**Phase4**

**Development part-2**

**Hardware Setup:**

Install IoT devices like GPS trackers, sensors, and cameras on public transportation vehicles (buses, trains, etc.).

Ensure these devices are capable of collecting data related to location, passenger count, weather conditions, and vehicle health.

**Hardware Setup:**

**GPS Module:**

You'll need a GPS module to track the bus's location.

**Microcontroller** (e.g., Raspberry Pi or Arduino):

This will serve as the IoT device on the bus.

I**nternet Connectivity:**

To transmit data, you need internet connectivity, which could be through a GSM module or Wi-Fi.

**GPS Data Retrieval:**

Use Python libraries to interface with the GPS module and retrieve location data.

**Data Transmission**:

Transmit the GPS data to a central server. This can be done using MQTT, HTTP, or other communication protocols.

**Server-Side Programming**:

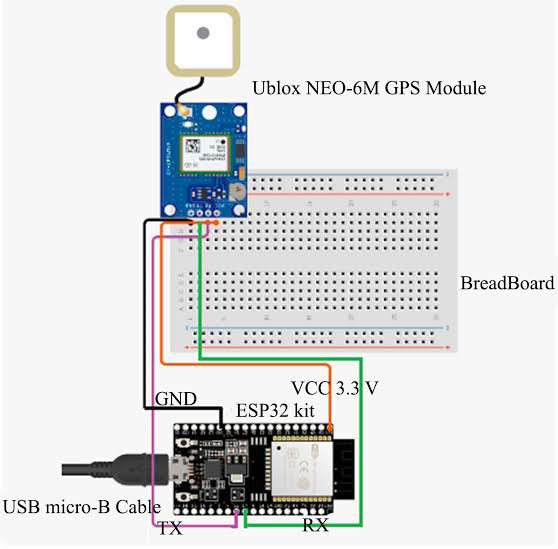
Develop a server application in Python or any other suitable language to receive and store the GPS data.Store data in a database to track the historical location of the bus.

**Web Interface** (Optional):

Create a web-based interface for users to track the bus's real-time location.

**Real-time Monitoring:**

Implement real-time monitoring features to display the bus's current location and possibly other information (speed, direction, etc.).



**PYTHON CODE:::**

# Import necessary libraries (ensure you have these installed)

import random

import time

from threading import Thread

**# Simulate IoT data from buses (longitude, latitude, passenger count)**

**class BusSimulator:**

def \_\_init\_\_(self, bus\_id):

self.bus\_id = bus\_id

self.latitude = random.uniform(30.0, 35.0)

self.longitude = random.uniform(-120.0, -115.0)

self.passenger\_count = random.randint(0, 50)

def update\_data(self):

while True:

# Simulate data changes

self.latitude += random.uniform(-0.01, 0.01)

self.longitude += random.uniform(-0.01, 0.01)

self.passenger\_count = random.randint(0, 50)

time.sleep(10) # Simulate data update every 10 seconds

**# Simulate multiple buses**

num\_buses = 5

buses = []

for i in range(num\_buses):

bus = BusSimulator(bus\_id=i)

buses.append(bus)

t = Thread(target=bus.update\_data)

t.daemon = True

t.start()

**# Passenger app to get real-time information**

class PassengerApp:

def \_\_init\_\_(self):

pass

def get\_real\_time\_info(self, bus):

while True:

print(f"Bus {bus.bus\_id} - Latitude: {bus.latitude:.4f}, Longitude: {bus.longitude:.4f}, Passengers: {bus.passenger\_count}")

time.sleep(10) # Check for updates every 10 seconds

**# Simulate passenger apps**

passenger\_apps = []

for bus in buses:

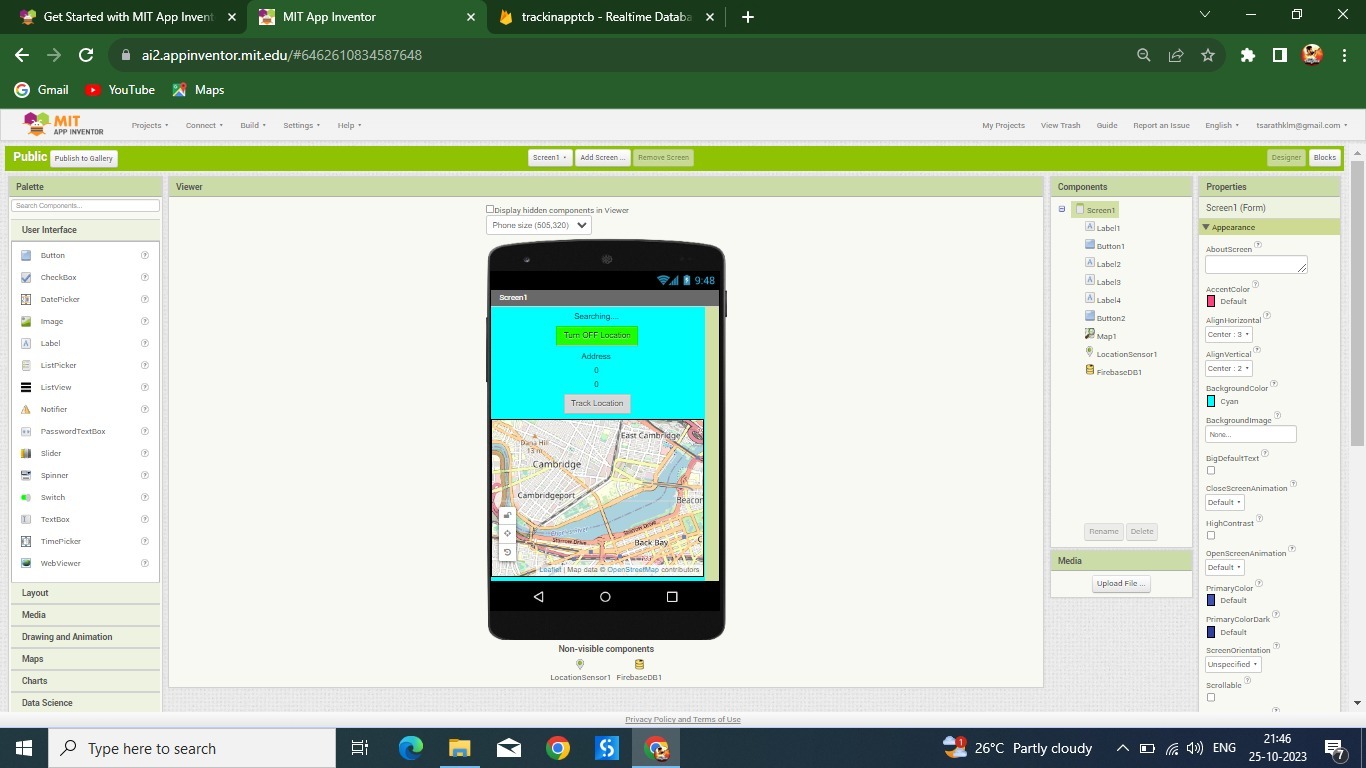
passenger\_app = PassengerApp()

