

Train Detection & Alert System (Quick Gate)

Status Document – 02



B.Sc. (Hons) Degree in Information Technology
Specialized Data Science

Department of Information Technology

Group: 23-302

Biyanwila B.D.V.J. – IT20212490

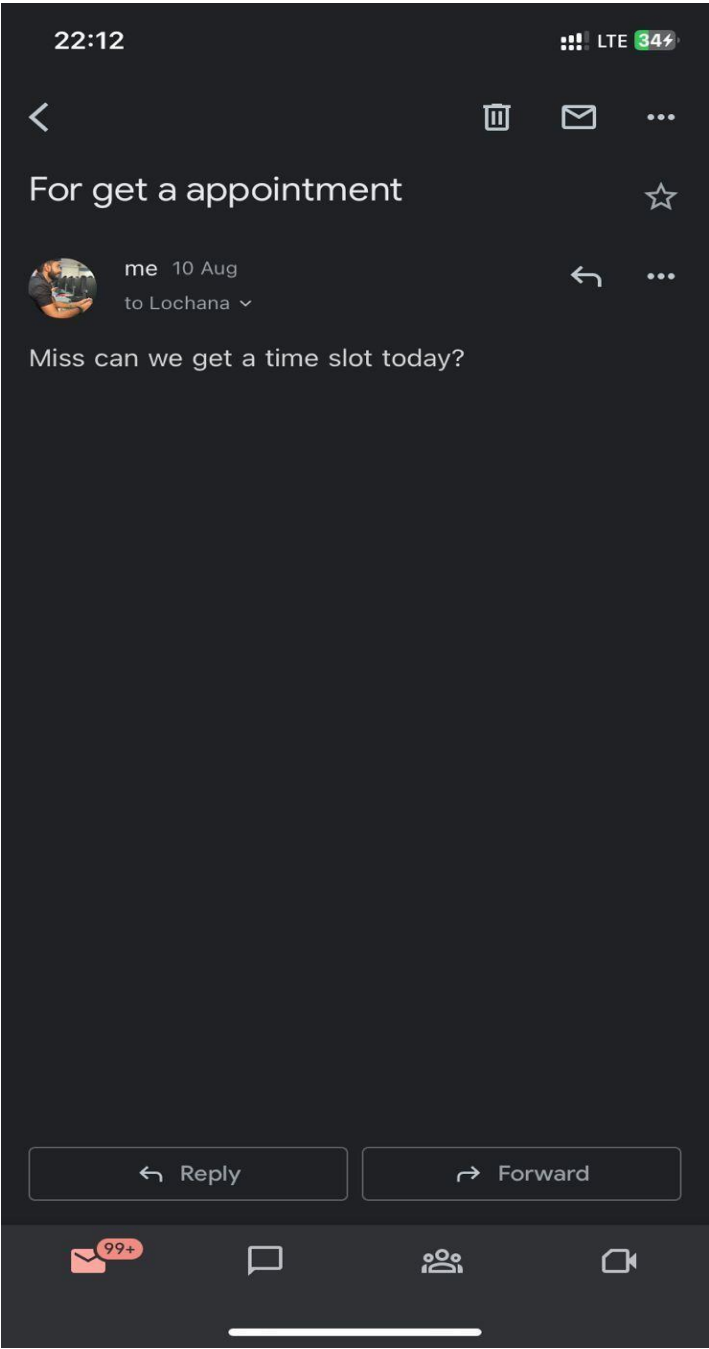
Submission Date: 07/09/2023

Contents

1. Communication with Supervisors	2
1.1. Visiting the Supervisor and the co-supervisor regularly	3
2. Research Group	10
2.1. Create a MS team group with team members.....	10
2.2. MS teams meetings with members.	10
2.3. MS teams files	11
2.4. Create a WhatsApp group with team members.	12
2.5. Chat in WhatsApp group with group members	13
2.6. Files shared in WhatsApp group with group members	14
2.7. WhatsApp group calls.	15
3. Update CDAP Cloud Submissions	16
4. Completed 90% of the Research Project.....	16
4.1 Main Back-end code.....	16
4.2 Defining the roads near the railway crossing.....	17
4.3. Distance Prediction display using API.....	18
4.4. Data retrieving to the Firebase Database	20

1. Communication with Supervisors

1.1. Visiting the Supervisor and the co-supervisor regularly.



16:51

LTE 72



Re: For Research Progress Meeting

Inbox



Lochana Rajamanthri 24 May

to me ▾



Dear Vihan,

Come and meet me at 9.30 am.

Best Regards,
Lochana Rajamanthri

Lecturer| SLIIT Business School | SLIIT |
www.sliit.lk/business<<http://www.sliit.lk/business>>

Phone: +94(0)11 754 4801 - Ext 4614 | Email:
lochana.r@sliit.lk

[cid:9b4528d7-0575-4a0d-a6eb-94d7daa594e6]
<<https://www.facebook.com/fobsliit/>>

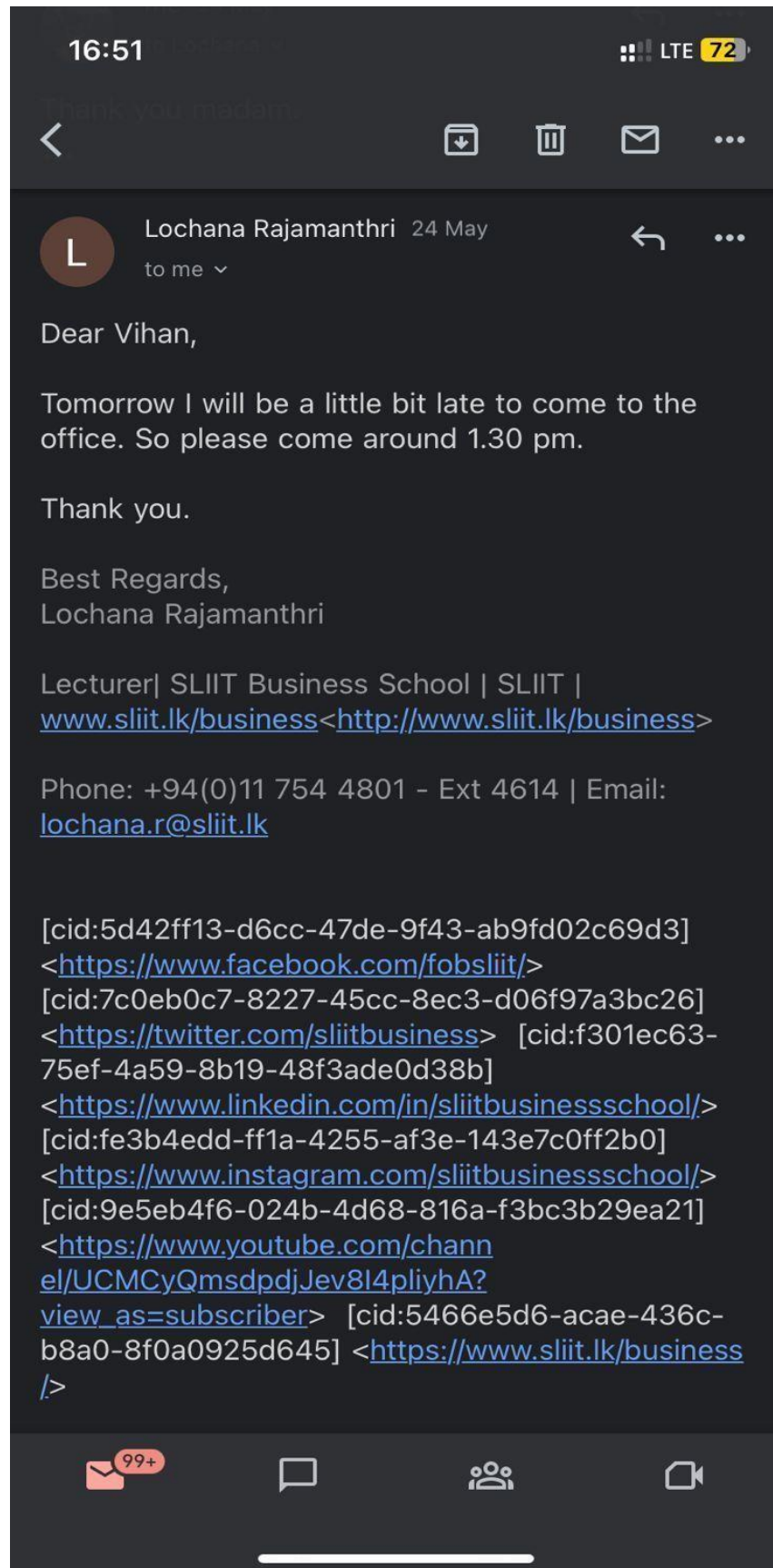
[cid:33b68246-b587-455d-9de1-1555a4c7af6f]
<<https://twitter.com/sliitbusiness>> [cid:a3a1ef60-

