# SB8040 - PROFESSIONAL READINESS FOR INNOVATION, EMPLOYABILITY AND ENTREPRENEURSHIP

# INTELLIGENT PEOPLE AND VEHICLE COUNTING SYSTEM FOR SECRETARIAT

**PROJECT REPORT SUBMITTED BY**


**VALLARASU N(513120106091)(TEAM LEAD)**

**SRINATH R(513120106083)**

**BHARATH R(513120106703)**

**PARTHIBAN R(513120106062)**


**TEAM ID: NM2023TMID22244**

**INDUSTRY MENTOR: Baradwaj , Dinesh, Hemanth**

**FACULTY MENTOR: R Kesavraj**

# TABLE OF CONTENTS

# 1.INTRODUCTION

## 1.1 Project overview:

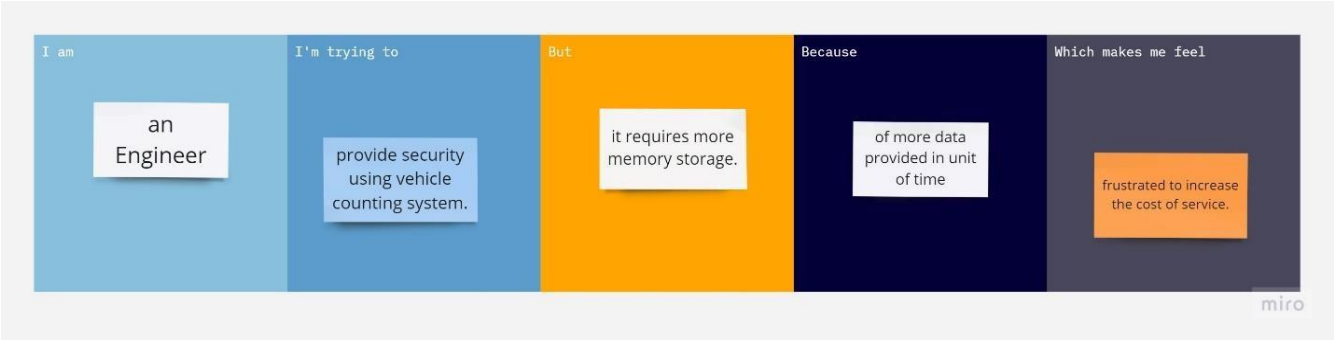
The Intelligent People and Vehicle Counting System for Secretariat using IoT is a project aimed at developing a smart and automated solution for monitoring and counting the number of people and vehicles entering and exiting a secretariat. The system utilizes Internet of Things (IoT) technologies to gather data, process it in real-time, and provide meaningful insights to efficiently manage the flow of people and vehicles within the premises. The Intelligent People and Vehicle Counting System for Secretariat using IoT offers an automated and intelligent solution for accurately monitoring and managing the flow of people and vehicles within a secretariat. By providing real-time data, analytics, and reporting, the system empowers administrators to optimize resource allocation, enhance security measures, and improve overall operational efficiency. Designed to accurately count and monitor the number of people and vehicles entering and exiting a secretariat. By utilizing IoT devices, sensors, and data analysis techniques, the system aims to provide real-time information for better resource management, security, and decision-making.

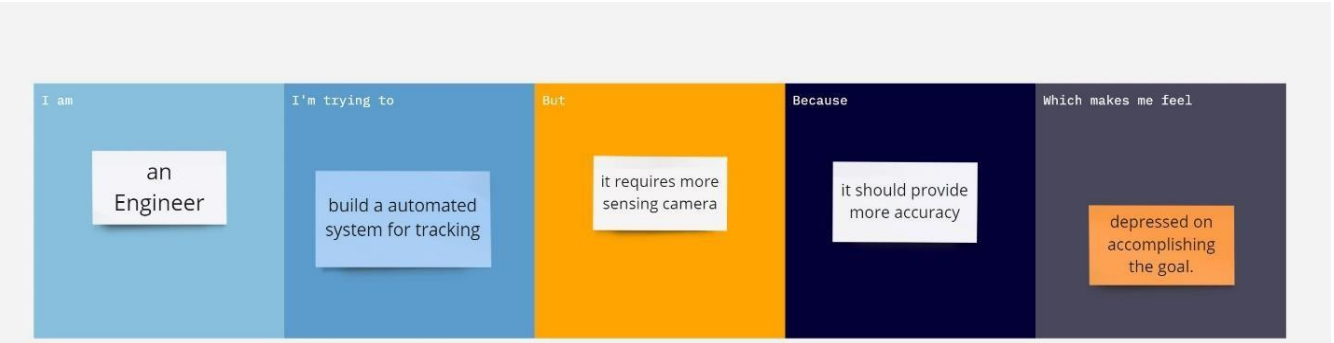
## 1.2 Purpose:

The purpose of the Intelligent People and Vehicle Counting System for Secretariat using IoT is to enhance the management, security, and efficiency of a secretariat by accurately counting and monitoring the number of people and vehicles entering and exiting the premises. The system serves the following key purposes.By providing real-time and accurate people and vehicle counts, the system helps in efficient resource management within the secretariat. The occupancy data can be used to optimize staff allocation, manage seating arrangements, and ensure appropriate utilization of facilities.The system contributes to enhancing security measures by detecting any unusual or suspicious patterns in people or vehicle counts. It can help identify potential security breaches, track movement patterns, and provide valuable insights for security personnel to take appropriate actions. The real-time monitoring and reporting capabilities of the system enable administrators to make informed decisions to optimize operational efficiency. By analyzing the count data, administrators can identify peak hours, manage crowd flow, and implement measures to improve visitor experience and reduce congestion

