WEEK1-

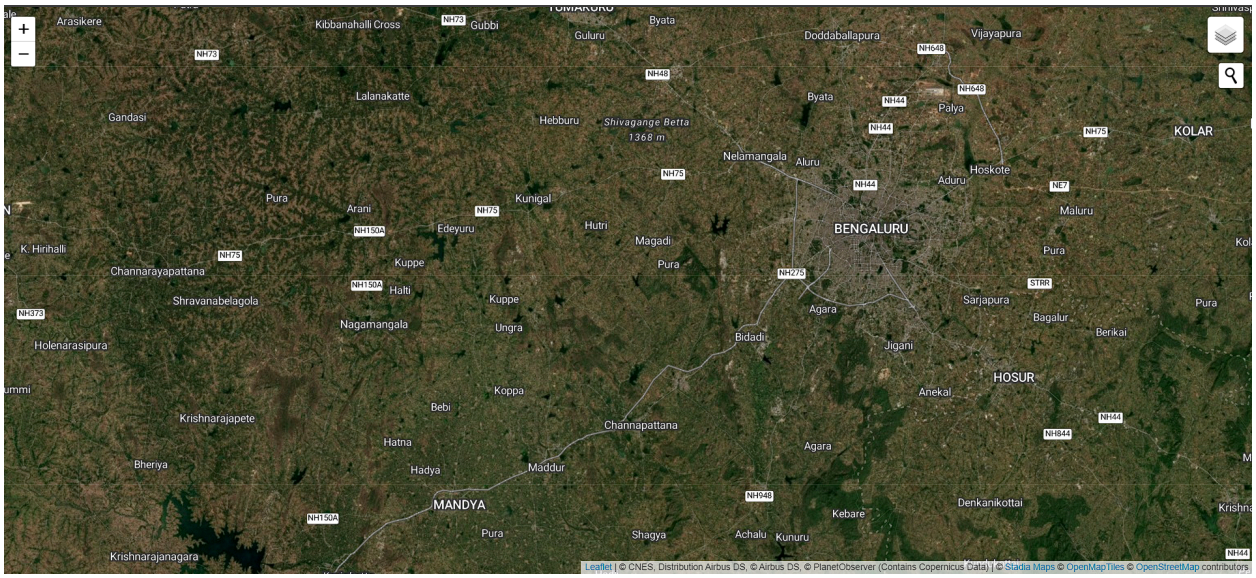
- Problem Statement divided into three parts
    1. NLP
    2. Creating a frontend showing the maps
    3. Speech to text
- In the middle of searching the dataset required for showing the maps. Came across different map datasets like OSM, NASAWorldView and Copernicus.We are still deciding as to how the map layers can be implemented on each other after doing some more research regarding the same.
- We also chose the voice dataset on which our model will be trained. The link for the same is : https://commonvoice.mozilla.org/en/datasets.
- We have finalized the Leaflet library for the GIS.
- We will be exploring more on the deep learning side and hence will start with reading and exploring more deep learning neural networks.
- After that is done we plan to split the tasks among ourselves with each of us focusing on the subtopics mentioned above.

WEEK 2-

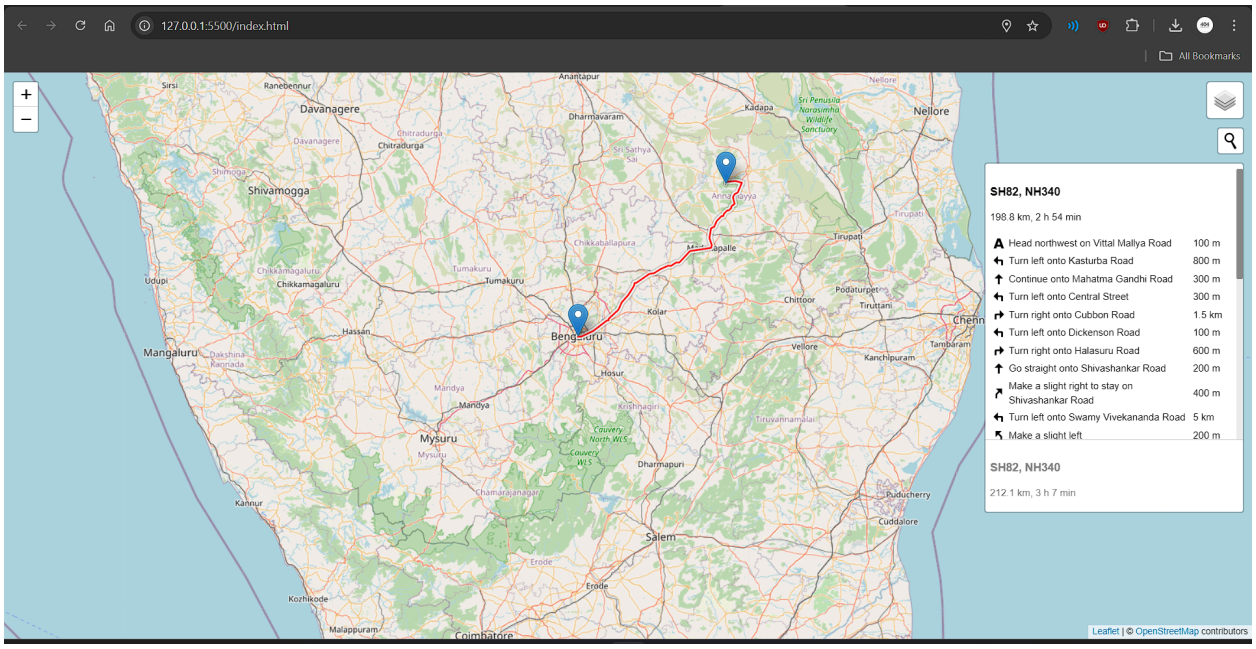
We learned the basics of deep learning, nlp and leaflet.js.