5cde-4fb5-9e86-9e82f00f0e45]
<<https://www.linkedin.com/in/sliitbusinessschool/>>

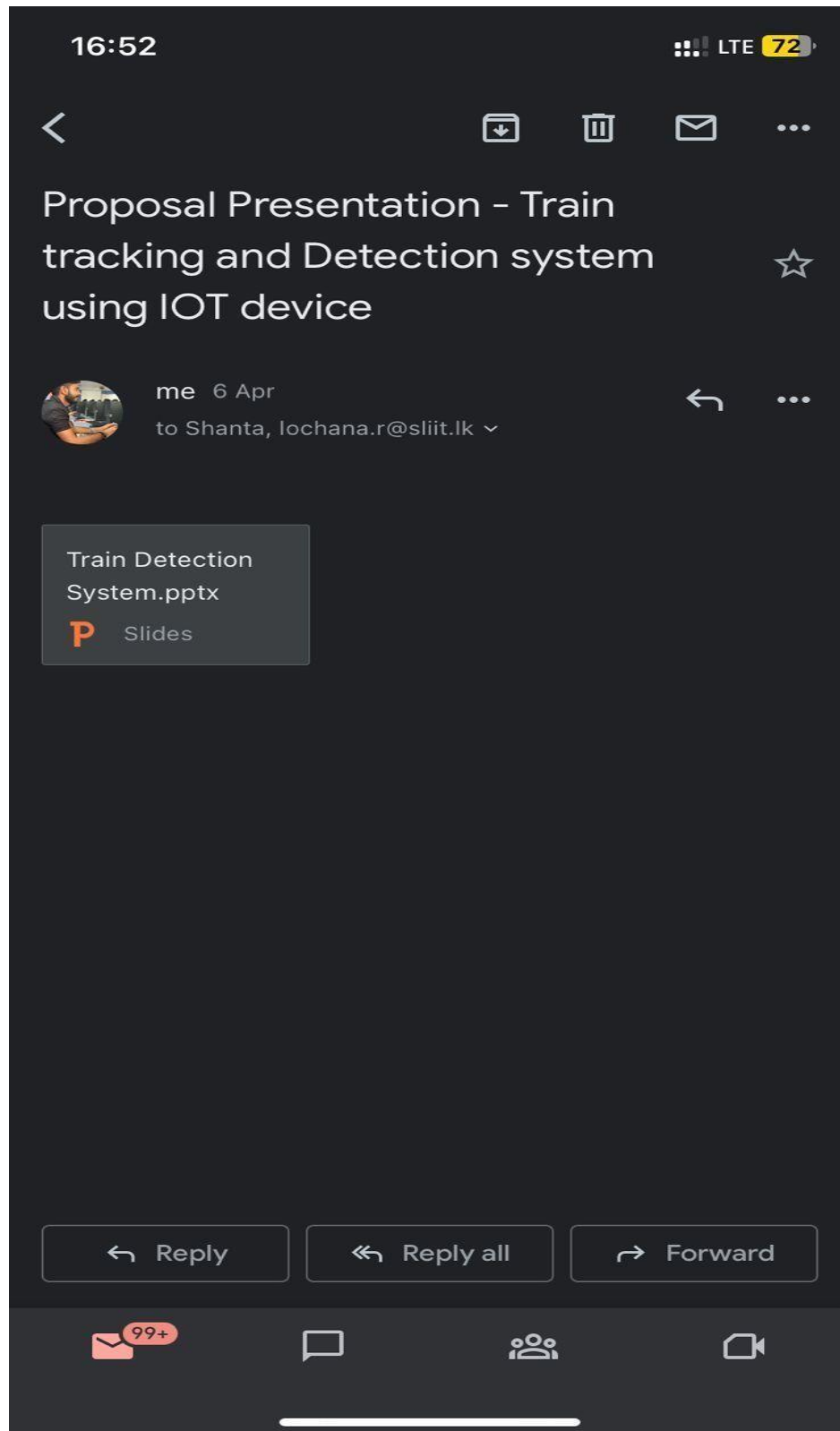
[cid:88295679-b7f8-435f-a5ab-f93ea4386997]
<<https://www.instagram.com/sliitbusinessschool/>>

[cid:7a3ce6b1-7a36-42b3-8f9b-7911246ec592]
<https://www.youtube.com/channel/UCMCyQmsdpdjJev8I4pliyhA?view_as=subscriber> [cid:edacb115-b90c-4a24-ba1c-db7d8105dfbd] <<http://www.sliit.lk/business>>





1.1. Sending research project-related files to the Supervisor and Co-supervisor



16:52

LTE 72



Train Tracking and Detection System for Citizens



me 14 Mar

to Shanta, lochana.r@sliit.lk



Train Tracking and Detection S...



