A REPORT ON:

COVID PATIENT ALERT SYSTEM

Presented By:

GROUP 4

Group Members:

E Swarup Kumar

Sanjog Samuel Samantaray

Ashutosh Karmakar

Swastik Kanhar

AVS Chandrasekhar

Problem Statement:

In a colony of Khurda District, four close friends are staying. There name's are Adi, Mohit, Rahul and Sumit. Once lockdown over, they started roaming out without mask and not maintaining social distancing with other people. Suddenly after a few days, they started feeling weak and little fever. So they have decided to make IOT project in which they can record their body temperature data in Fahrenheit four time a day and save it in JSON format on the cloud. (Download data: https://github.com/iambhatter/JSON-Data.git) They have designed an IoT based system on Node-Red dashboard on AWS EC2 Instance, where dashboard will display data of each four members individually day and time-wise on the graph. Node-Red has to pick data from JSON file as per date only. Now they have designed IoT MQTT Broker on AWS IOT Core so that Node-Red can publish data on IoT Core, and they also set up Email service with AWS SNS, and also place on condition when body temperature will rise up and more than 100 F, then remaining three-person will receive an email. They have also designed Android App on MIT App Inventor to track their location and save it in a local database for future record. You all have to design the same project in the assigned group. And submit the documentation of the project in .pdf file with description and project output screenshot of each screen.

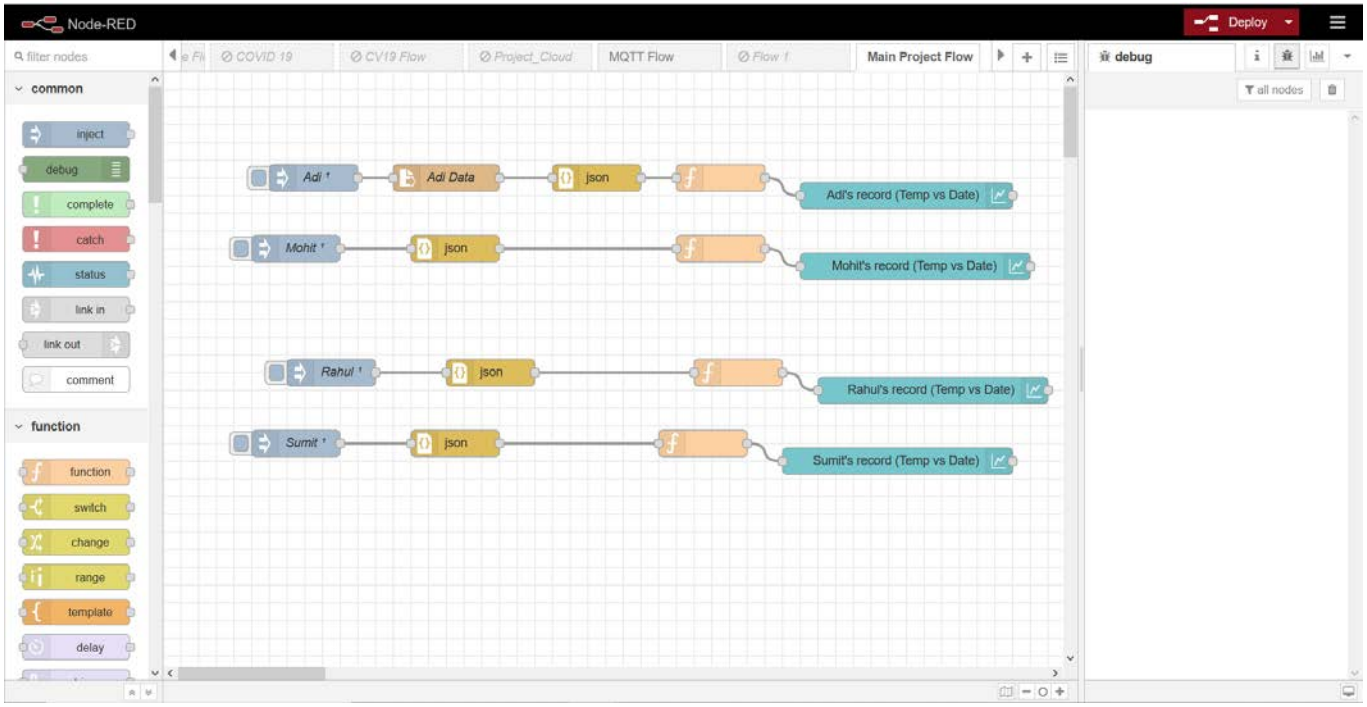
Solution

We differentiated our problem statement into 3 parts to get a better understanding and to tackle it efficiently. They are:

I. The first part is to plot the Temperature vs Date of 4 individual in node red.
II. The second part was to get the required values in AWS and to send emails through SNS service based on the problem statement.
III. And at the last part but not least, it was to get an app designed to track their locations using MIT App Inventor and to store those locations in a local database

Part-1:

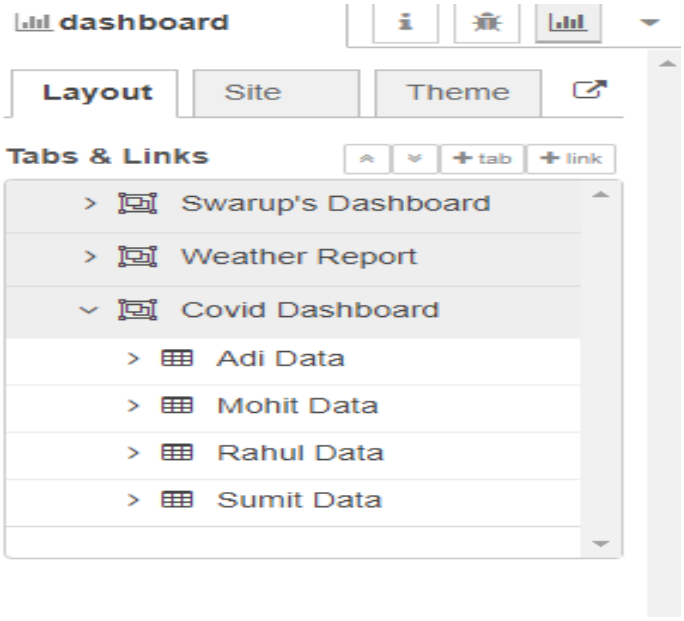
This part includes the designing of **Temperature vs Date** on Node-Red dashboard on AWS EC2 Instance.