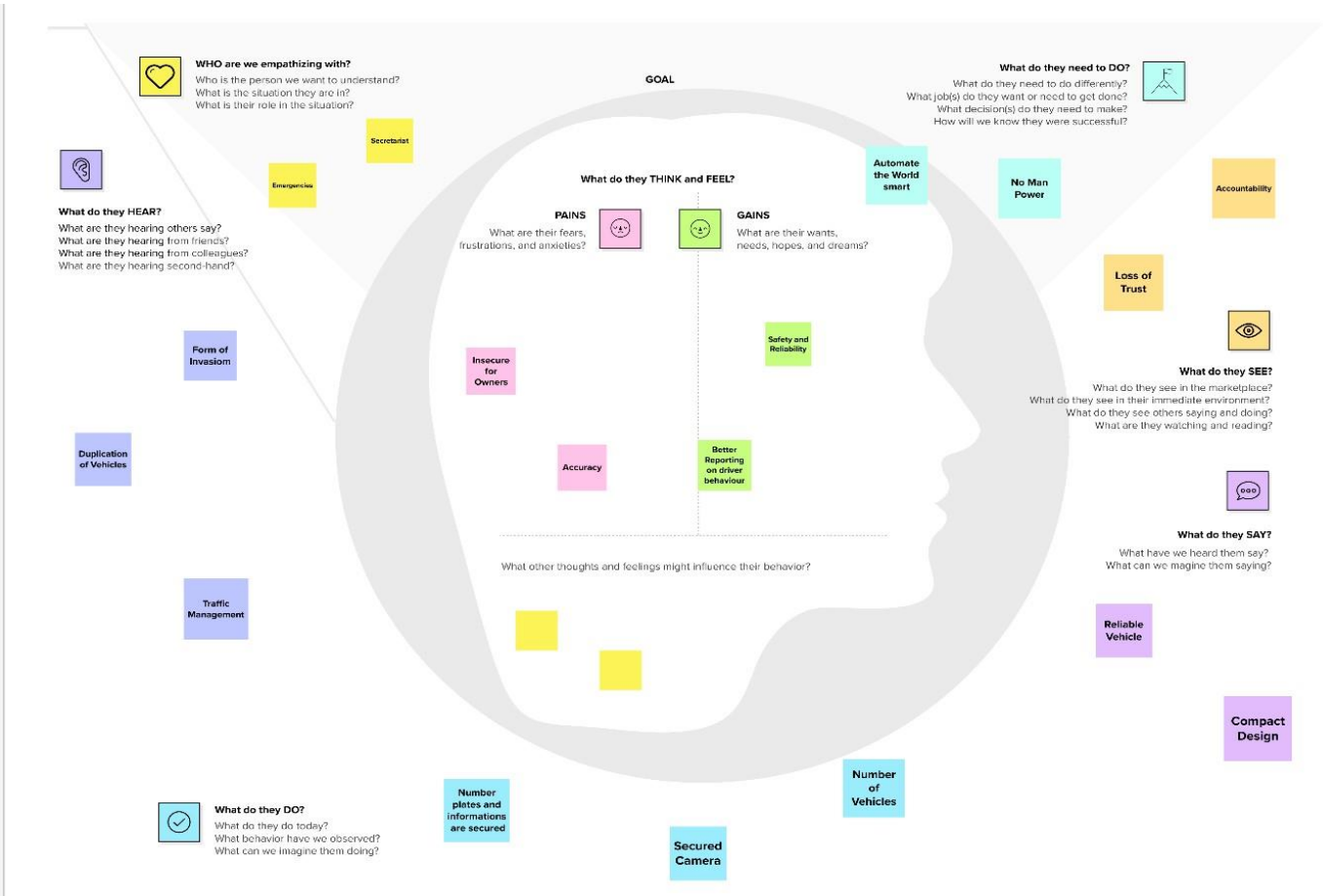
# 2.Ideation & Proposed solution

## 2.1 Problem Statement Definition:

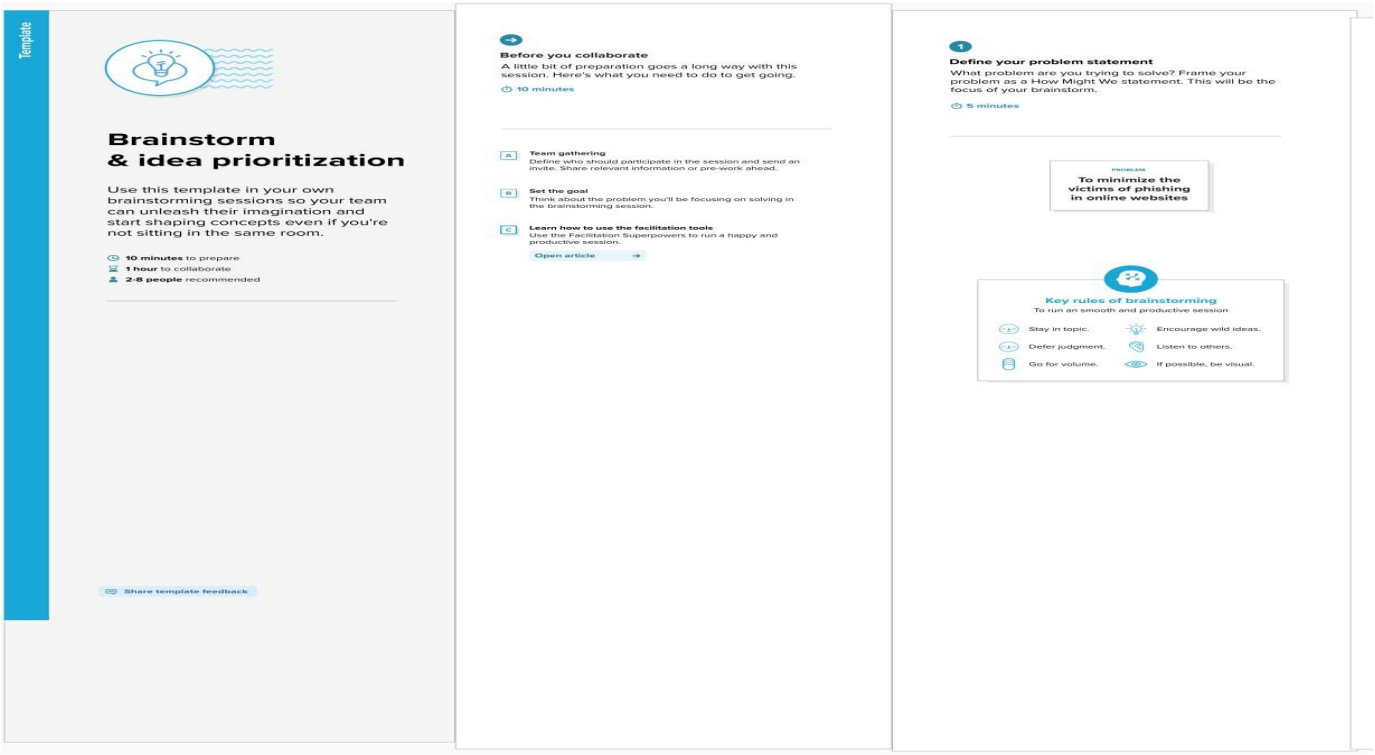


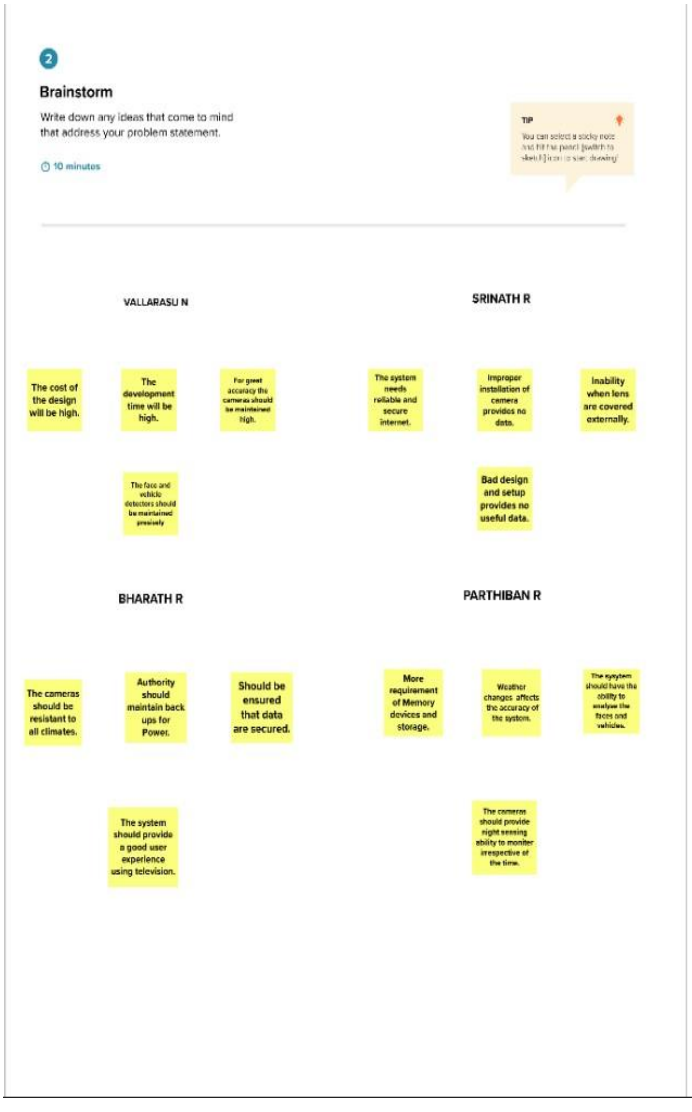



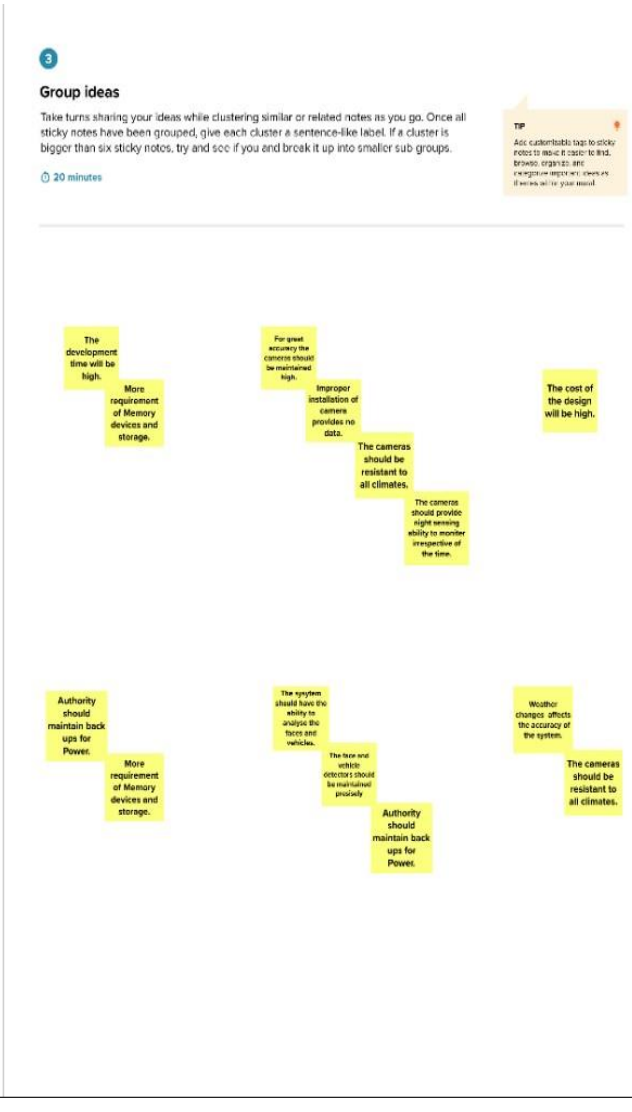

## 2.2 Empathy Map:

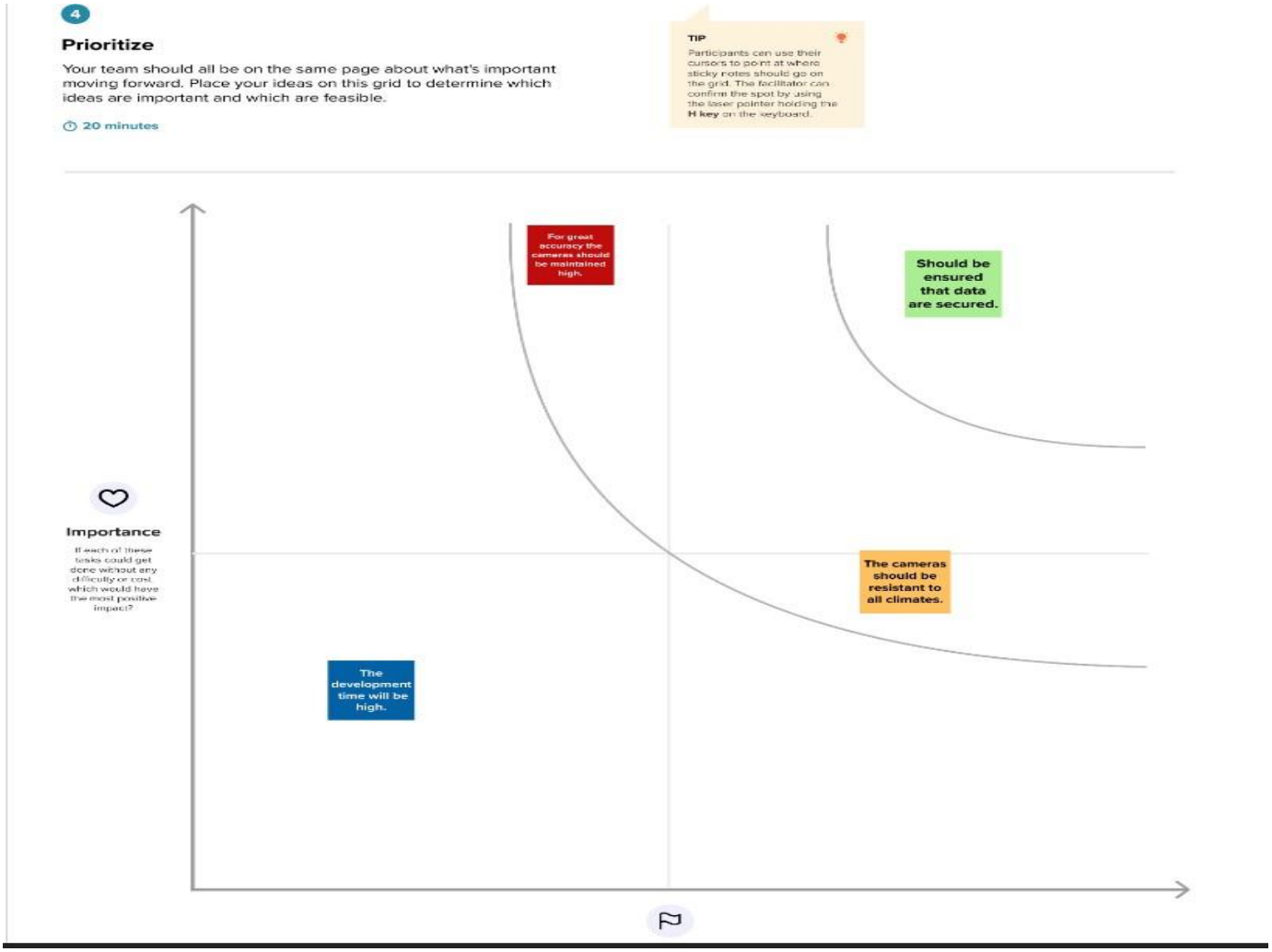


## 2.3 Ideation & Brainstorming:






## 2.4 Proposed Solution:

| S.No. | Parameter | Description |
|---|---|---|
| 1. | Problem Statement (Problem to be solved) | To develop a solution that can accurately count the number of vehicles passing through a certain point. This system should be able to collect real-time data about the number and types of vehicles passing through the area, and be able to transmit this data to a central server for further analysis and processing.The main goal of this system would be to improve traffic management, reduce congestion, and enhance the overall efficiency of the transportation network. |

| | | |
|---|---|---|
| 2. | Idea / Solution description | An idea is to deploy sensors such as radar, ultrasonic, or infrared at strategic locations on the road to detect and count vehicles passing through that point. These sensors can be connected to a microcontroller or single-board computer equipped with wireless connectivity modules such as Wi-Fi, Bluetooth to transmit data to a central server or cloud-based platform for further analysis and processing. |
| 3. | Novelty / Uniqueness | This system uses low-cost and energy-efficient sensors such as radar, ultrasonic, or infrared<br>The system uses a mesh network architecture, allowing each sensor node to communicate with nearby nodes and uses machine learning algorithms to filter out noise and eliminate false readings, making it more accurate and reliable.<br>A cloud-based platform for data storage, processing, and analysis, allowing traffic managers and planners to access real-time reports and insights about traffic flow. |
