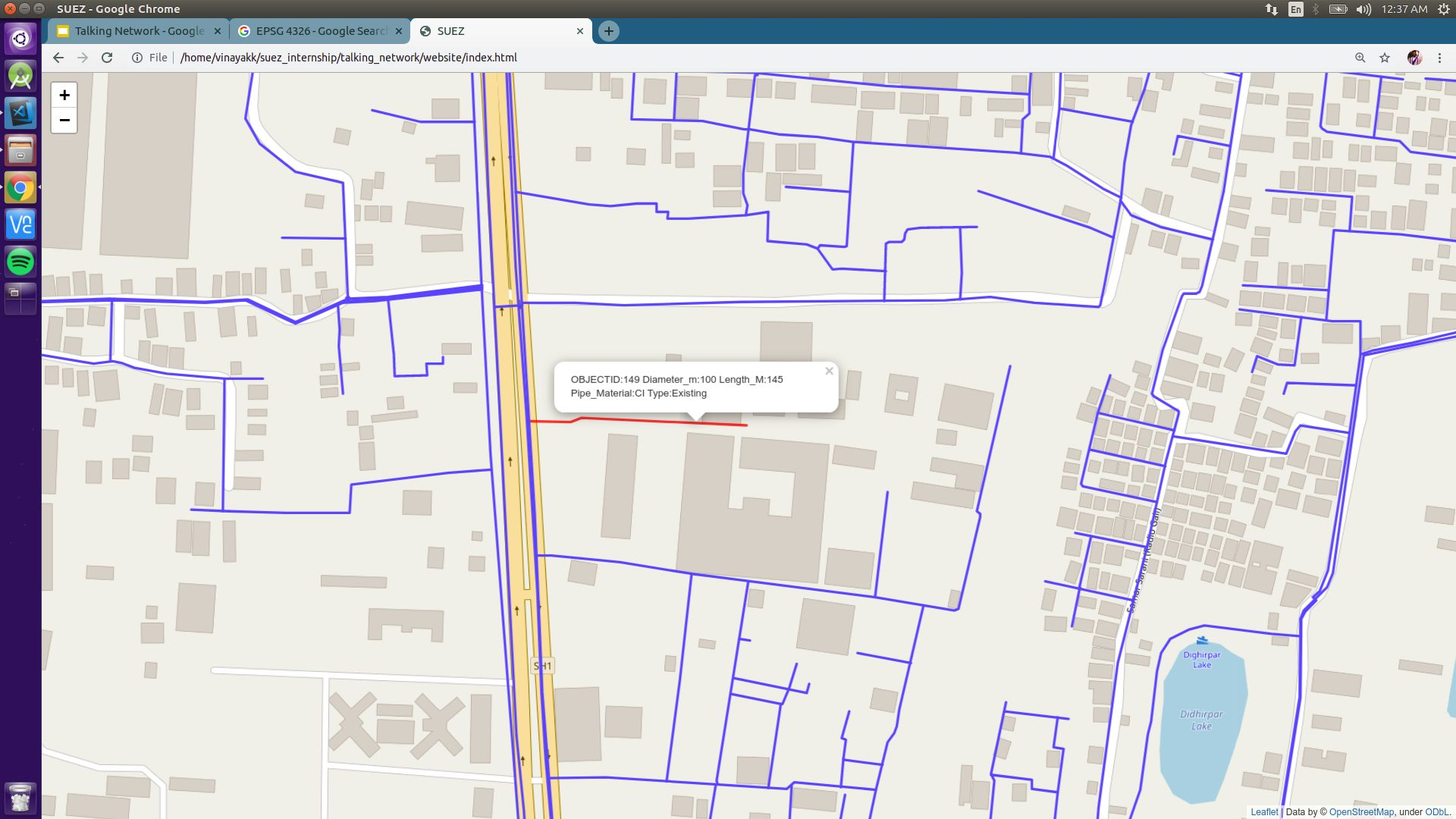


Projecting Required Data on Map:

The next part of the project involves receiving voice feedback on selection of a particular pipeline. To develop that, we initially divided the problem trivially as:

1. Selecting and highlighting the pipeline.
2. Popup displaying the required data.

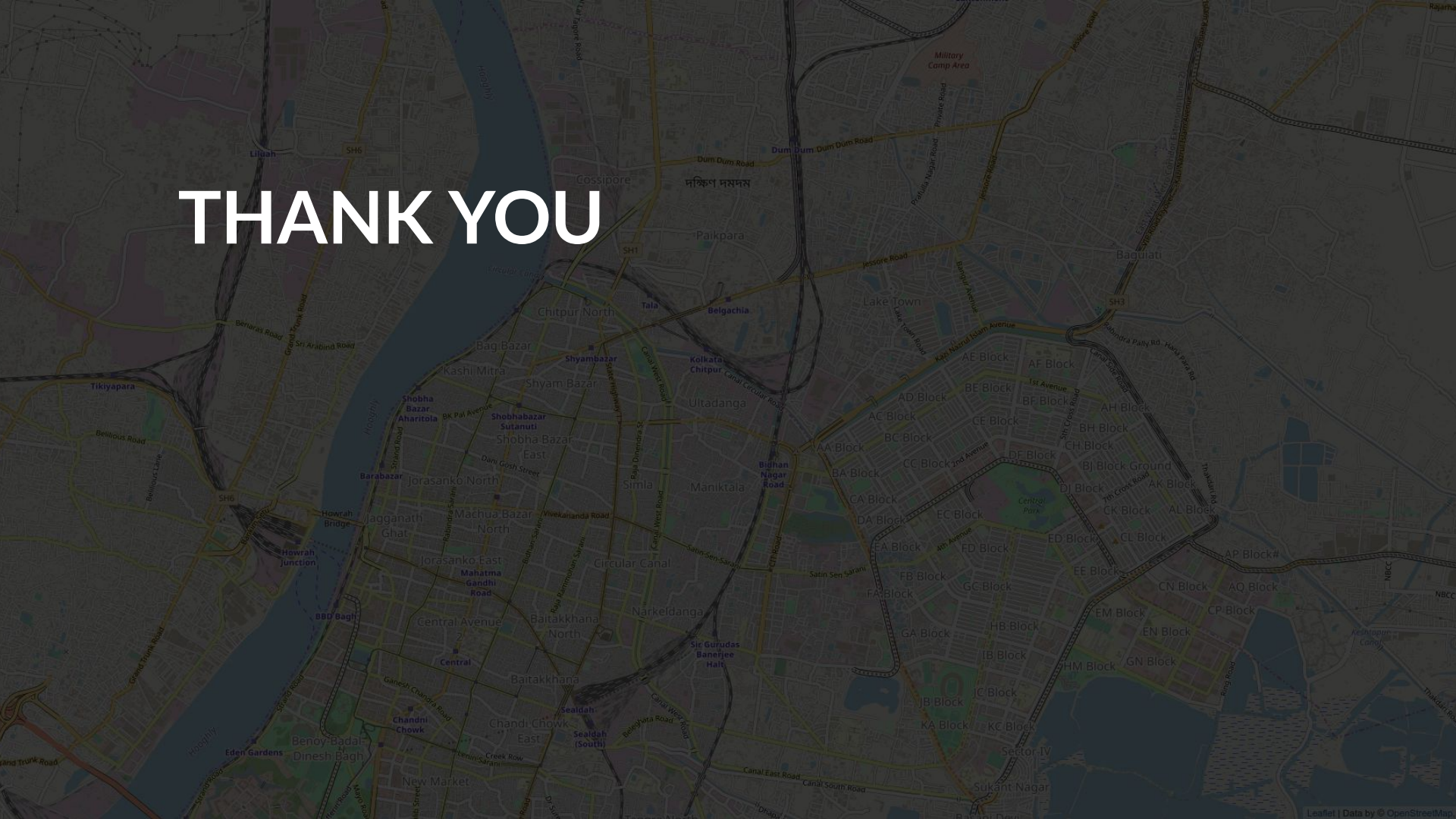
These approaches straight up required good knowledge about web development which we developed over a day or two and built upon the presently working prototype...


SUEZ - Google Chrome
12:37 AM
Talking Network - Google
EPSG 4326 - Google Searcl
SUEZ
File | /home/vinayakk/suez_internship/talking_network/website/index.html
OBJECTID:149 Diameter_m:100 Length_M:145
Pipe_Material:CI Type:Existing
SH1
Dighirpar Lake
Dighirpar Lake
Leaflet | Data by © OpenStreetMap, under ODbL.

Future Goals:

We plan to work on:

- Connecting the app to Google Voice API to receive a audio feedback.
- Improving the WebUI for visual appeal.
- Scaling the app for larger data.
- Deploying the webMap on the local server.


THANK YOU