zip

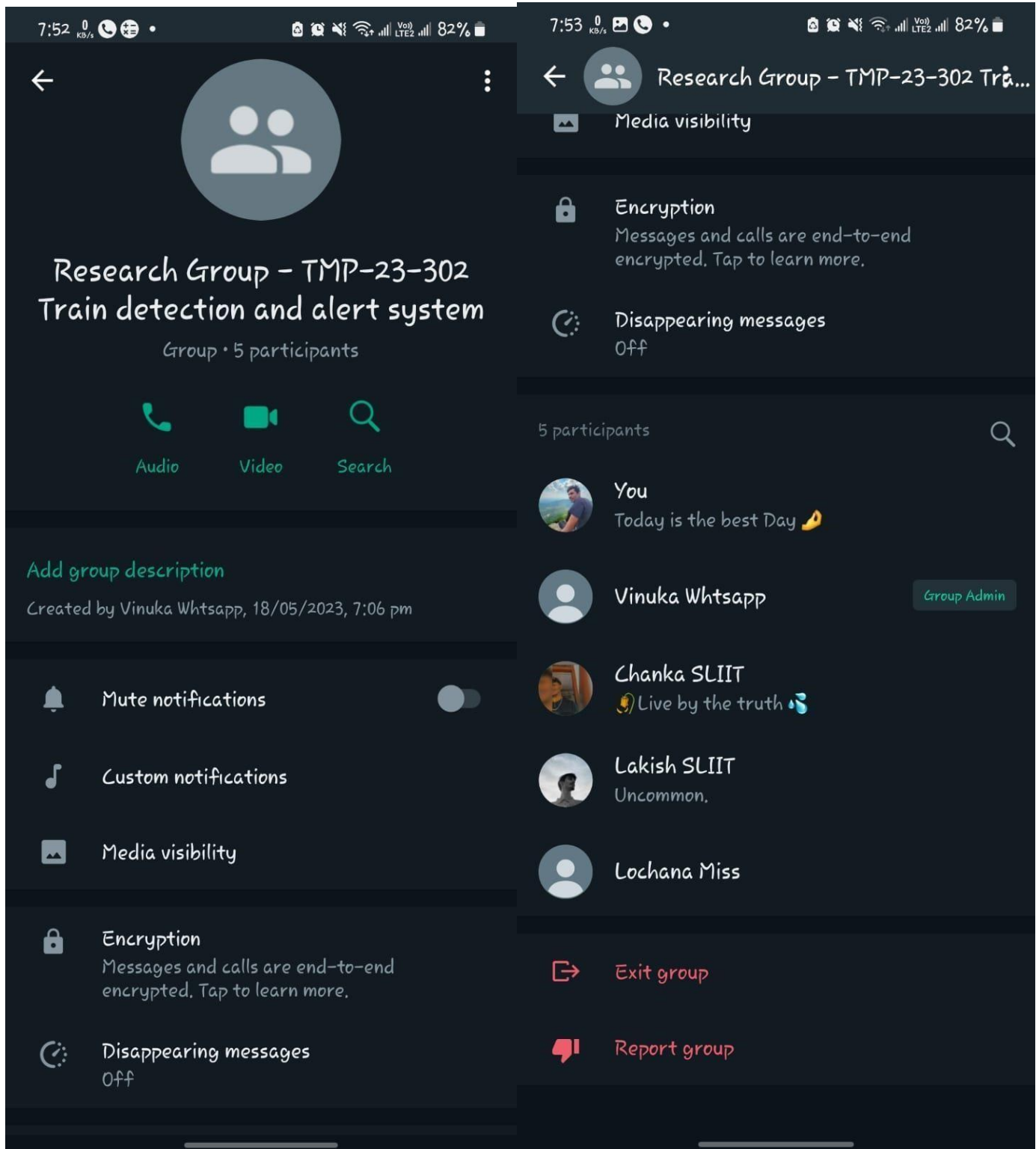
Reply

Reply all

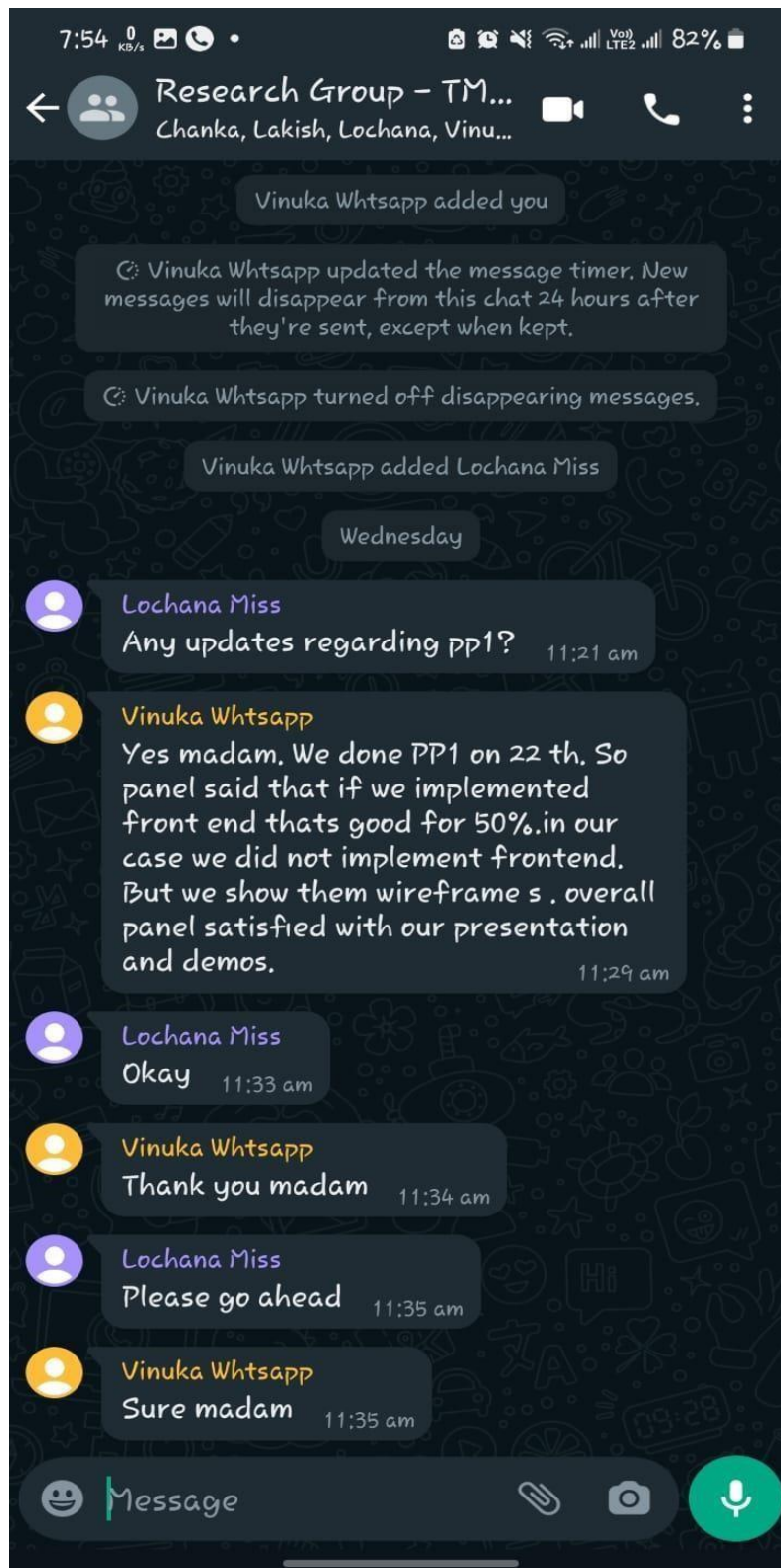
Forward



1.2. Create a WhatsApp Group with supervisor and co-supervisor