| 4. | Social Impact / Customer Satisfaction | The system can help reduce traffic congestion and improve overall traffic flow .By providing insights into traffic patterns and behaviors<br>The system can contribute to reducing greenhouse gas emissions by optimizing traffic flow and reducing congestion.The system can help reduce operational costs for traffic management authorities by automating the process of vehicle counting and data analysis, minimizing the need for manual labor and resources. |
| 5. | Business Model (Revenue Model) | Secretariat could pay a monthly or annual fee to access the vehicle counting system and receive regular updates and reports on traffic patterns and congestion hotspot .They could pay a fee each time they access the system or request a report on traffic patterns and congestion . The company could offer additional services, such as real-time traffic updates or personalized route optimization, and charge a fee for these services. |

| | | |
|---|---|---|
| 6. | Scalability of the Solution | The solution should have a modular architecture that allows for the addition of new sensors, nodes, and devices as needed, without requiring major modifications to the existing infrastructure. The solution can leverage cloudbased platforms and services that can scale up or down automatically and can use wireless connectivity which can support large-scale deployments and minimize the need for complex wiring or cabling ,which can process data locally at the sensor nodes or microcontrollers, reducing the load on the central server and improving overall scalability. |

# 3.Requirement Analysis:

## 3.1 Function Requirements:

Following are the functional requirements of the proposed solution.

| FR No. | Functional Requirement (Epic) | Sub Requirement (Story / Sub-Task) |
|---|---|---|