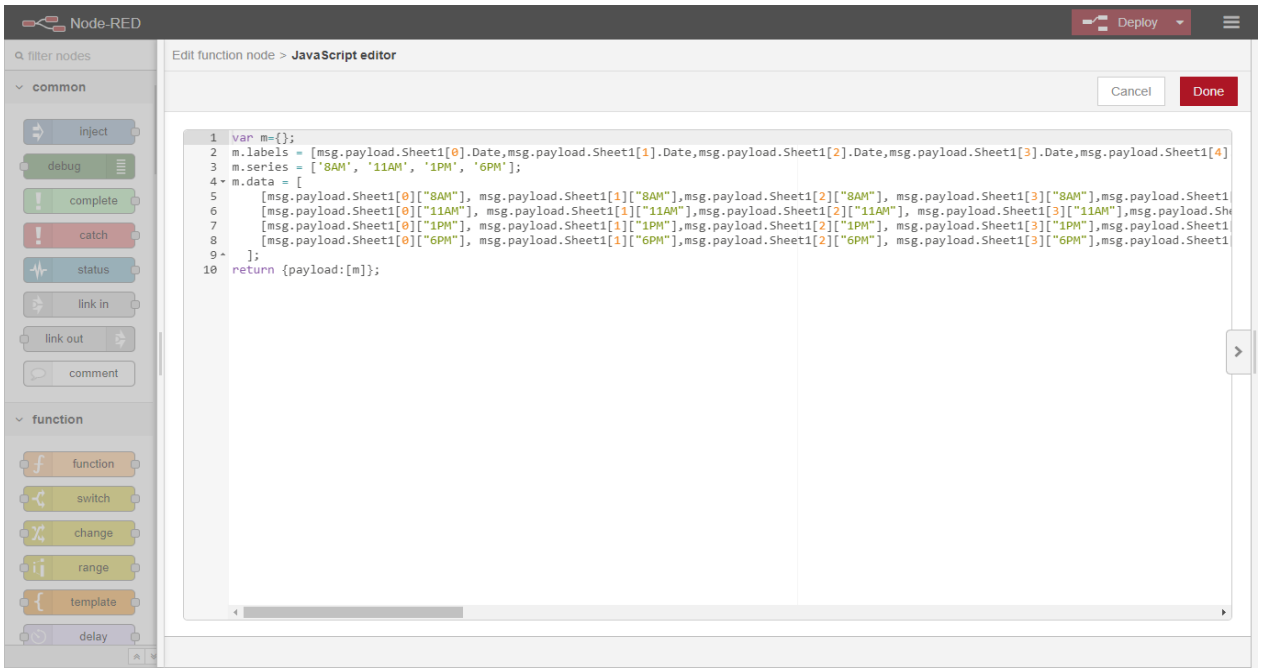
The Above picture is what the flow looks like for the part 1. 4 different inject nodes were used for 4 individual graph plots.

When the inject node is injected it fetches the file and pushes it to the json node to convert it into Object notation.

Similarly, for all others, we have implemented the same.



This the configuration of the Dashboard.



This is the inside of the function node

.

- **Label**: It refers to the values on the X-Axis. Here, in the X-Axis , it should be the date as per the data given in json file. So we have fetched the date data from the json.
- **Series:** It refers to the time of each day, i.e the no. of lines on the graph. Here, it is 8AM,11AM,1PM,6PM.
- **Data**: It refers to the main data i.e the temperature. Column wise data refers to the temperature of a particular day, like 20 no. of column for 20 days.

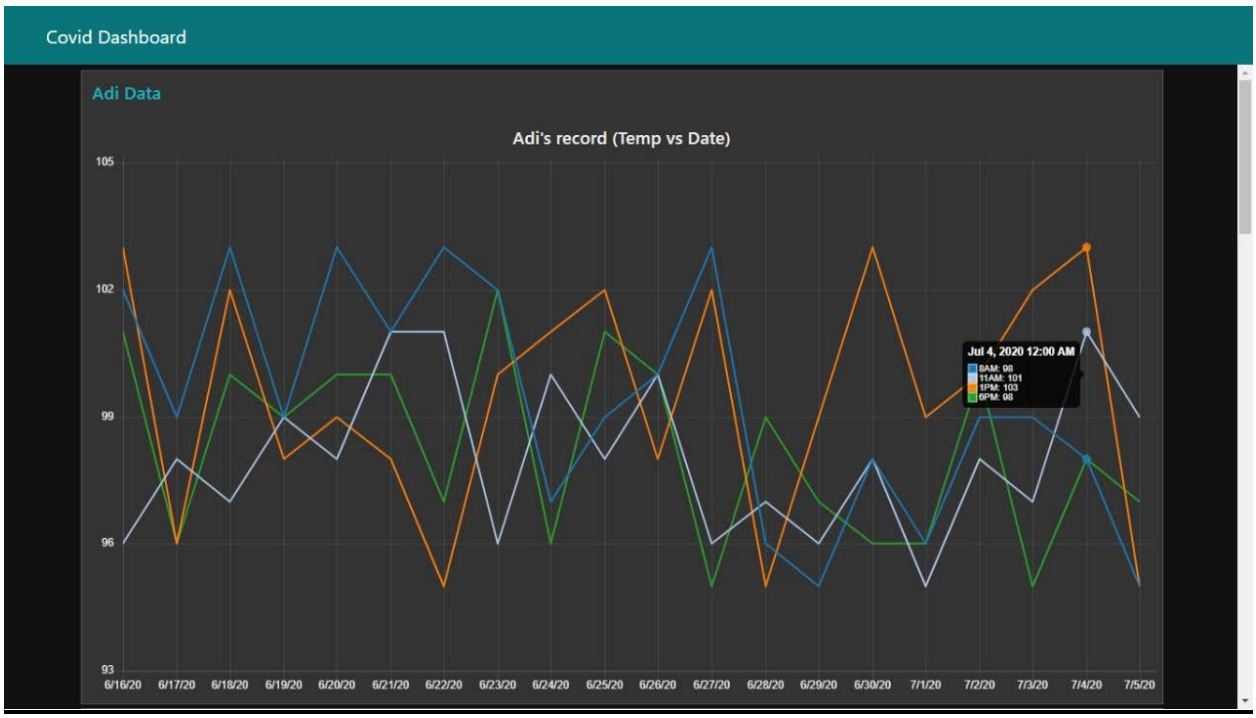
Plotting of the Graph

X-axis: - Date (from 16/6/20 to 5/7/20)