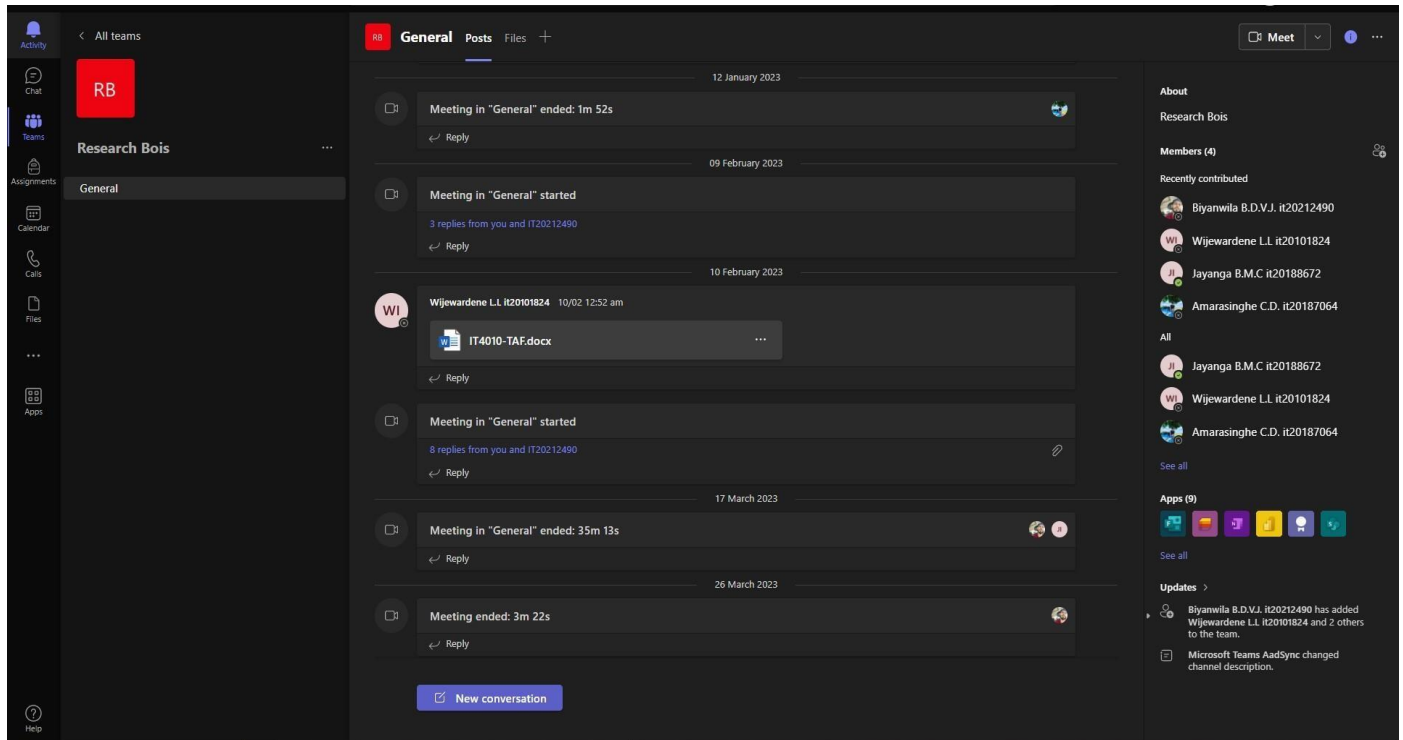


1.3. Share ideas with supervisor through WhatsApp chats.

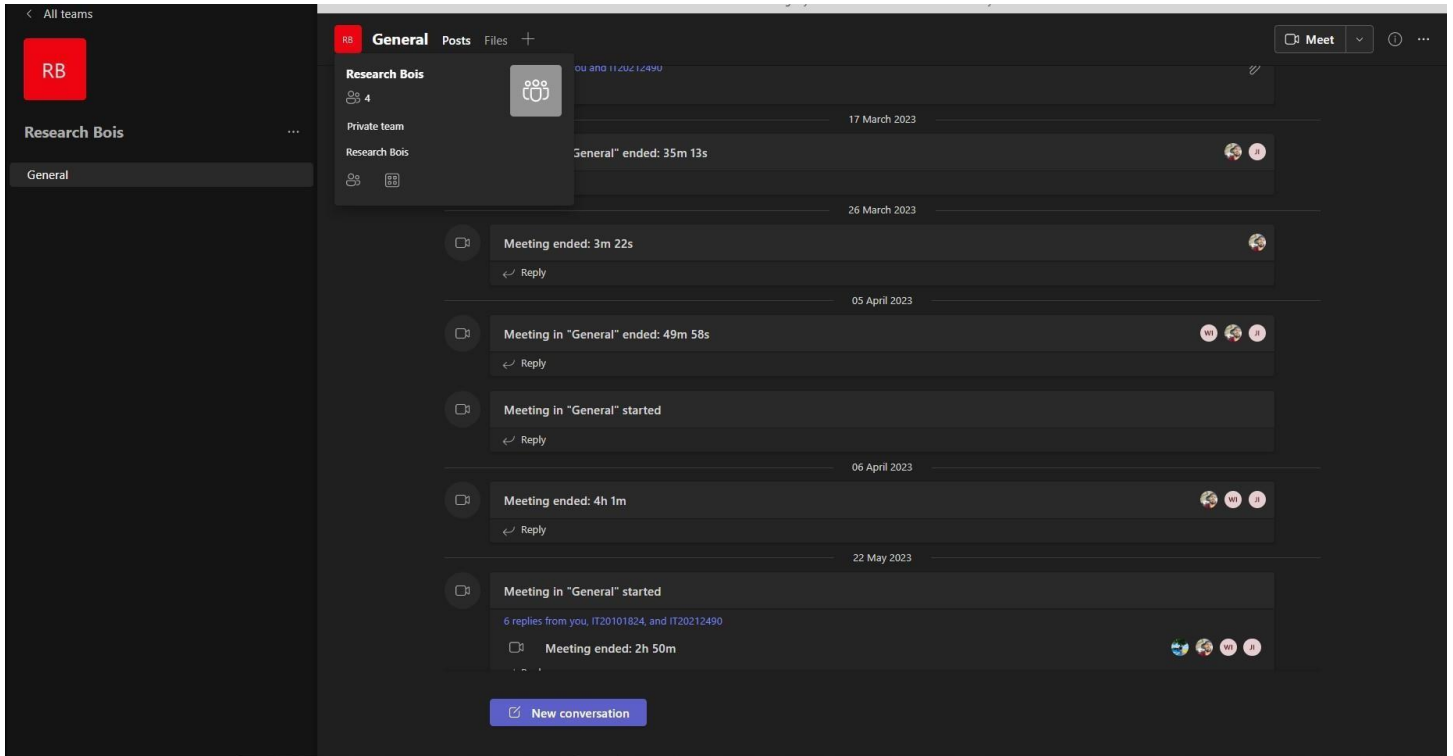


2. Research Group

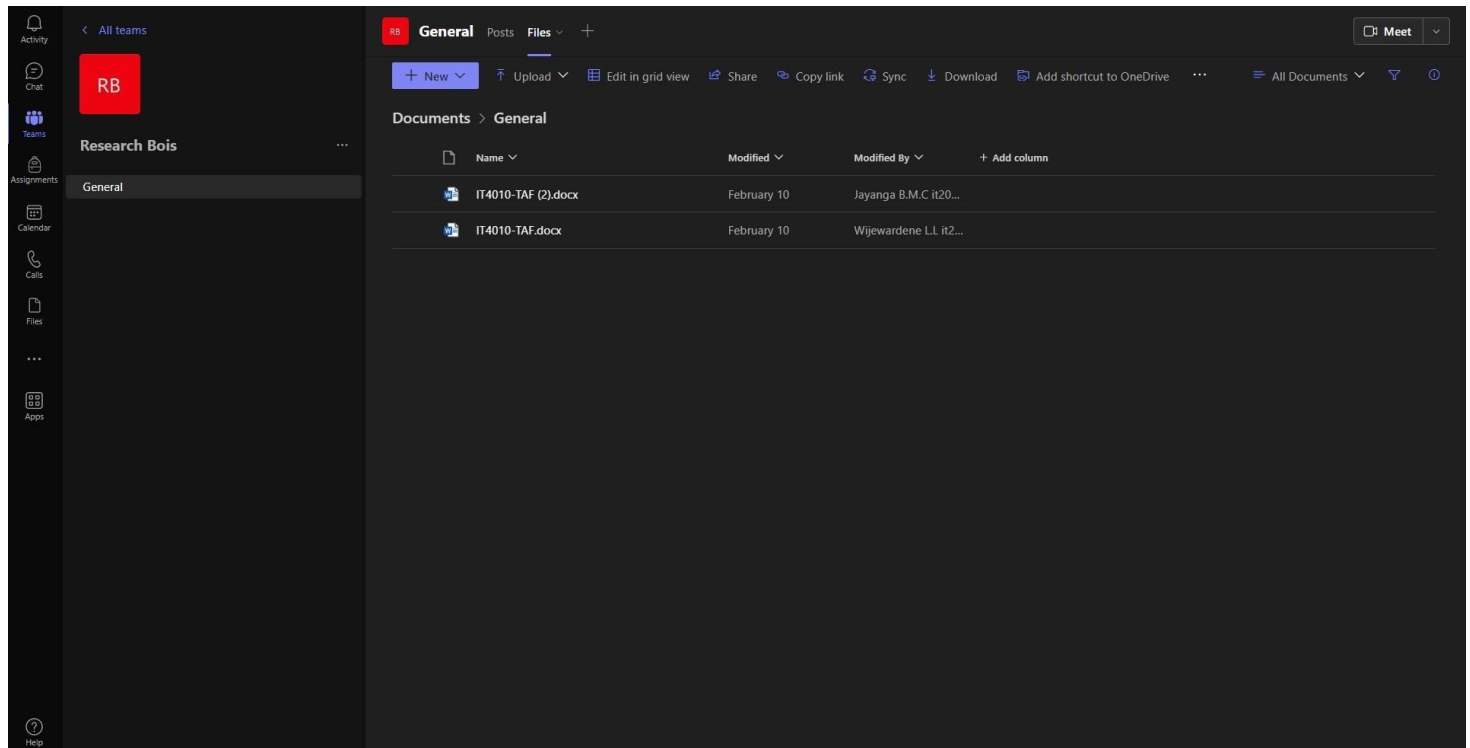