| FR-1 | People Monitoring | 1.Real-time tracking and monitoring of individuals within a designated area.<br>2.Identification of authorized personnel and detection of unauthorized individuals.<br>3.Tracking of individuals' movements and activities.<br>4.Generation of alerts or notifications for unusual or suspicious behavior.<br>5.Integration with existing access control systems for seamless identification and authentication. |
| FR-2 | Vehicle Monitoring | 1.Real-time tracking and monitoring of vehicles within a specific area.<br>2.Identification and authentication of authorized vehicles.<br>3.Monitoring of vehicle speed, direction, and location.<br>4.Detection of unauthorized vehicles or suspicious activities.<br>5.Integration with traffic management systems for congestion monitoring and traffic flow optimization. |
| FR-3 | Data Visualization and Reporting | 1.Collection and storage of monitoring data for analysis.<br>2.Generation of reports and statistics on people and vehicle movements.<br>3.Identification of patterns, trends, and anomalies in the collected data<br>4.Predictive analytics for identifying potential security risks or operational inefficiencies.<br>5.Visualization of data through dashboards and interactive interfaces. |
| FR-4 | Integration and Interoperability | 1.Seamless integration with existing security systems (e.g., CCTV cameras, access control systems).<br>2.Compatibility with various IoT devices, sensors, and communication protocols. |

| | | |
|---|---|---|
| | | 3.Integration with other operational systems such as traffic management, emergency response, or facility management systems.<br>4.Support for data sharing and interoperability between different monitoring systems. |
| FR-5 | Data Security and Privacy | 1.Establish appropriate access controls to ensure only authorized users can access and modify data<br>2.Implement measures to protect data privacy, such as anonymizing or aggregating data to prevent identification of individual vehicles or drivers<br>3.Develop a disaster recovery plan to ensure data can be recovered in the event of a system failure or other disaster. |

