

**KGiSL INSTITUTE OF TECHNOLOGY**

(Approved By AICTE, New Delhi, Affiliate to Anna University

Recognized by UGC, Accredited by NBA(IT)

265, KGISL Campus, Thudiyalur Road, Saravanampatti, Coimbatore-641035**.)**

**DEPARTMENT OF**

**ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**

**NAAN MUDHALVAN - INTERNET OF THINGS**

**PUBLIC TRANSPORT OPTIMIZATION**

**NAME:** MAILEASH KANNA V.S

**REG NO:** 711721243056

**NM ID:** au711721243056

**TEAM MENTOR:** Mr**.** Mohankumar M

**TEAM EVALUATOR:** Ms. Akilandeeswari M

**Phase 3: Development Part 1**

**Building a public transport optimization system using IoT sensors**

**Materials and Components Needed:**

1. IoT Sensors

2. Microcontrollers or Single-Board Computers

3. Power Supplies

4. Connectivity Options

5. Housings and Mounting Equipment

6. Development Tools and Accessories

7. Sensors and Modules for Additional Data Collection (Optional)

8. Software Components

9. Transit Information Platform

10. Data Storage and Analysis Tools

11. Real-time Monitoring and Optimization Software

12. Security and Privacy Measures

13. Scaling and Maintenance Plan

14. Documentation and User Manuals

15. Regulatory Approvals and Permits

**Procedure:**

**1. Planning and Requirements Analysis:**

• Define the project's objectives and requirements.

• Identify the specific IoT sensors and data you need to collect.

• Determine your budget and timeline for implementation.

**2. Selecting IoT Sensors:**

• Choose appropriate sensors for real-time location and ridership data.

• Consider additional sensors for temperature, cameras, or other data.

**3. Hardware Setup:**

• Install IoT sensors in each public transportation vehicle.

• Connect sensors to microcontrollers or single-board computers.

**4. Development Environment Setup:**

• Set up the development environment on the microcontrollers.

• Install Python and necessary libraries for script development.

**5. Write Python Script:**

• Develop a Python script that collects data from sensors.

• Process the data and send it to the transit information platform.

**6. Transit Information Platform:**

• Set up a central platform to receive and process sensor data.

• Create RESTful APIs for data ingestion and storage.

**7. Data Storage and Analysis:**

• Store the received data in a database or data storage solution.

• Implement data analysis tools for real-time tracking and reporting.

**8. Real-time Monitoring and Optimization:**

• Use data for real-time monitoring and route optimization.

• Implement algorithms for resource allocation and scheduling.

**9. Security and Privacy:**

• Ensure data security through encryption and access controls.

• Comply with data privacy regulations and obtain necessary permissions.

**10. Scale and Maintain:**

• Scale the infrastructure as your transportation system grows.

• Plan for regular maintenance and troubleshooting.

**11. Documentation and User Manuals:**

• Create documentation for installation and maintenance procedures.

• Prepare user manuals for system operators and administrators.

**12. Regulatory Compliance:**

• Ensure compliance with local regulations and obtain required permits.

• Address any legal and ethical concerns related to data collection

**Python Scripts on IoT**

import requests

import time

from gps\_module import get\_current\_location # Implement GPS module functions

from passenger\_counter import count\_passengers # Implement passenger counter module

# Define the URL of the transit information platform

platform\_url = "https://your-transit-platform.com/api/data"

while True:

try:

# Get GPS location

current\_location = get\_current\_location() # Implement this function

# Count passengers

passenger\_count = count\_passengers() # Implement this function

# Create a data payload

payload = {

"vehicle\_id": "12345", # Replace with the actual vehicle ID

"timestamp": time.time(),

"location": current\_location,

"passenger\_count": passenger\_count

}

# Send data to the platform

response = requests.post(platform\_url, json=payload)

if response.status\_code == 200:

print("Data sent successfully")

else:

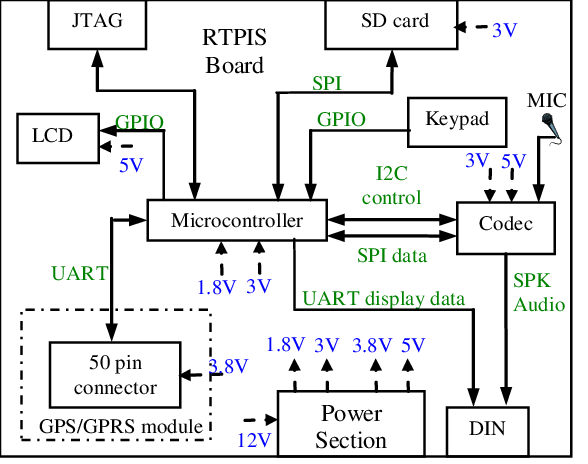
print("Failed to send data")

# Wait for a predefined interval before sending the next update

time.sleep(60) # Send data every minute, adjust as needed

except Exception as e:

print("An error occurred:", e)



**CONCLUSION:**

In this project, we've outlined the creation of an IoT-driven public transportation optimization system. It involves deploying IoT sensors in vehicles to capture real-time location and ridership data, which is then sent to a central transit platform via a Python script. Key components encompass sensor selection, hardware setup, software development, and security measures. The platform allows real-time monitoring, route optimization, and data analysis. This system enhances public transportation efficiency, improves service quality, and aids in data-informed decision-making, benefiting both passengers and operators. Proper customization is required to suit specific sensor and platform needs, ensuring seamless functionality.