2.1. Create a MS team group with team members.



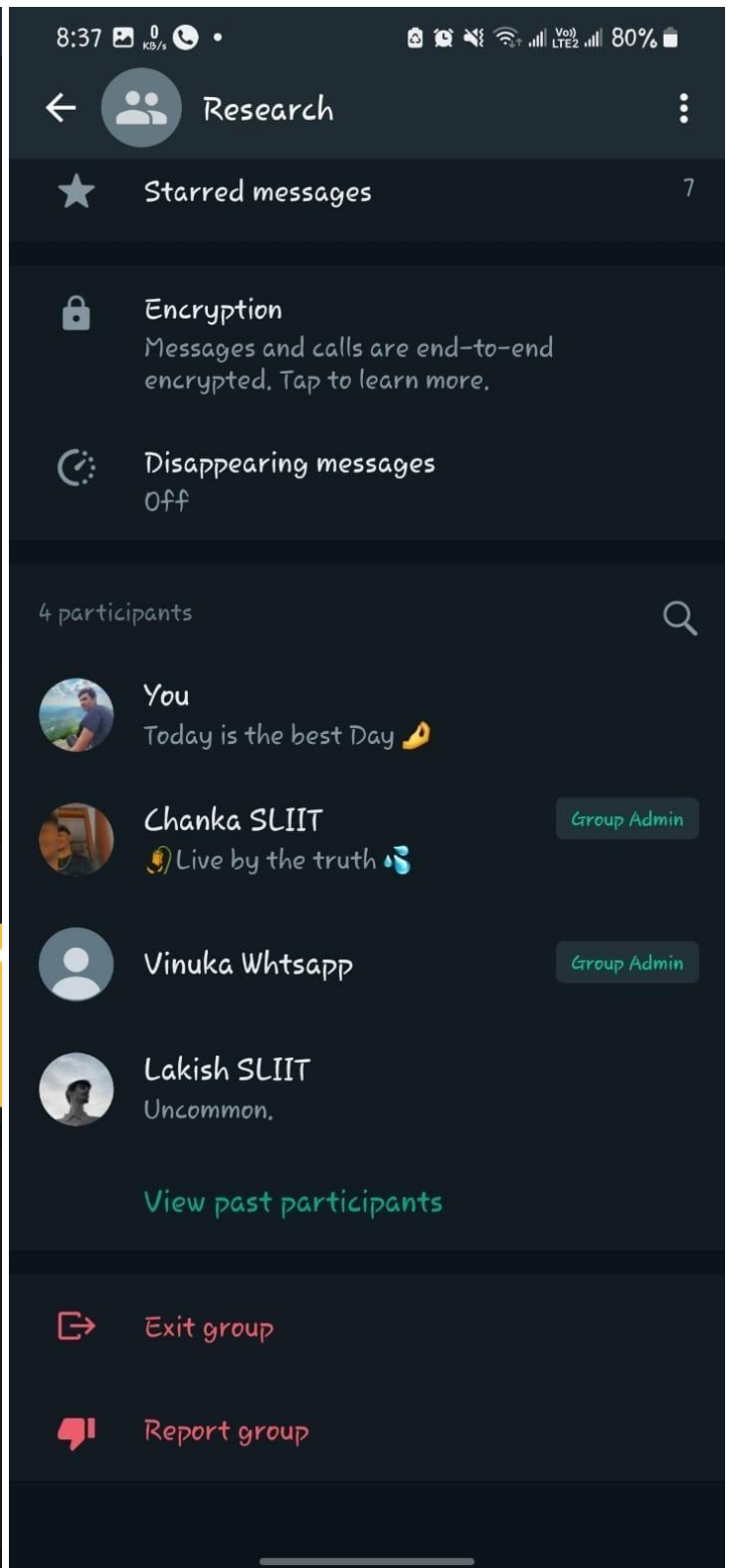
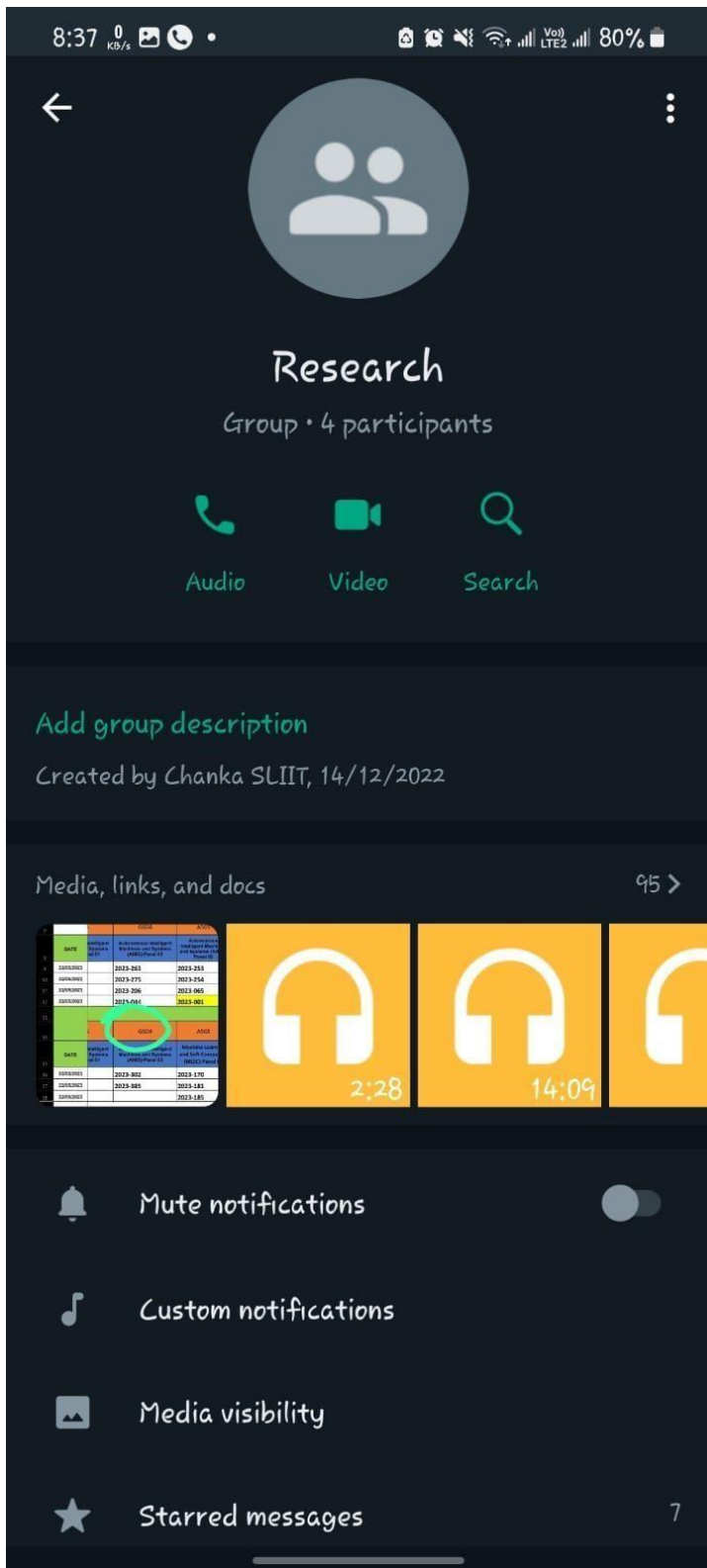
2.2. MS teams meetings with members.