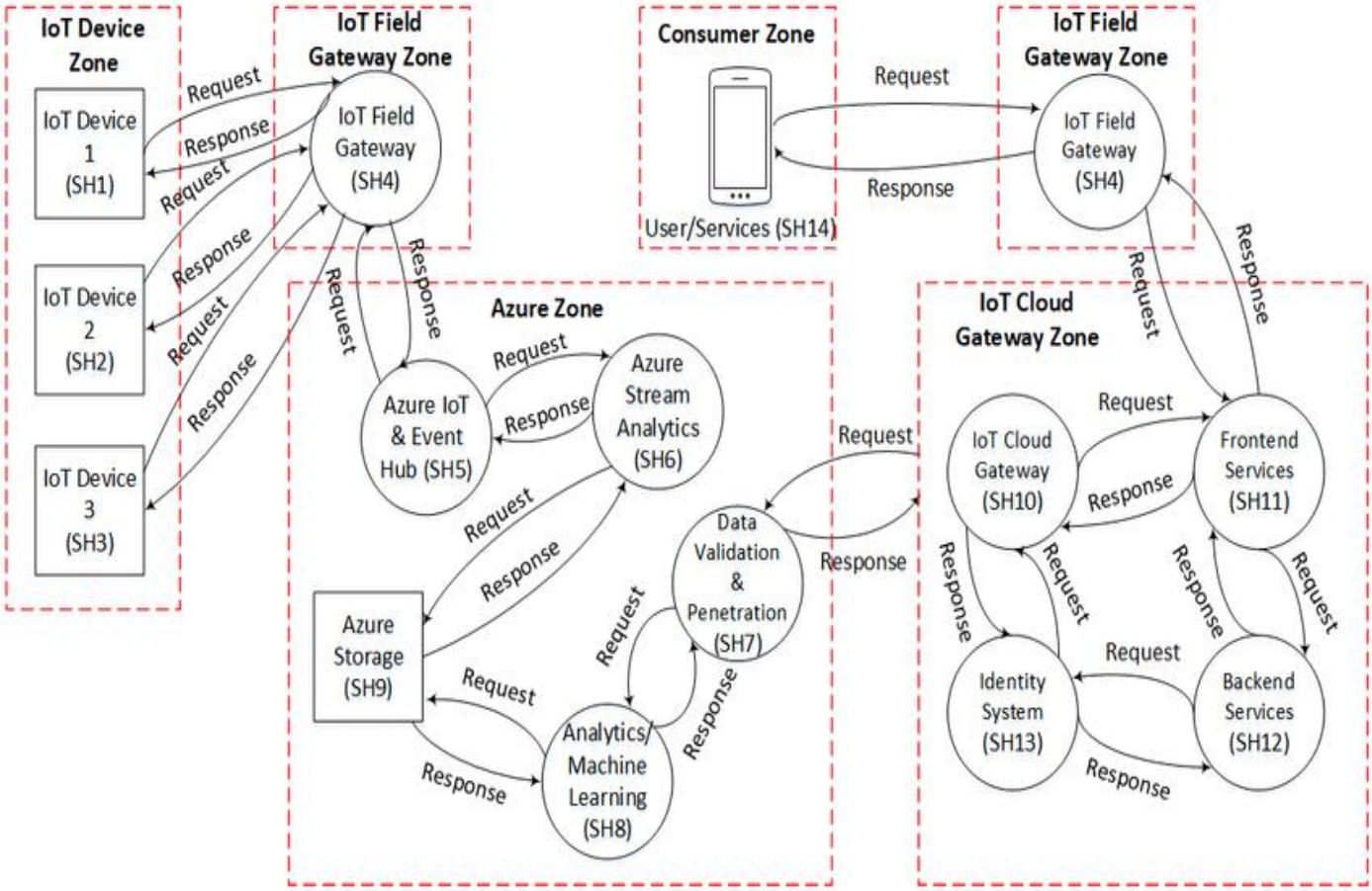
## 3.2 Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

| FR No. | Non-Functional Requirement | Description |
|---|---|---|
| NFR-1 | **Real-time Responsiveness** | Minimal latency in processing and analyzing monitoring data.Real-time alerts and notifications for immediate action. Efficient data processing and communication to ensure timely response to events. |
| NFR-2 | **Security** | Robust authentication and access control mechanisms to prevent unauthorized access. Encryption of data transmitted between devices and the monitoring system.Data integrity and confidentiality to protect sensitive information. Protection against cyber threats and vulnerabilities. |
| NFR-3 | **Reliability** | High availability of the monitoring system to ensure uninterrupted operation.<br>Redundancy and fault tolerance mechanisms to handle hardware or software failures.<br>Regular backups and data recovery mechanisms to prevent data loss. |
| NFR-4 | **User-Friendly Interface:** | Intuitive and user-friendly interfaces for system administrators and operators.<br>Easy configuration and management of monitored devices and system settings.<br>Customizable dashboards and reports to suit specific user requirements. |
| NFR-5 | **Power Efficiency:** | Optimization of power consumption for IoT devices to prolong battery life.<br>Energy-efficient data transmission protocols and algorithms. |
| NFR-6 | **Scalability** | Ability to handle a large number of monitored devices and data streams.<br>Support for the expansion of the monitoring system without significant performance degradation. Scalable storage and processing capabilities to accommodate increasing data volumes. |

# 4.Project Design

## 4.1 Data flow Diagram:




## 4.2 Solution & Technical Architecture:

## Table-1: Components & Technologies:

| S.No | Component | Description | Technology |
|---|---|---|---|
| 1. | IoT Sensors | Video cameras, infrared sensors, or lidar sensors capture data about people and vehicles. | MQTT (Message Queuing Telemetry Transport) or HTTP (Hypertext Transfer Protocol) can be used to transmit sensor data to the gateway. |

| | | | |
|---|---|---|---|
| 2. | IoT Gateway | Acts as a bridge between the sensors and the cloud platform. | MQTT or HTTP can be used for communication between the gateway and the sensors. |
| 3. | Cloud Infrastructure | Services provided by cloud providers like AWS,<br>Azure, or GCP for data processing, storage, and scalability. | IBM Cloud, Amazon Web Services<br>(AWS), Google Cloud Platform (GCP), Microsoft Azure, Apache Cassandra, Apache Hadoop, Apache Spark. |
| 4. | IoT Core Services | Platforms like AWS IoT Core, Azure IoT Hub, or Google Cloud IoT Core for managing and processing IoT device data. | Apache Kafka, RabbitMQ, or AWS IoT Core message broker for reliable and scalable message queuing. |
| 5. | Data Analytics | Tools like Apache Spark, TensorFlow, or PyTorch for performing AI/ML algorithms on the collected data. | Artificial Intelligence (AI) and Machine Learning (ML). |
| 6. | Computer Vision | TensorFlow, PyTorch, or Keras for developing and training AI models for people and vehicle detection. | OpenCV or specialized computer vision libraries for object detection, tracking, and counting. |
| 7. | Dashboarding tools | Dashboarding Tools: Grafana, Kibana, or Power BI for creating interactive dashboards and visualizations. | Development frameworks like React, Angular, or Flutter for building user interfaces to access and interact with the system. |
| 8. | System Integration | Provide integration capabilities and APIs (Application Programming Interfaces) to allow other systems or applications to access and utilize | APIs, microservices architecture, eventdriven architecture, message queuing. |
| | | the counting data. This can enable interoperability with existing infrastructure or integration with third-party applications. | |
| 9. | Security | Implement robust security measures to protect the data collected and transmitted throughout the system. Consider privacy regulations and anonymize personal information if required. | Data encryption, access controls, and authentication mechanisms are in place. |
| 10. | Infrastructure and Maintenance | Establish monitoring systems to track the health and performance of the sensors, gateways, cloud infrastructure, and analytics algorithms. | Proactive maintenance, remote firmware updates, and troubleshooting. |