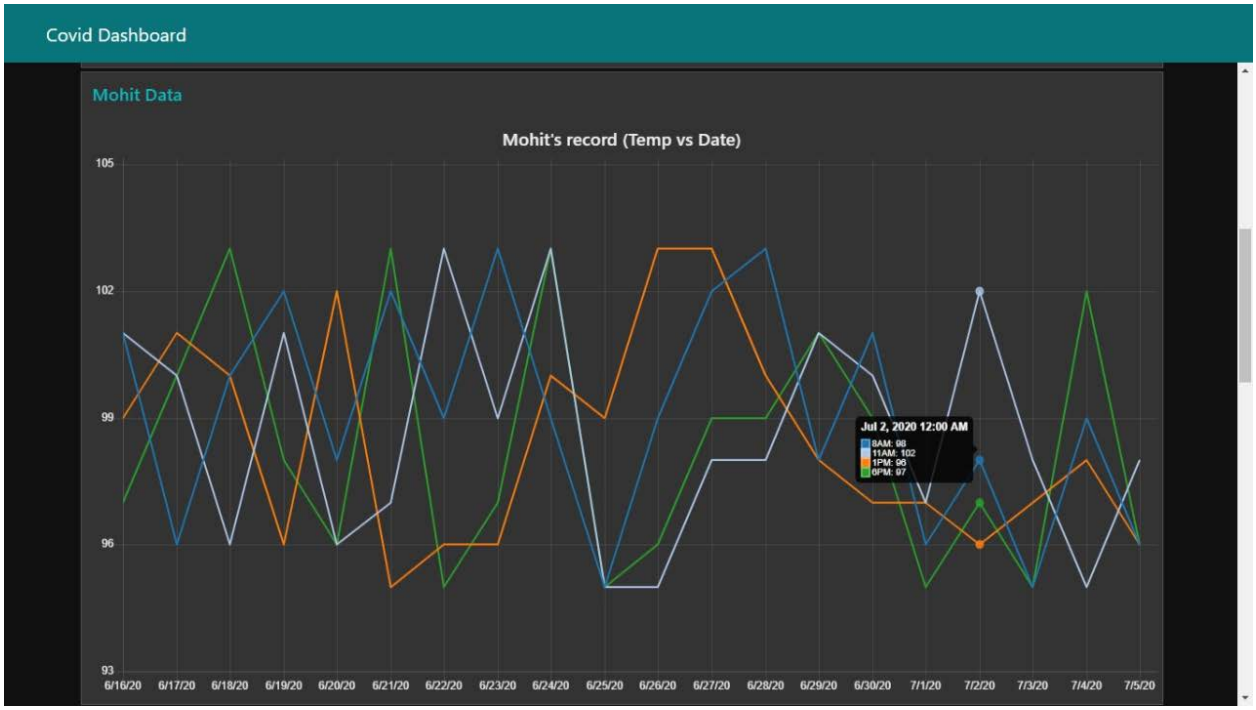
Y-axis: -Temperature Axis (Ranging from 93 to 105) in Fahrenheit.

As you can see from the graph,

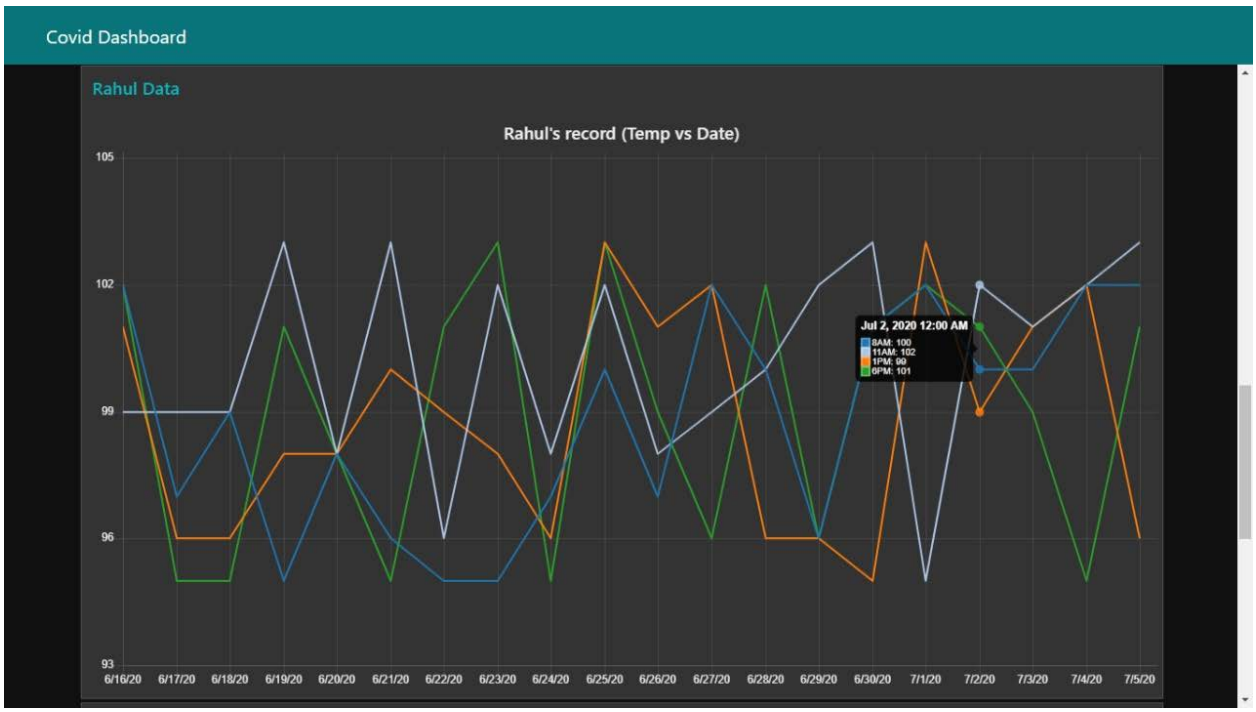
I. Dark blue line refers to the line of data at 8AM.
II. Light blue line refers the line of data at 11AM.
III. Orange line refers the line of data at 1PM.
IV. Green line refers the line of data at 6PM.