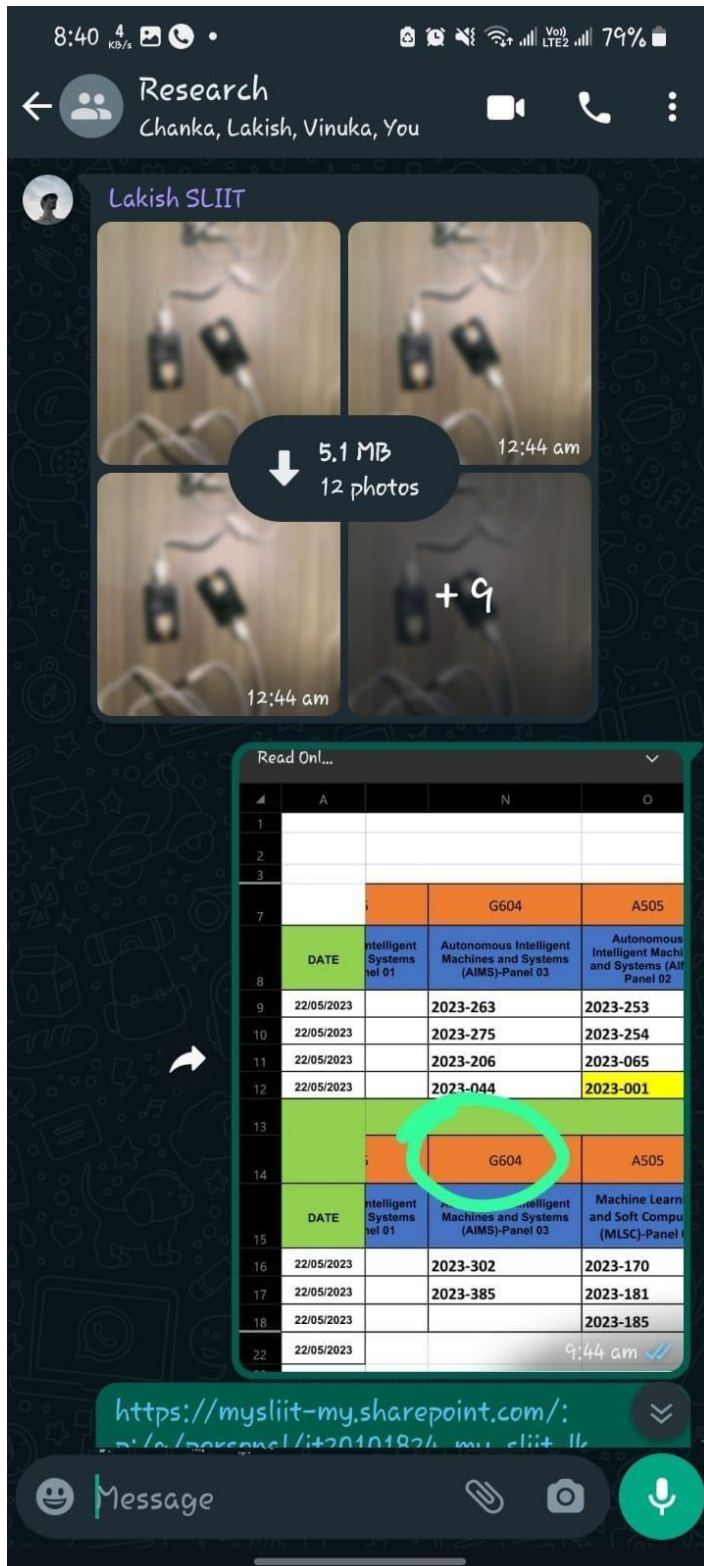
2.3. MS teams files



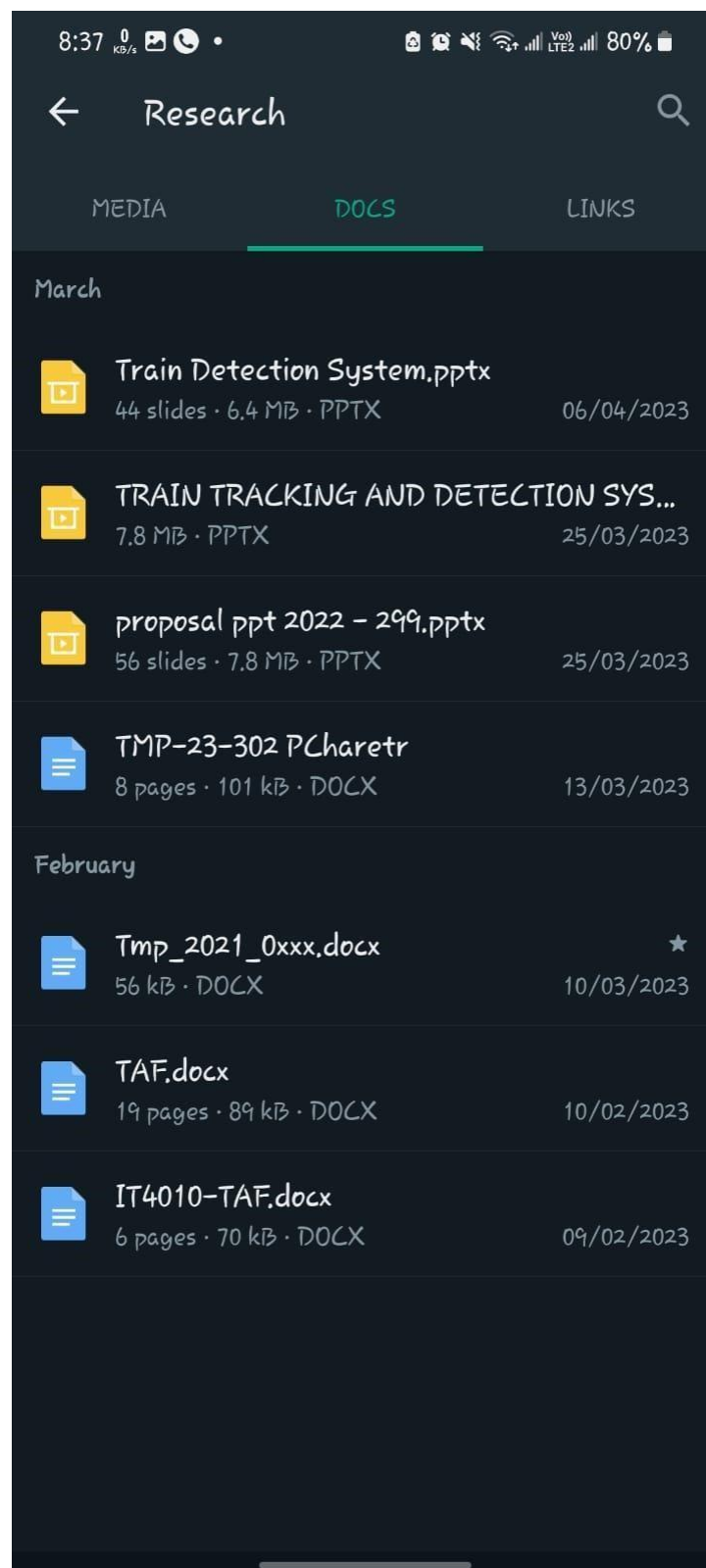
2.4. Create a WhatsApp group with team members.



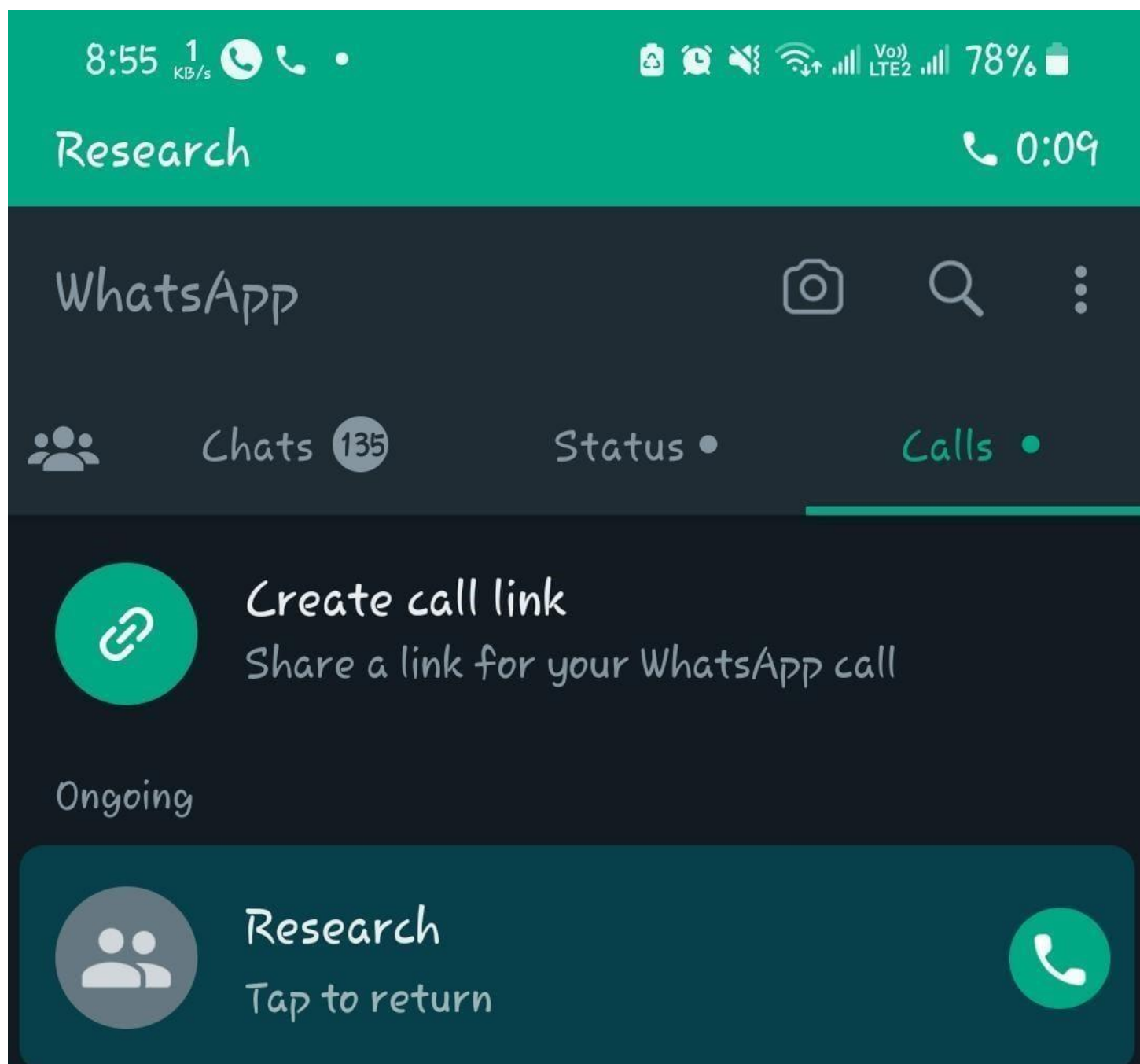
2.5. Chat in WhatsApp group with group members



2.6. Files shared in WhatsApp group with group members



2.7. WhatsApp group calls.



3. Update CDAP Cloud Submissions

SharePoint

Search this library

CDAPSubmissionCloud

Private group

Not following

2 members

New

Upload

Share

Copy link

Sync

Add shortcut to OneDrive

Download

Export to Excel

Automate

Integrate

All Documents

2023RegCloud > 23-302-Students

Name	Modified	Modified By	File Size
1. Project Proposal	April 20	CDAP SUIT	2 items
2. Status Document 1	April 20	CDAP SUIT	5 items
3. Progress Presentation - 1	April 20	CDAP SUIT	2 items
4. Research Paper	April 20	CDAP SUIT	5 items
5. Progress Presentation - 2	April 20	CDAP SUIT	3 items
6. Final Report & Presentation	April 20	CDAP SUIT	2 items
7. Status Document 2	April 20	CDAP SUIT	2 items
8. Website	April 20	CDAP SUIT	1 item
9. Log Book	April 20	CDAP SUIT	3 items
Marking Schemes	April 20	CDAP SUIT	1 item
Project Registration Documents	April 20	CDAP SUIT	5 items
Panel Comments for the Students.xlsx	July 2	CDAP SUIT	47.8 KB