## Table-2: Application Characteristics:

| S.No | Characteristics | Description | Technology |
|---|---|---|---|
| 1. | Open-Source Frameworks | The system should be built using open-source frameworks to reduce development costs and improve flexibility. It should use popular frameworks for web development, database management, and API development. | To achieve this, the system can use technologies such as React.js, Node.js, MongoDB, and Express.js. |
| 2. | Security Implementations | The system should have robust security measures to ensure the safety of sensitive customer data such as billing information. It should have authentication and authorization mechanisms, data encryption, and secure communication protocols | To achieve this, the system can use technologies such as SSL/TLS, OAuth, and IBM Cloud Identity and Access Management. |
| 3. | Scalable Architecture | The system should be able to handle a large volume of usage data from multiple water meters. It should be able to scale up or down based on the demand for water usage data | The system can use cloud technologies such as IBM Cloud, AWS, and Google Cloud Platform for scalability. |
| 4. | Availability | The system should have a highly available architecture to ensure uninterrupted service to customers. It should be designed to handle failures and recover quickly. | To achieve this, the system can use technologies such as IBM Cloud Availability Monitoring, Kubernetes, and Istio. |
| 5. | Performance | The system should be designed to handle a large volume of usage data and process it quickly to ensure fast and accurate billing. It should use technologies such as caching, load balancing, and clustering for improved performance | To achieve this, the system can use technologies such as Redis, NGINX, and Apache Kafka. |

## 4.3 User Stories

| User Type | Functional Requirement (Epic) | User Story Number | User Story / Task | Acceptance criteria | Priority | Team Member |
|---|---|---|---|---|---|---|
| Traffic engineer | Real-time tracking and monitoring of individuals within a designated area. | USN-1 | As a traffic engineer, I want the intelligent counting system to provide accurate and realtime data on people and vehicle counts, categorized by different lanes, so that I can optimize lane assignments and manage traffic congestion effectively. | The system should accurately count people and vehicles in each lane of a road or intersection. The data should be available in real-time and easily accessible for lanespecific analysis. | High | Vallarasu N |
| Parking facility manager | Identification of authorized personnel and detection of unauthorized individuals.. | USN-2 | As a parking facility manager, I want to implement the people and vehicle counting system to monitor parking lot occupancy, so that I can efficiently manage parking space allocation and inform customers about available parking spots. | The system should accurately count the number of vehicles entering and exiting the parking facility. The data should be processed in real-time and displayed on digital signage or a mobile application for customers to view parking availability. | Medium | Vallarasu N |
| City administrator | Generation of alerts or notifications for unusual or | USN-3 | As a city administrator, I want to utilize the people and vehicle counting system to identify peak traffic hours and popular routes, so that I can optimize public transportation schedules and allocate resources accordingly. | The system should provide data on people and vehicle counts during different time intervals and at various locations. | Low | Vallarasu N |

| | | | | | | |
|---|---|---|---|---|---|---|
| | suspicious behavior | | | The data should be aggregated and presented in a visual format that highlights peak traffic hours and popular commuting routes.. | | |
| Retail store manager | Integration with existing access control systems for seamless identification and authentication | USN-4 | As a retail store manager, I want to leverage the people counting feature of the system to measure customer conversion rates, so that I can evaluate the effectiveness of marketing campaigns and store layout modifications. | The system should accurately count the number of people entering the store and compare it to the number of purchases made.The data should be available for analysis on a daily, weekly, and monthly basis.The system should allow for the segmentation of data based on different areas or sections within the store. | High | Srinath R |
| **User Type** | **Functional Requirement (Epic)** | **User Story Number** | **User Story / Task** | **Acceptance criteria** | **Priority** | **Team Member** |
| City planner | Detection of unauthorized vehicles or suspicious activities | USN-5 | As a city planner, I want the people and vehicle counting system to provide data on the average speed of vehicles in different zones, so that I can identify areas with potential speeding issues and implement appropriate | The system should calculate and record the average speed of vehicles passing through specific zones. The data should be visualized on a map or dashboard, highlighting areas with high average speeds. | Medium | Srinath R |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | traffic calming measures. | | | |