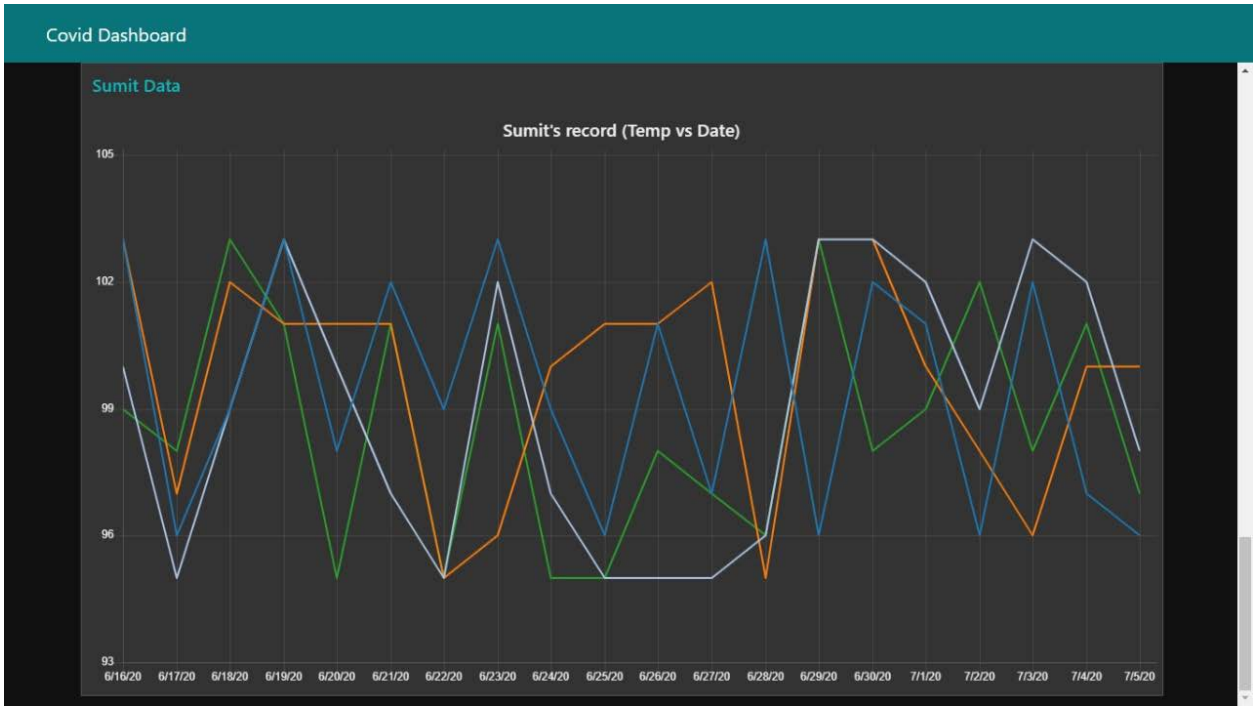
The above graph is the graph of Adi from the given Adi.json file.



The above graph is the graph of Mohit from the given Mohit.json file.



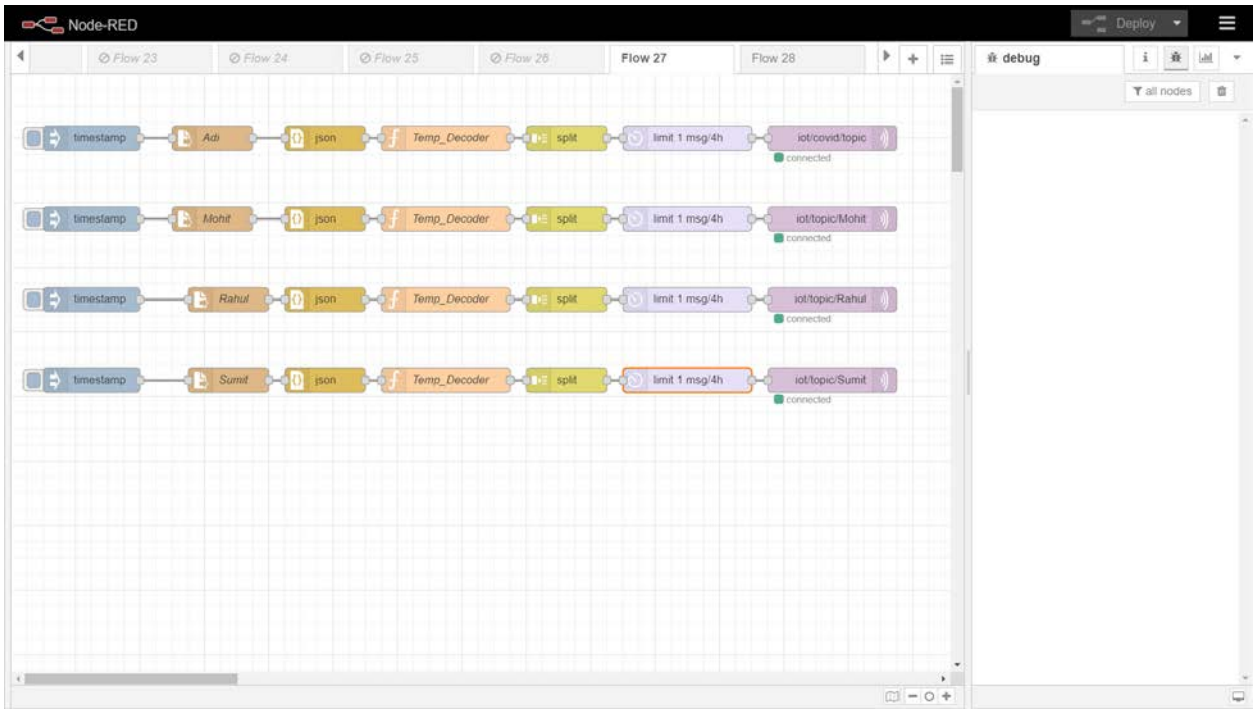
The above graph is the graph of Rahul from the given Rahul.json file.



The above graph is the graph of Sumit from the given Sumit.json file.

Part-2:

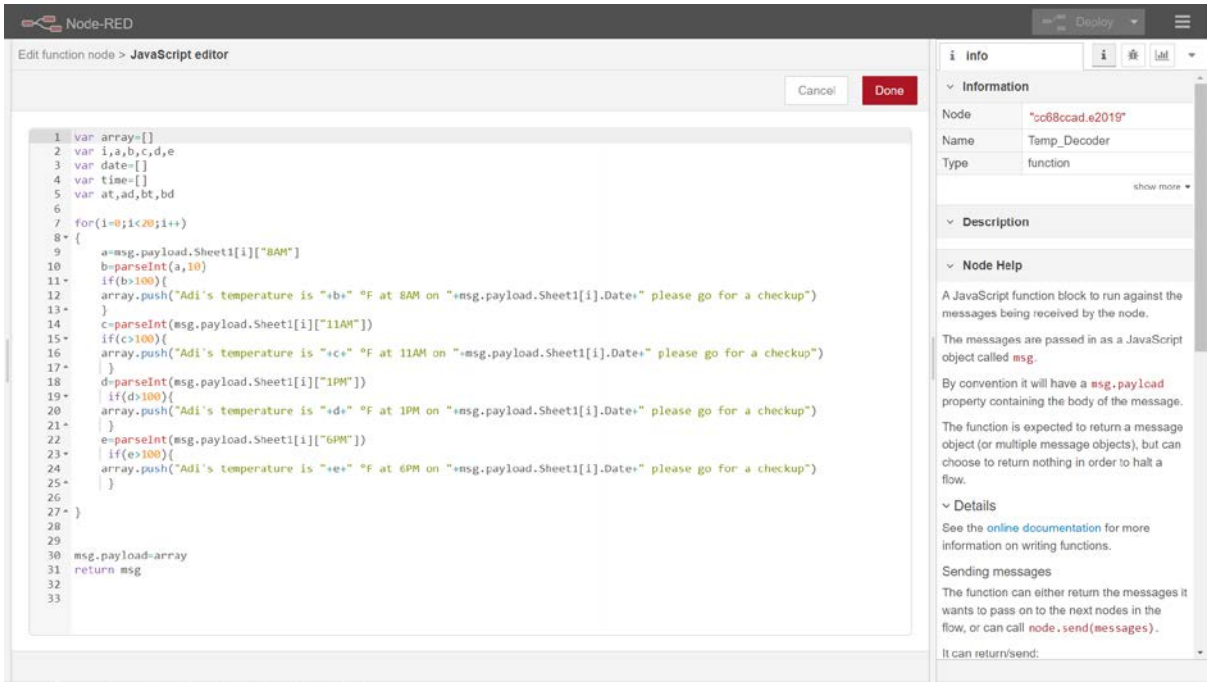
This part includes the procedure to get the required values in AWS and to send emails through SNS service.