Count

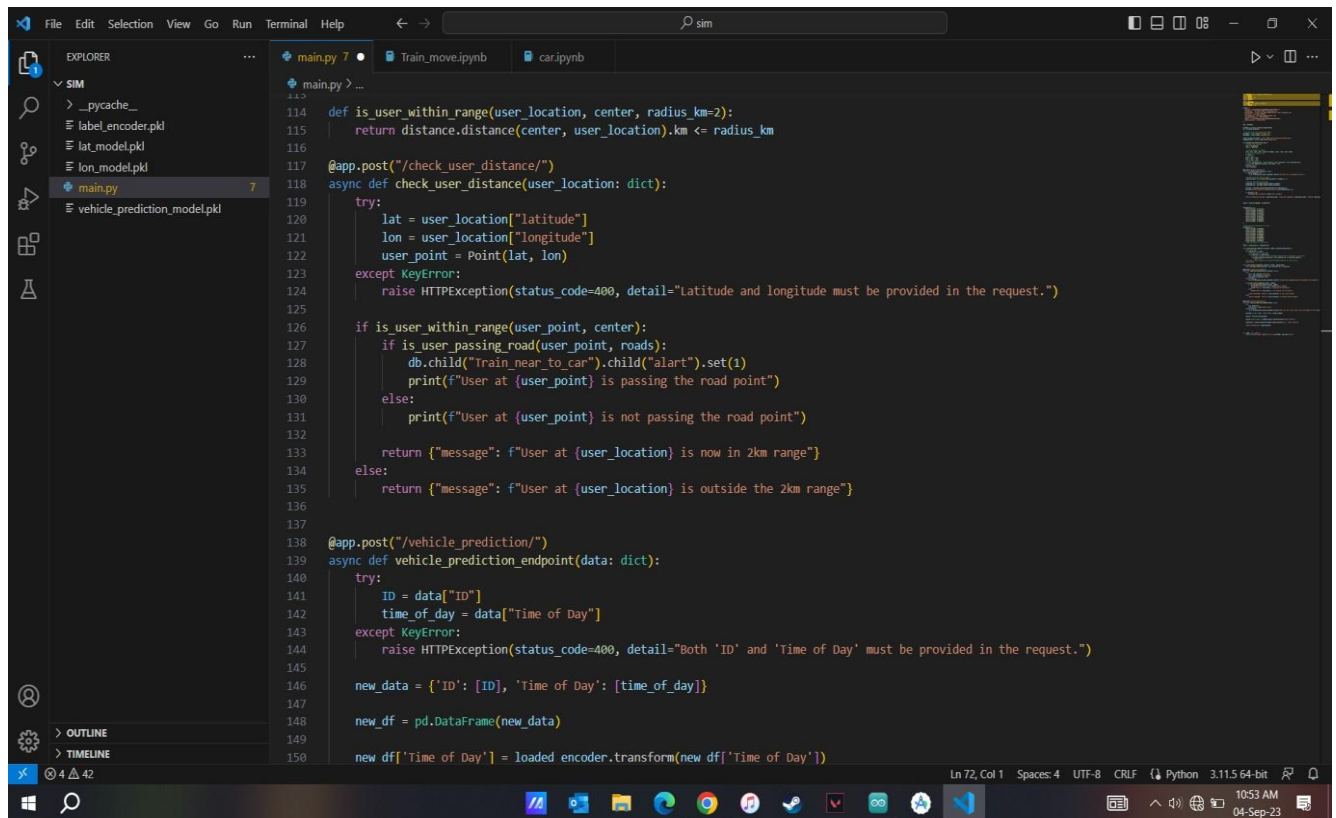
13

Count

2

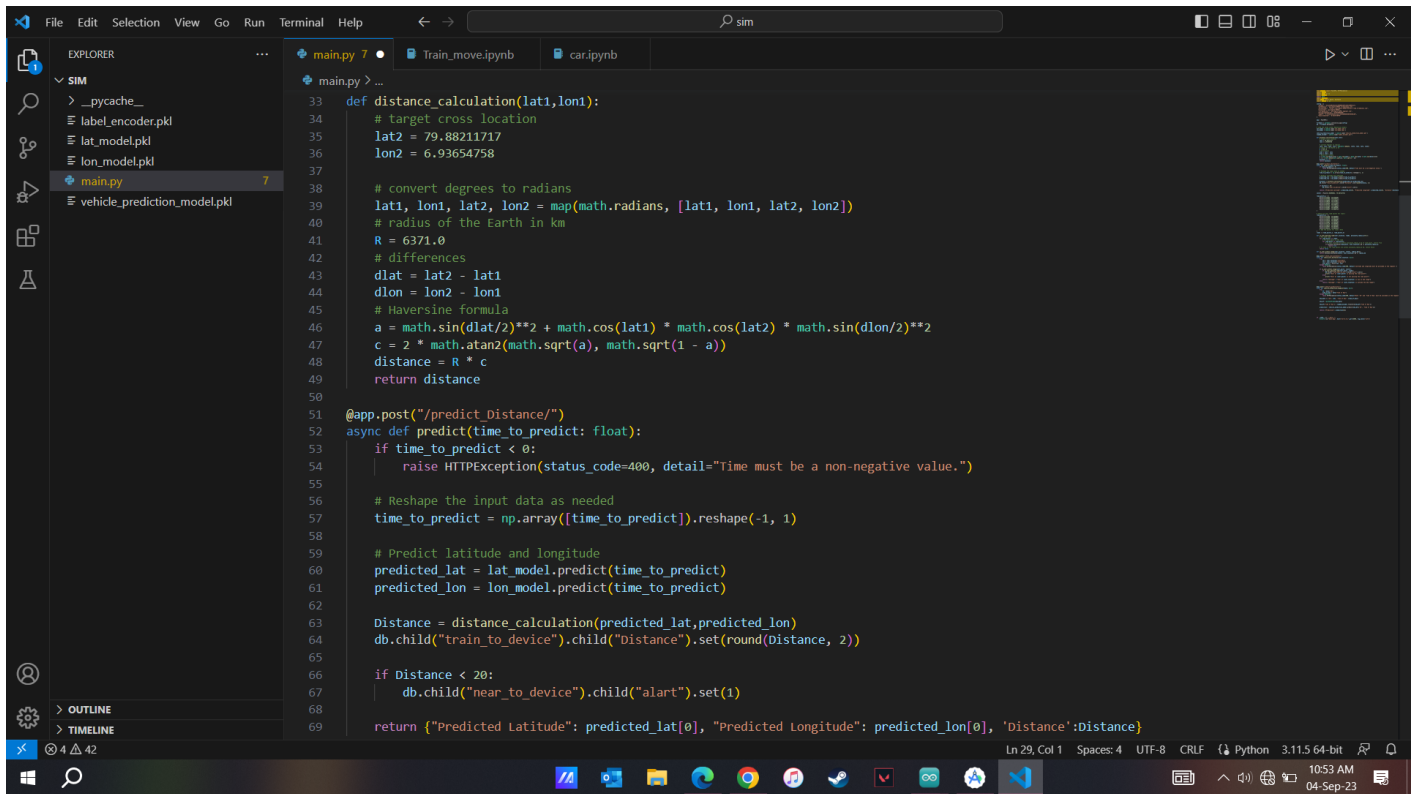
4. Completed 90% of the Research Project

4.1 Main Back-end code



```
114 def is_user_within_range(user_location, center, radius_km=2):
115     return distance.distance(center, user_location).km <= radius_km
116
117 @app.post("/check_user_distance/")
118 async def check_user_distance(user_location: dict):
119     try:
120         lat = user_location["latitude"]
121         lon = user_location["longitude"]
122         user_point = Point(lat, lon)
123     except KeyError:
124         raise HTTPException(status_code=400, detail="Latitude and longitude must be provided in the request.")
125
126     if is_user_within_range(user_point, center):
127         if is_user_passing_road(user_point, roads):
128             db.child("Train_near_to_car").child("alarm").set(1)
129             print(f"User at {user_point} is passing the road point")
130         else:
131             print(f"User at {user_point} is not passing the road point")
132
133         return {"message": f"User at {user_location} is now in 2km range"}
134     else:
135         return {"message": f"User at {user_location} is outside the 2km range"}
136
137
138 @app.post("/vehicle_prediction/")
139 async def vehicle_prediction_endpoint(data: dict):
140     try:
141         ID = data["ID"]
142         time_of_day = data["Time of Day"]
143     except KeyError:
144         raise HTTPException(status_code=400, detail="Both 'ID' and 'Time of Day' must be provided in the request.")
145
146     new_data = {'ID': [ID], 'Time of Day': [time_of_day]}
147
148     new_df = pd.DataFrame(new_data)
149
150     new_df['Time of Day'] = loaded_encoder.transform(new_df['Time of Day'])
```

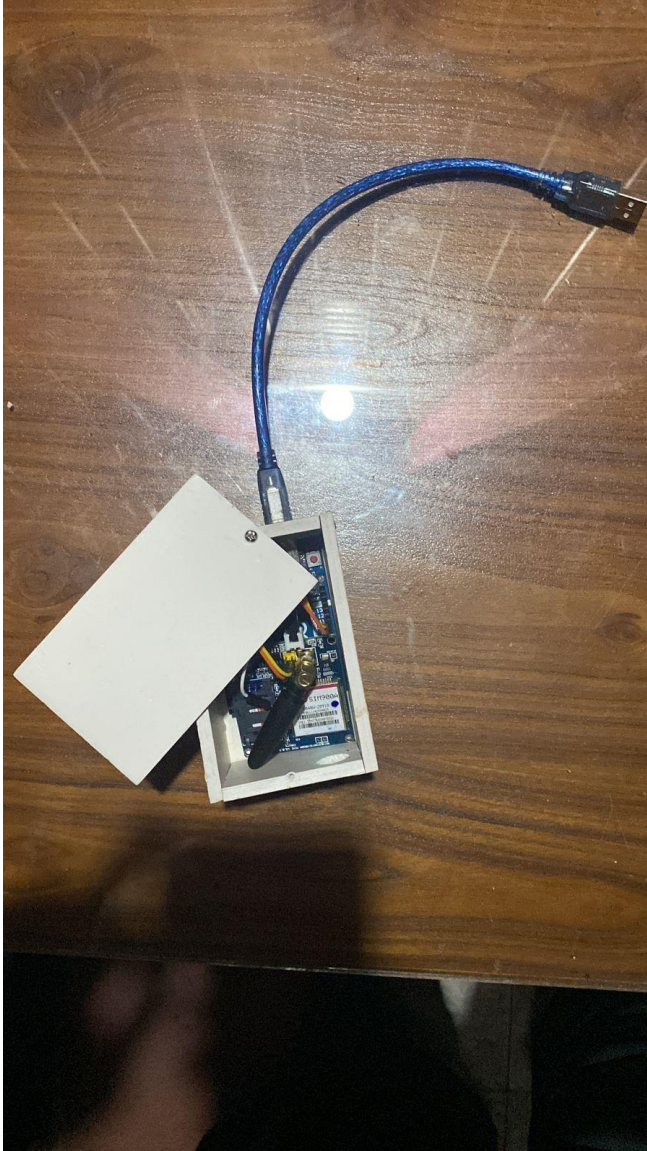
4.2 Defining the Train location and target location.