WEEK 3-

We created a basic website with multiple layers of maps and overlays taken from https://leaflet-extras.github.io/leaflet-providers/preview/



Added a routing feature:

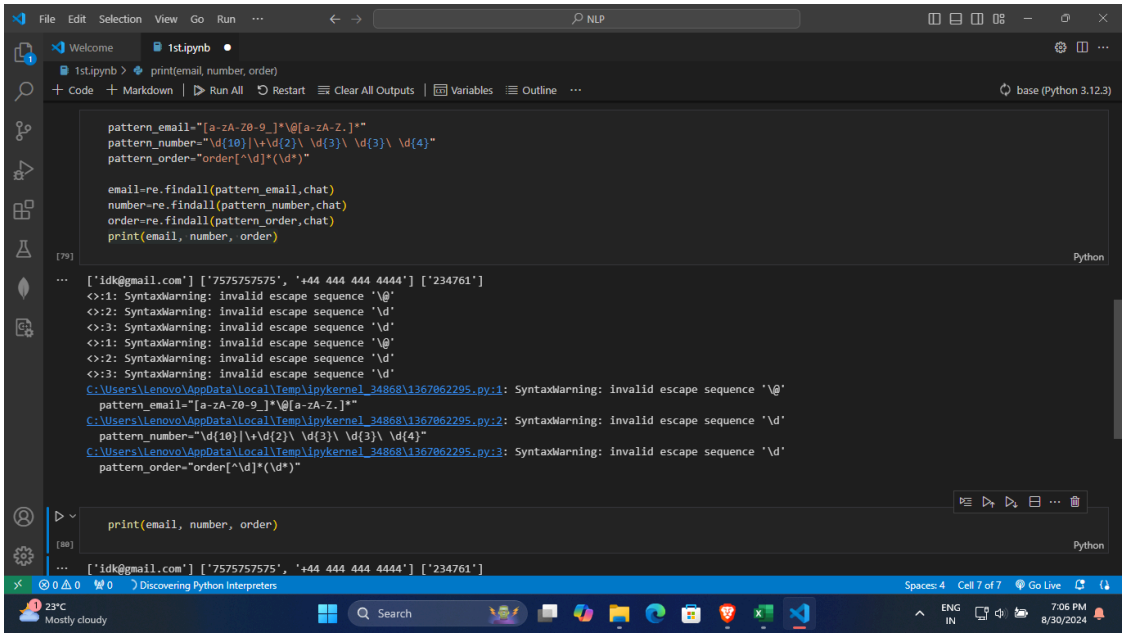


We planned to add features like getting the current location of the user.

We are in the process of learning nlp and deep learning.

Week 4-

Started with nlp
Learnt pattern recognition in texts and how to extract the keywords



Usage of re module.

WEEK 5-

Did some NLP practice and looked into some models for speech recognition.

https://youtu.be/ENLEjGozrio?si=TzV7zgM6KziIY6ni Video on steps involved in NLP from basics to text preprocessing.

https://www.youtube.com/watch?v=2kSPbH4jWME&t=16s : Video on Real-time speech recognition using python.

WEEK- Don't know

NLP(Dev and Richa)-

UNDERSTANDING OF BERT:-

We looked for many models, and chose BERT(a pre-trained model for text summarization). We looked into how bert works.

We imported two models using apis.
one for preprocessing
And the other for encoding.

Preprocessing

It divides the sentences into tokens and stores them into a dictionary.

Encoders

It uses embedding to create a dense layer of continuous vectors and stores them in a dictionary. Basically creating a new array for each token with 728 columns because we are using bert base.

IMPLEMENTATION

We created a dictionary called “intent_map” to map words like ‘look for’ and ‘find’ to ‘zoom’. Then we took 2 sentences preprocessed them and tokenized them and then printed the tokens.

We then tried a manual extraction of important keywords(LIke the command zoom and the location delhi).

We loaded a pre-trained bert based ner(named entity recognition) pipeline and created an object for it with grouped entities true that actually groups the tokens together to form actual words. It takes in sentences and classifies the ner tokens into different entity groups like LOC(location), PER(person) etc. and gives a confidence score which is the sureness of its classification of the particular word. Classification is made smartly according to the context of the sentence.