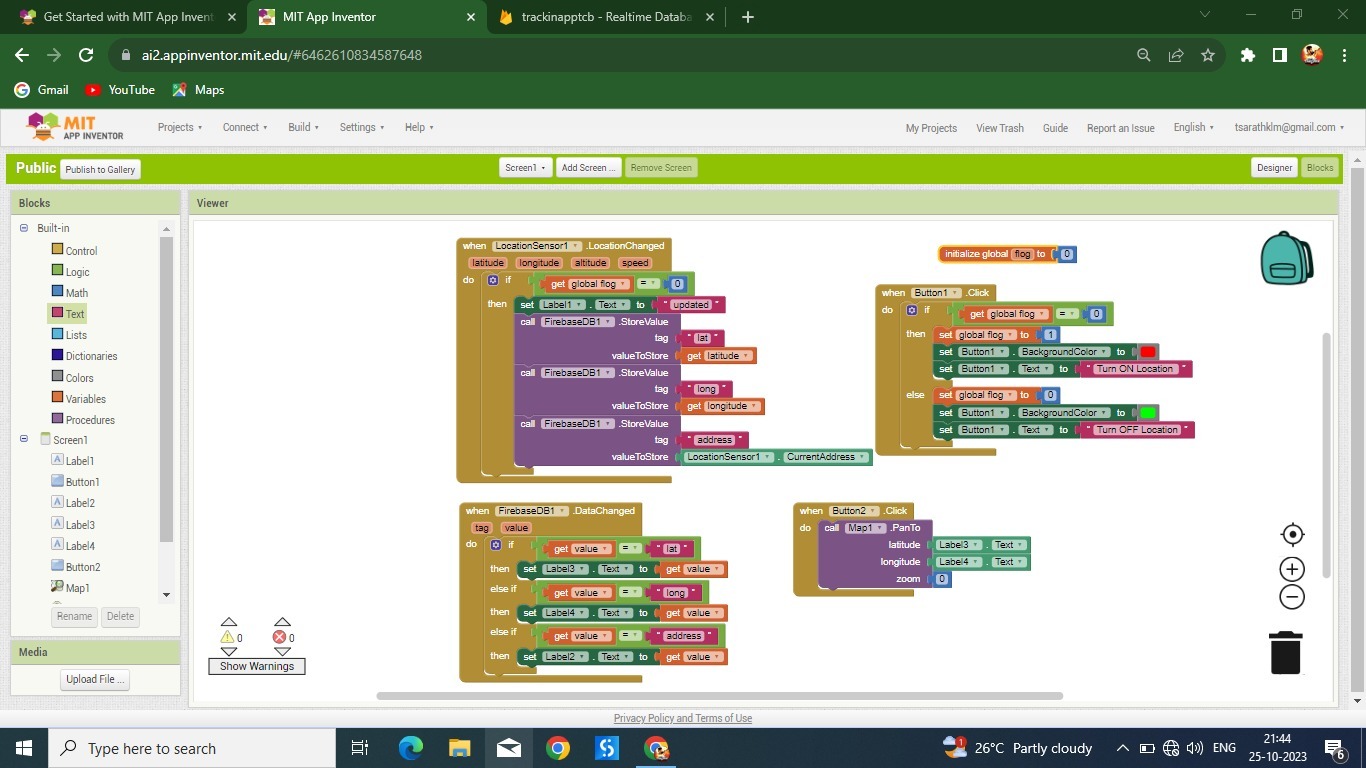
t = Thread(target=passenger\_app.get\_real\_time\_info, args=(bus,))

t.daemon = True

t.start()

**Public Transportation Optimization Project is Tracking using MIT App Inventor**





**Conclusion:**

Public transportation optimization is an important field that can benefit from technology like MIT App Inventor. MIT App Inventor is a platform that allows users to create mobile applications, and it can be used to develop apps related to public transportation, such as route planning, real-time tracking, or ticketing systems.