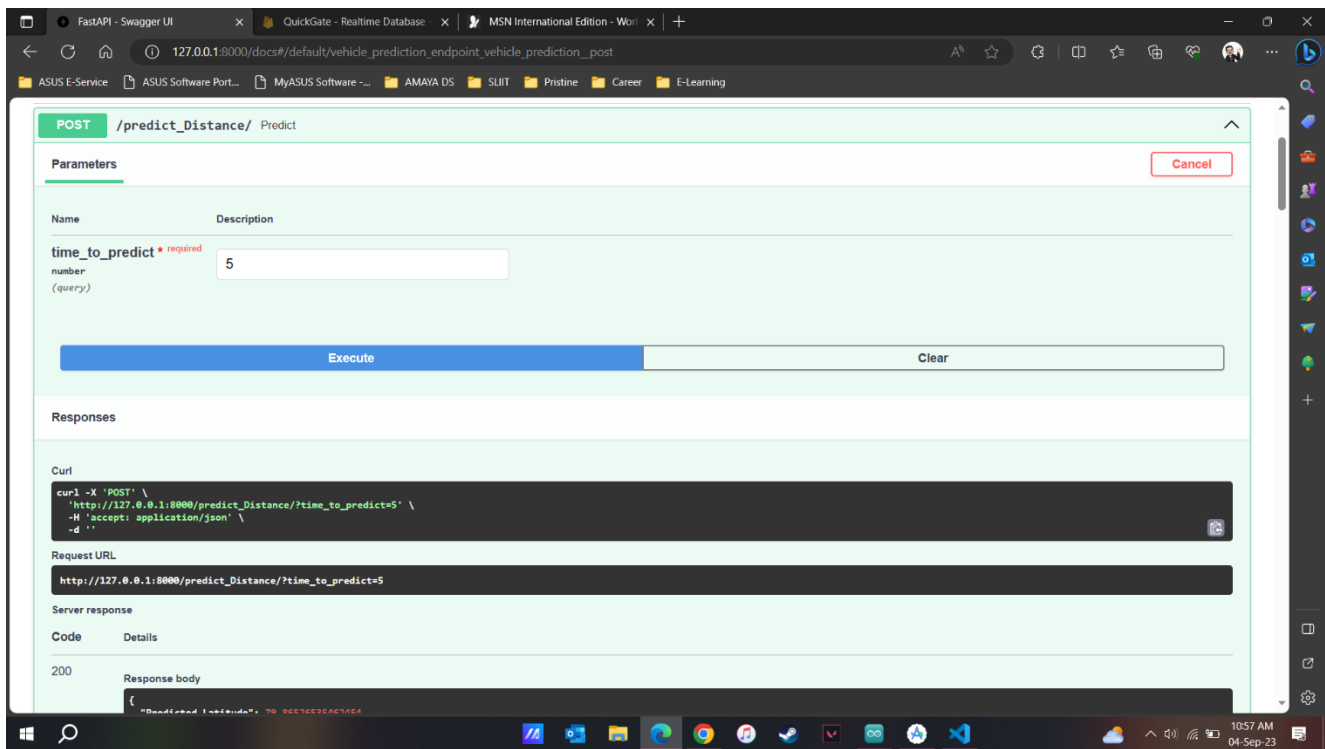
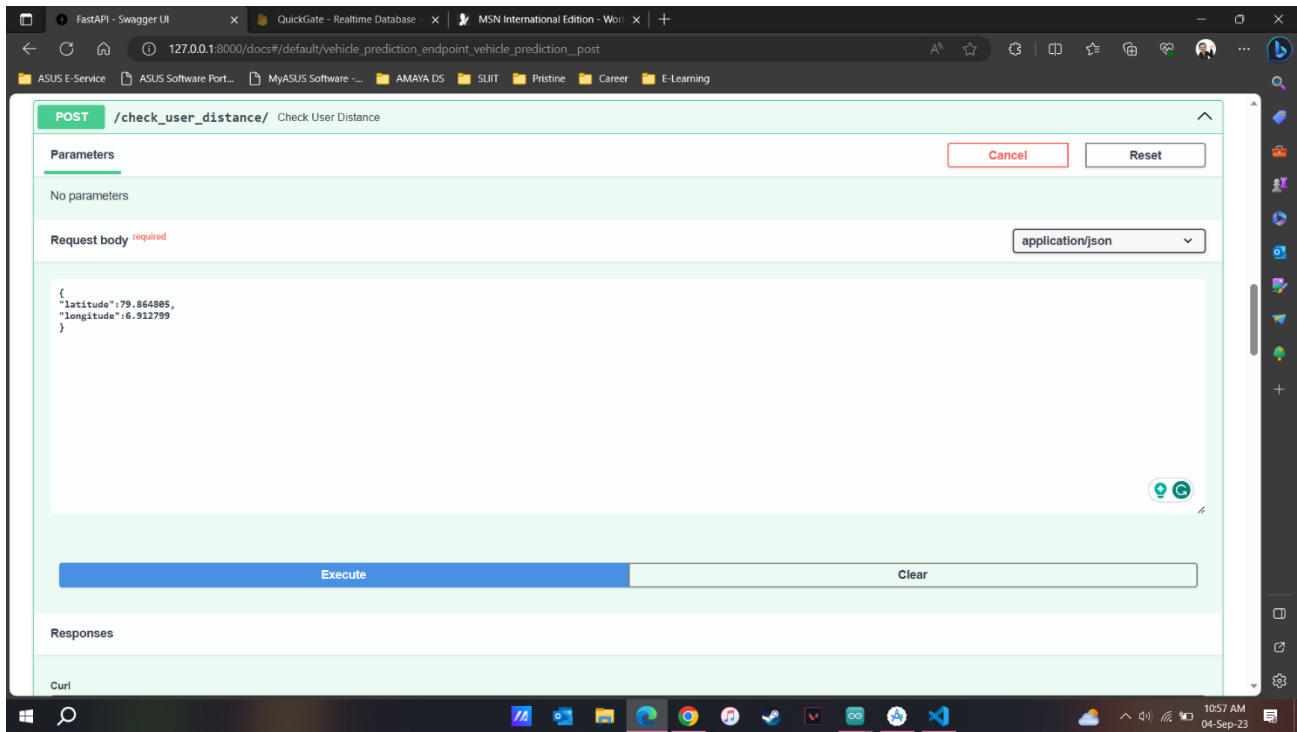
The screenshot shows a Visual Studio Code editor window with a Python file named `main.py` open. The file contains a function `distance_calculation` and an asynchronous `predict` function. The `distance_calculation` function takes `lat1` and `lon1` as arguments and returns a distance value. The `predict` function takes a `time_to_predict` argument and returns a dictionary with predicted latitude, longitude, and distance. The code is written in Python 3.11.5 64-bit. The status bar at the bottom shows the current line and column as Ln 29, Col 1, and the file encoding as UTF-8 CRLF.

```
33 def distance_calculation(lat1,lon1):
34     # target cross location
35     lat2 = 79.88211717
36     lon2 = 6.93654758
37
38     # convert degrees to radians
39     lat1, lon1, lat2, lon2 = map(math.radians, [lat1, lon1, lat2, lon2])
40     # radius of the Earth in km
41     R = 6371.0
42     # differences
43     dlat = lat2 - lat1
44     dlon = lon2 - lon1
45     # Haversine formula
46     a = math.sin(dlat/2)**2 + math.cos(lat1) * math.cos(lat2) * math.sin(dlon/2)**2
47     c = 2 * math.atan2(math.sqrt(a), math.sqrt(1 - a))
48     distance = R * c
49     return distance
50
51 @app.post("/predict Distance/")
52 async def predict(time_to_predict: float):
53     if time_to_predict < 0:
54         raise HTTPException(status_code=400, detail="Time must be a non-negative value.")
55
56     # Reshape the input data as needed
57     time_to_predict = np.array([time_to_predict]).reshape(-1, 1)
58
59     # Predict latitude and longitude
60     predicted_lat = lat_model.predict(time_to_predict)
61     predicted_lon = lon_model.predict(time_to_predict)
62
63     Distance = distance_calculation(predicted_lat,predicted_lon)
64     db.child("train_to_device").child("Distance").set(round(Distance, 2))
65
66     if Distance < 20:
67         db.child("near_to_device").child("alarm").set(1)
68
69     return {"Predicted Latitude": predicted_lat[0], "Predicted Longitude": predicted_lon[0], 'Distance':Distance}
```

4.3. IoT devices used for the system



4.4. Distance Prediction display using API



4.5. Data retrieving to the Firebase Database