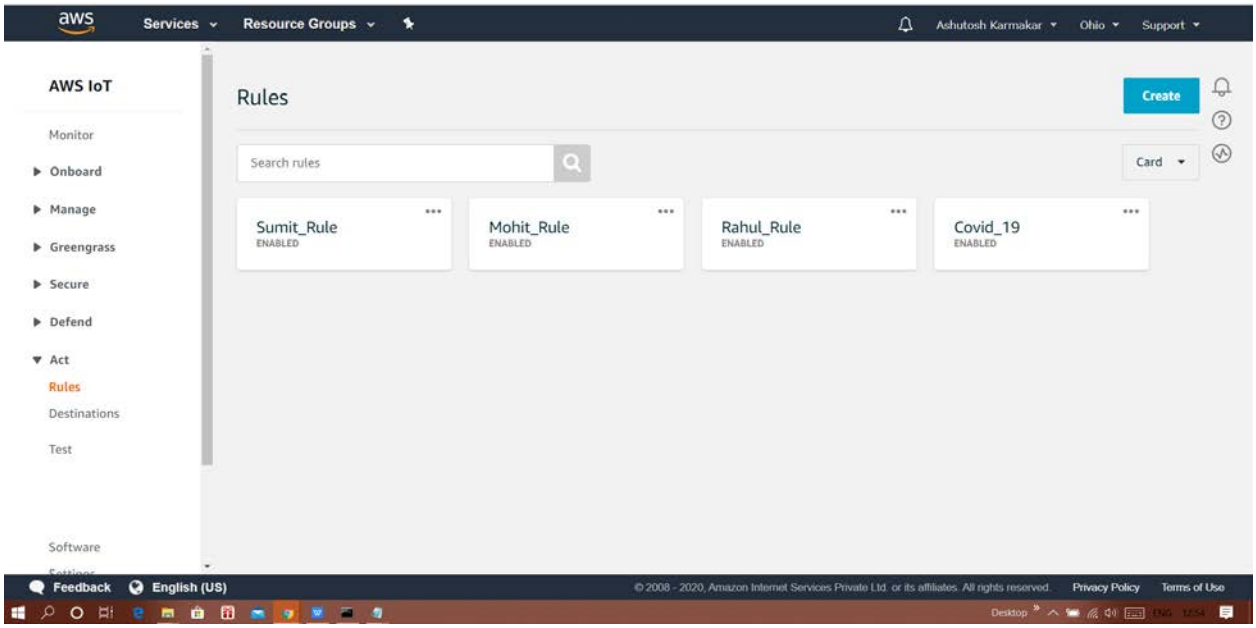
Here we created 4 unique brokers for each individual and connected them to unique topics, Rules and Roles

Nodes used:

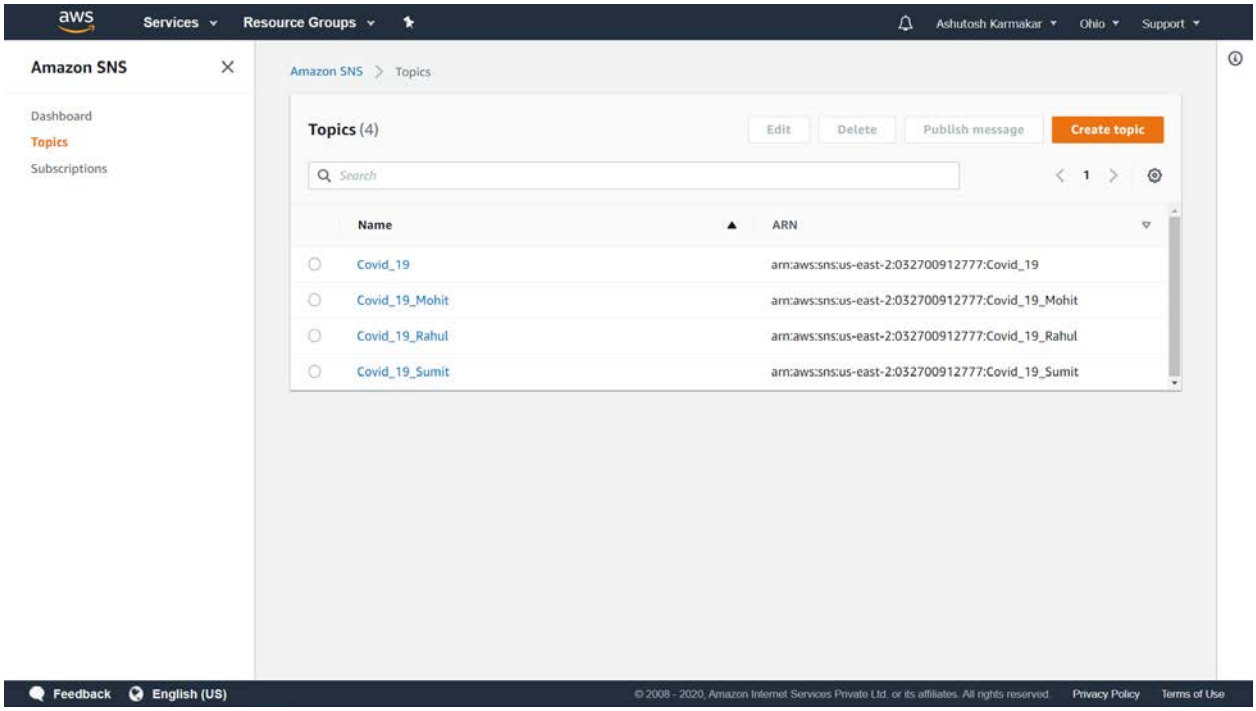
- **Time stamp**: To trigger the file node initially.
- **File in**: To connect with the file
- **JSON**: To convert the file output into JSON Object format
- **Function**: To decode the Object data and get the desired output in an array
- **Split**: To break the array into single values
- **Delay**: To send a single value in every 4-hour interval
- **MQTT Out**: To send the ouput values to AWS broker



This is the inside of the function node



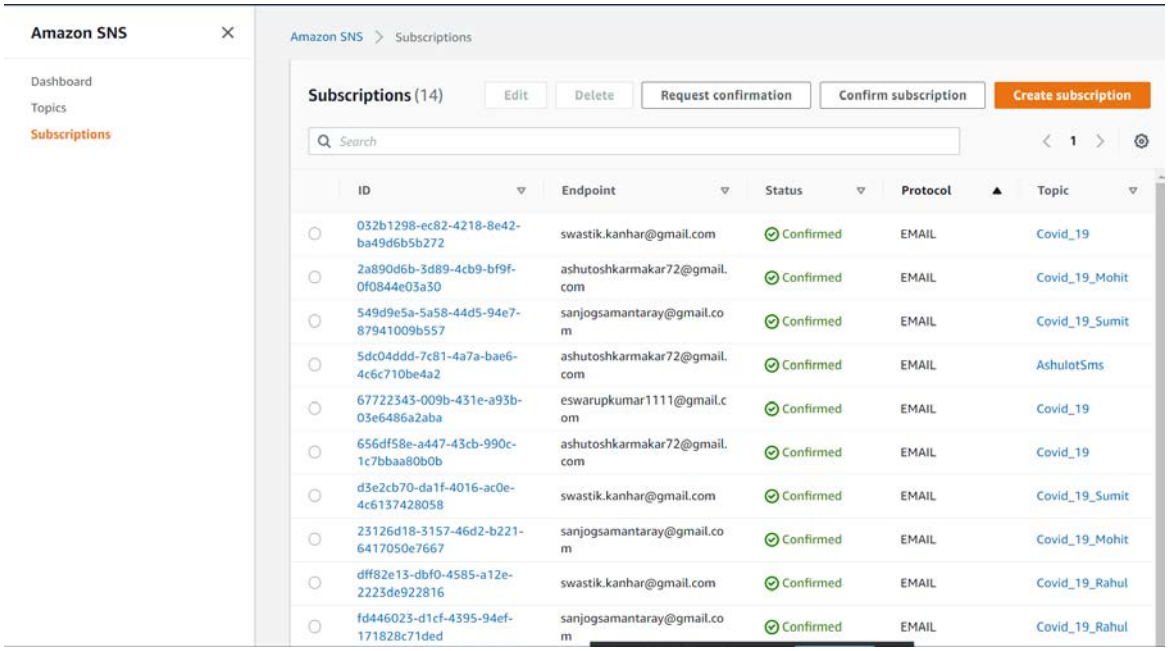
Here, we created 4 rules for each of the individuals. In each rule we assigned a role to enable a SNS email service which was sent to the other 3 individuals every time the temperature of the publishing individual raised over 100°F.

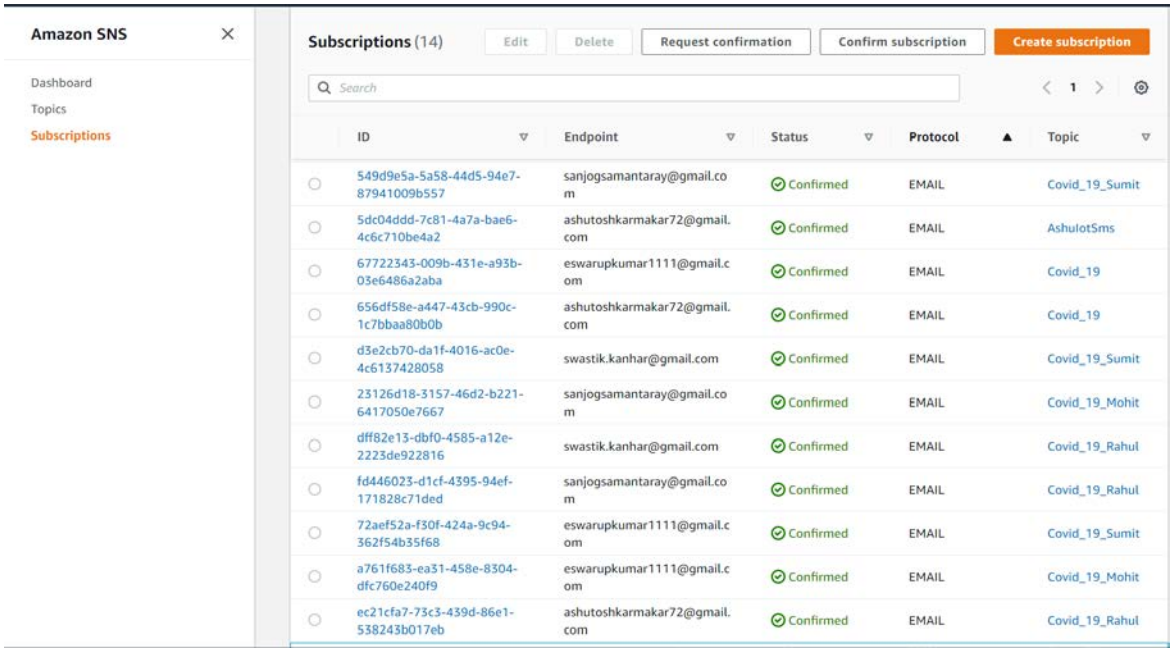


Every topic was published by a single individual and was subscribed by the rest of the individuals.

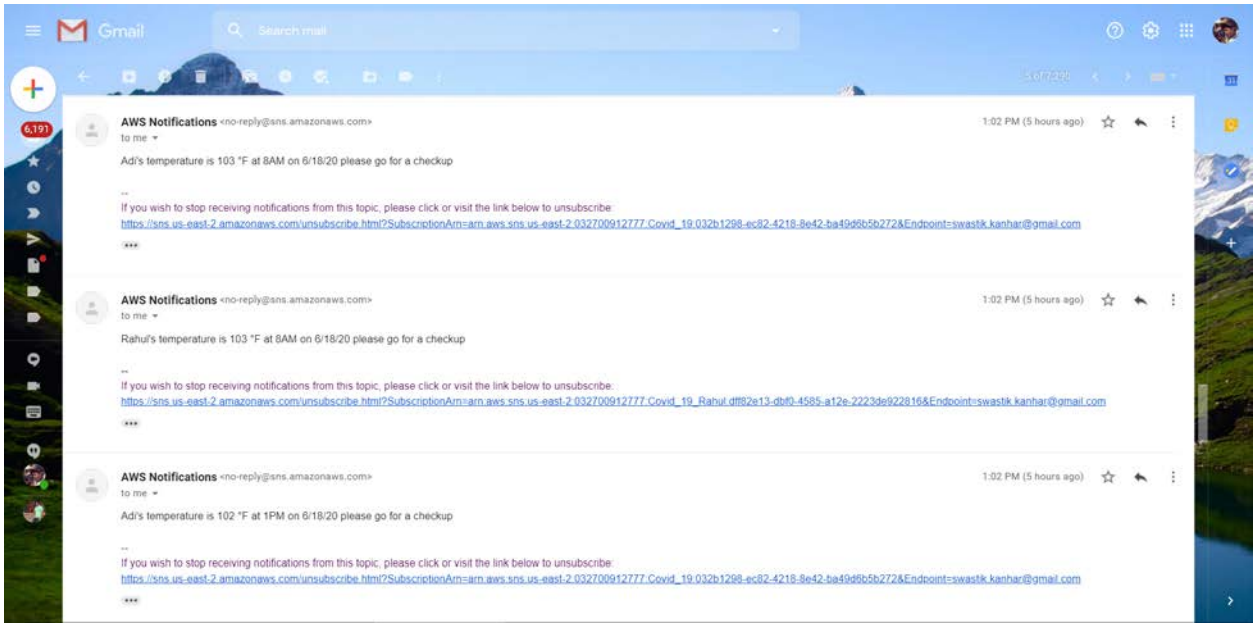
For Example:

Adi was the publisher of Covid_19, and rest of the three were subscribers for that same topic and so on.





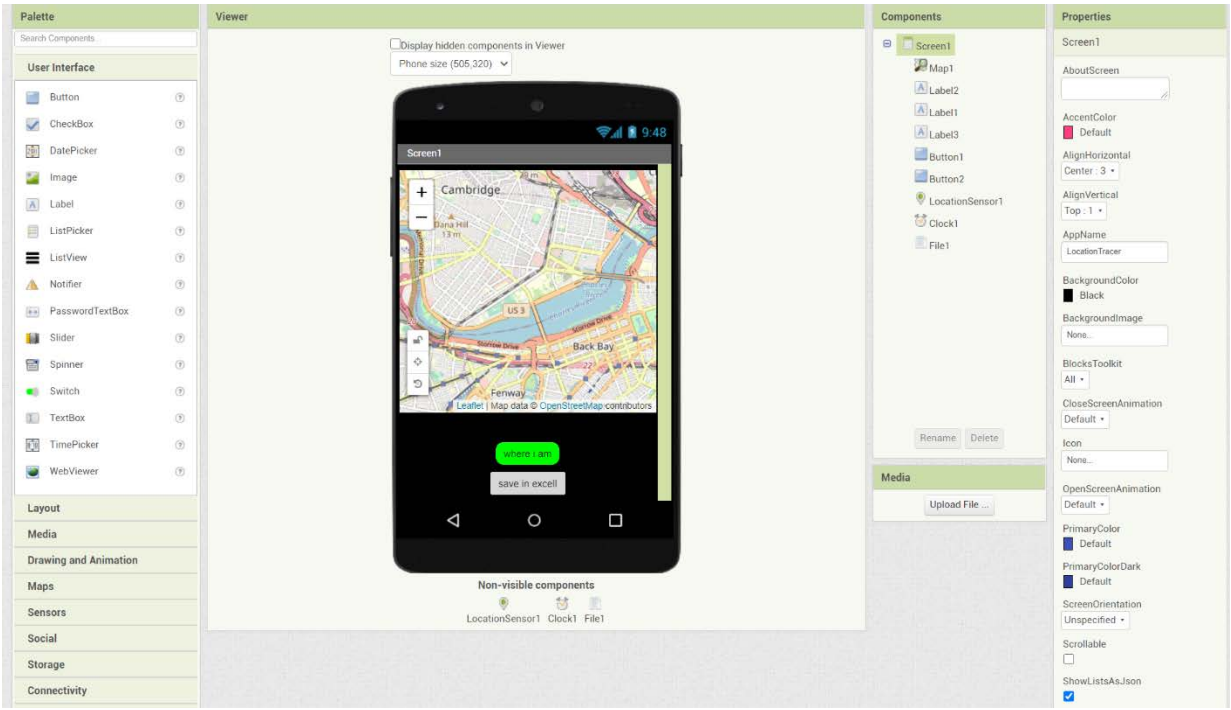
Here, the different topics are published and subscribers of all the topics are shown.



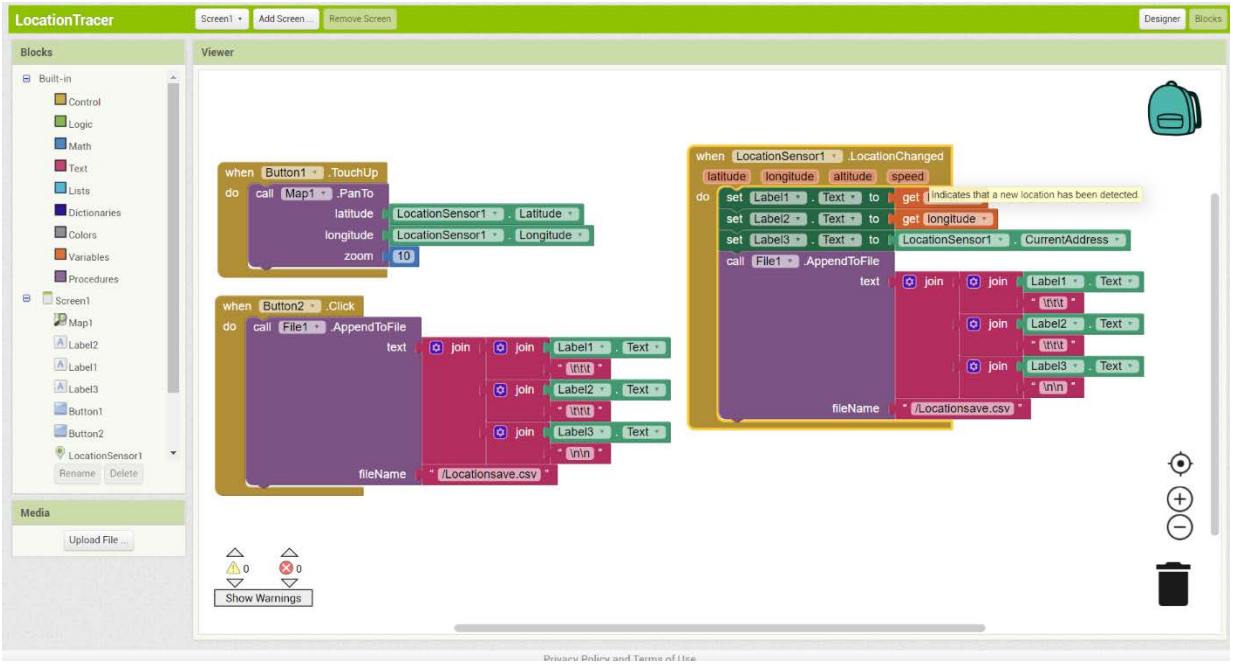
Here, we can see as a subscriber one gets an email every time any publisher's body temperature rises above 100°F and so do the other subscribers.

Part-3:

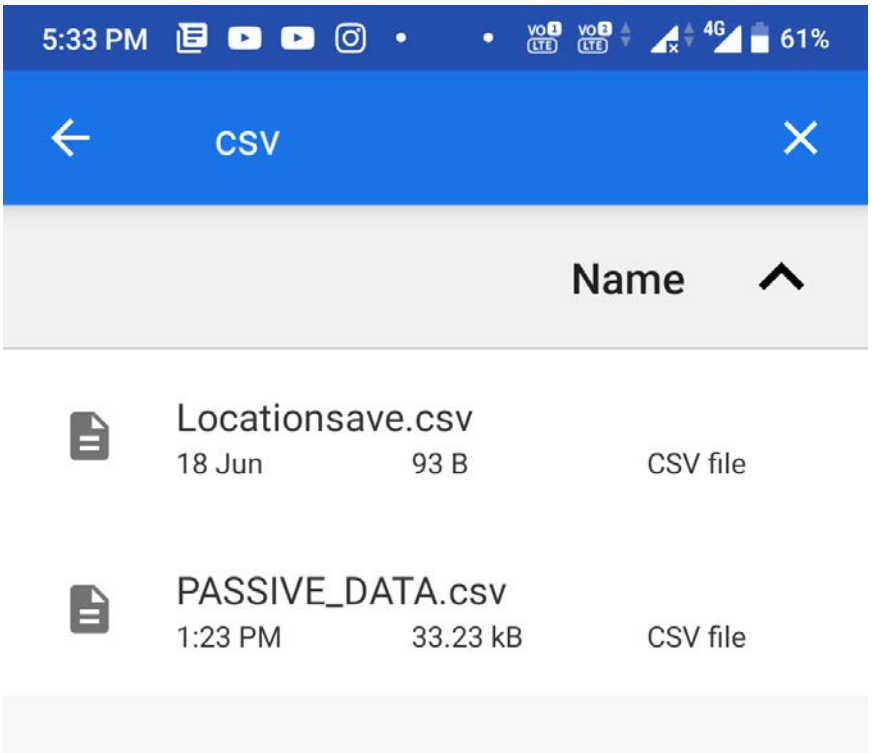
This part includes the procedure to get an app designed to track their individual locations using MIT App Inventor and to store those locations in a local database



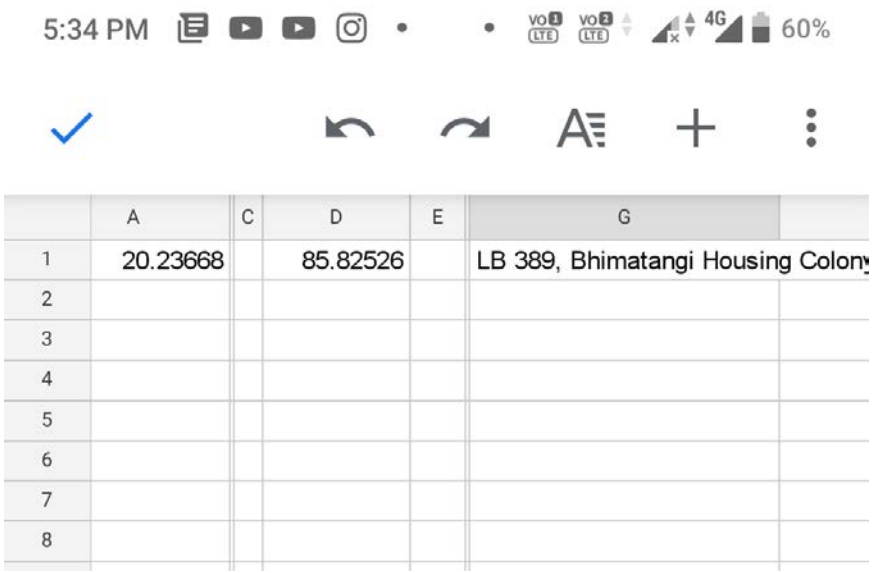
- We made a vertical layout and included a location sensor to trace the present location and to store that we used a file component for storing and retrieving files.
- We then included a 2-D container that renders map files in the background and then added three labels under the map and two switches.
    - **Label-1**: Latitude
    - **Label-2:** Longitude
    - **Label-3**: Current Address
    - **Switch-1**: Retrieve current latitude and longitude coordinates
    - **Switch-2**: To store current coordinates in local database
- When the location changes it triggers the file component and stores the location in the csv file.



- **Button-1:** This button calls the map to find the current latitude and longitude values
- **When location changes**: It automatically writes the latitude, longitude and current address in label-1, label-2 and label-3 respectively and appends them in the csv file. Name: *Locationsave.csv*
- **Button-2:** If user wants to manually append his location to the database then he can do so by clicking in this button.



We can find the database file Locationsave.csv in - internal memory/Andorid/data/Locationsave.csv



- The first column represents Latitude.
- The second column represents Longitude.
- The third column represents Current Address.

Conclusion:

This is a report based on COVID Patient Alert System designed to observe and upload your body temperature using MQTT broker and alert your friends who are regularly in physical contact with you if you show any symptoms of COVID 19 and gives emphasis to maintain social distancing amidst this pandemic.

STAY HOME
STAY SAFE

Know the COVID-19 SYMPTOMS

The following symptoms may appear 2-14 days after exposure:

- Fever
- Cough
- Shortness of Breath

Seek medical advice if:

- You develop worsening symptoms
- You have been in close contact with a person known to have COVID-19
- You live in or have recently been in an on area with ongoing spread of COVID-19