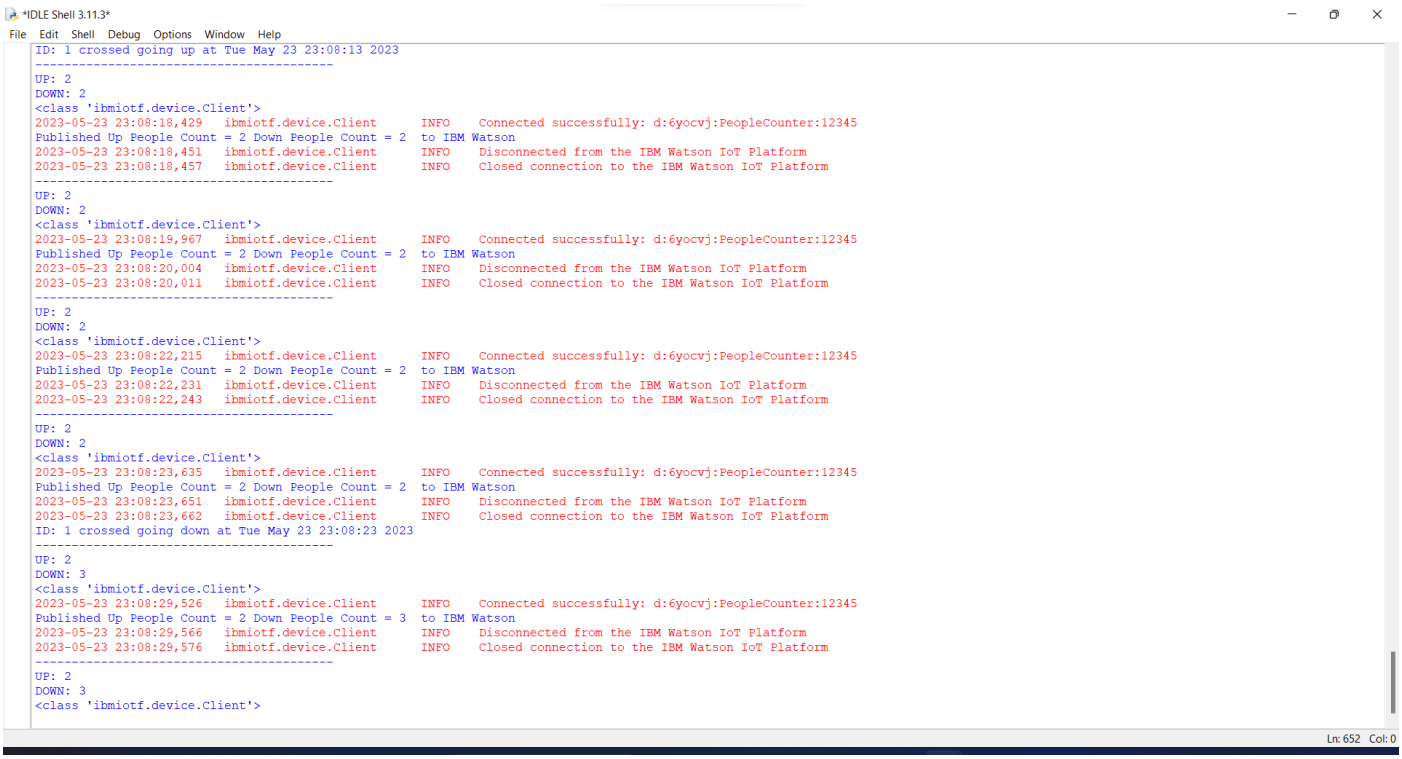
| | | | | | | |
|---|---|---|---|---|---|---|
| Public event organizer | .Identification of patterns, trends, and anomalies in the collected data | USN-6 | As a public event organizer, I want to utilize the people counting functionality of the system to estimate attendance and plan event logistics accordingly, ensuring a smooth and safe experience for attendees. | The system should provide accurate real-time data on the number of people entering the event venue. The data should be accessible to event organizers on their mobile devices or a dedicated monitoring system. | Low | Srinath R |
| Transportation researcher, | Predictive analytics for identifying potential security risks or operational inefficiencies. | USN-7 | As a transportation researcher, I want access to the people and vehicle counting data collected by the system to conduct in-depth analysis on traffic patterns, travel behavior, and urban mobility trends. | The system should accurately count the number of people entering and exiting the store, including repeat customers. The data should be available in real-time and accessible on a mobile device or dashboard. | High | Bharath R |
| Retail store owner | Visualization of data through dashboards and interactive interfaces | USN-8 | As a retail store owner, I want to install the people counting system to accurately measure foot traffic in my store and identify peak hours, so that I can schedule staff accordingly and optimize store layout for maximum efficiency. Acceptance Criteria: | Map should be easy to navigate and show routes for different bus and train lines. | Medium | Bharath R |

| | | | | | | |
|---|---|---|---|---|---|---|
| Building manager | Compatibility with various IoT devices, sensors, and communication protocols | USN-9 | As a building manager, I want to use the people counting system to track the occupancy of shared spaces, such as lobbies and elevators, to ensure compliance with social distancing measures and limit overcrowding. | The system should accurately count the number of people entering and exiting shared spaces. The data should be processed in real-time and displayed on digital signage or a mobile application for easy access. | Low | Bharath R |
| Transportation agency | Establish appropriate access controls to ensure only authorized users can access and modify data | USN-10 | As a transportation agency, I want to deploy the people and vehicle counting system to measure traffic volume and flow, so that I can identify bottlenecks and optimize traffic management. | The system should accurately count the number of people and vehicles passing through a specific location or route. The data should be available in real-time and displayed on a map or dashboard. The system should provide insights on traffic flow and potential bottlenecks. | High | Parthiban R |
| University administrator, | Implement measures to protect data privacy, such as anonymizing or aggregating data to prevent identification of individual | USN-11 | As a university administrator, I want to install the people counting system to monitor the occupancy of lecture halls and classrooms, so that I can optimize room allocation and ensure compliance with fire safety regulations. | The system should accurately count the number of people entering and exiting lecture halls and classrooms. The data should be available in real-time and accessible on a mobile device or dashboard. | Medium | Parthiban R |

| | | | | | | |
|---|---|---|---|---|---|---|
| | vehicles or drivers | | | | | |
| Museum curator | Develop a disaster recovery plan to ensure data can be recovered in the event of a system failure or other disaster. | USN-12 | As a museum curator, I want to install the people counting system to measure visitor traffic and engagement, so that I can optimize exhibit layouts and enhance the overall visitor experience. | The system should accurately count the number of people entering and exiting the museum and exhibit spaces. The data should be available in real-time and accessible on a mobile device or dashboard. | Low | Parthiban R |

# 5.Coding & Solutioning

## People and Vehicle Counting System:



# 6.Testing

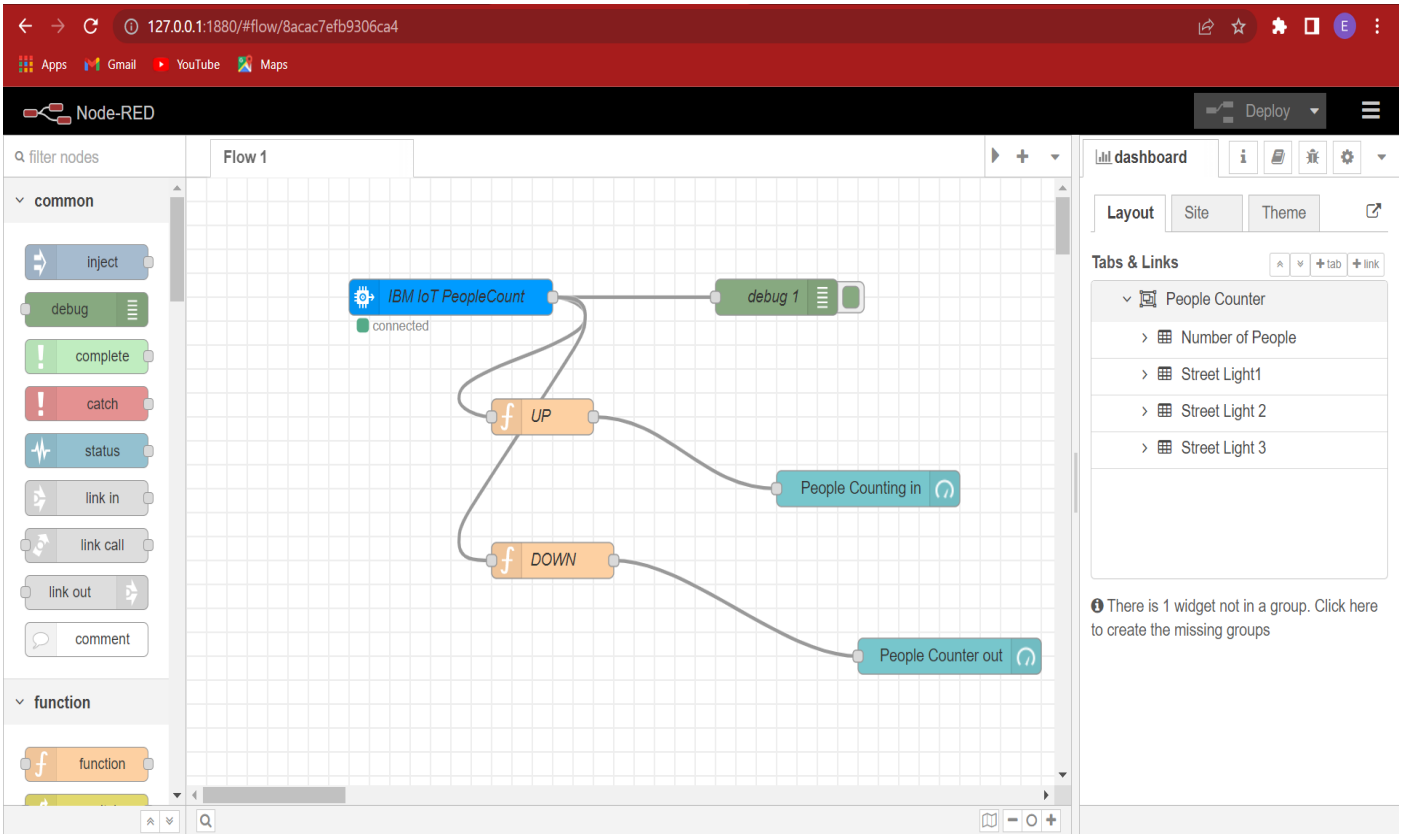
## 6.1 Python Code Screen Shot:

PersonCount.py - C:\Users\srina\OneDrive\Desktop\Project\PersonCount.py (3.11.3)

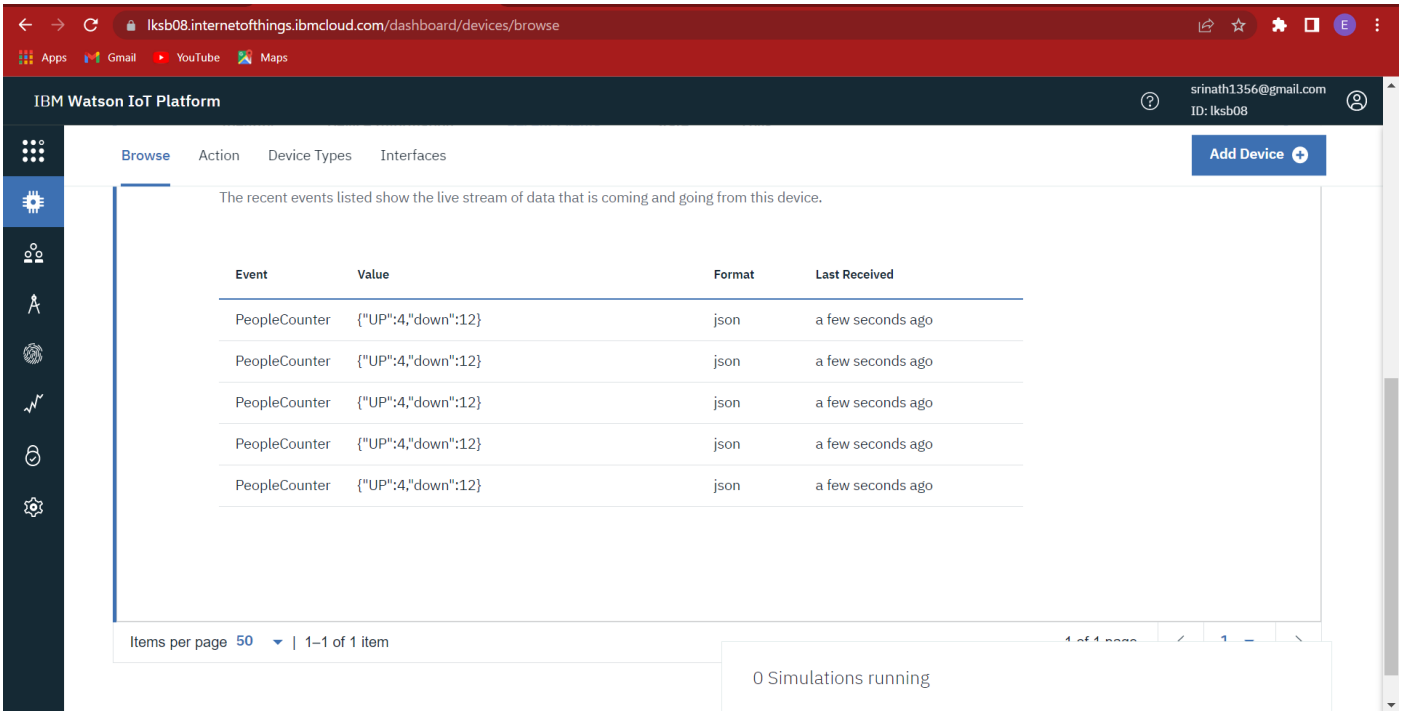
File Edit Format Run Options Window Help

```
##Contador de personas
##Federico Mejia
import numpy as np
import cv2
import Person
import time
import pyttsx3
import requests
import time
import sys
#import ibmiotf.application
import ibmiotf.device
import random
organization = "7ai91e"
deviceType = "device1"
deviceId = "12345"
authMethod = "token"
authToken = "12345678"
engine = pyttsx3.init()
engine.say('Hello')
engine.runAndWait()

#Contadores de entrada y salida
cnt_up   = 0
cnt_down = 0

#Fuente de video
#cap = cv2.VideoCapture(0)
#cap = cv2.VideoCapture('people.mp4')


#Propiedades del video
##cap.set(3,160) #Width
##cap.set(4,120) #Height

#Imprime las propiedades de captura a consola
cap = cv2.VideoCapture('people.mp4')
#cap = cv2.VideoCapture(0)
for i in range(19):
    print (i, cap.get(i))

w = cap.get(3)
h = cap.get(4)
frameArea = h*w
areaTH = frameArea/250
print ('Area Threshold', areaTH)
```

## 6.2 Node Red screen shot:



## 6.3 Cloud screen shot




# 7.Result

## 7.1.Performance Metrics:

# 8.Advantages & Disadvantages

### 8.1 Advantages:

Accuracy: The system provides accurate and reliable counting of people and vehicles entering and exiting the secretariat. By utilizing IoT devices and advanced counting algorithms, it minimizes errors and provides precise occupancy data.

Real-time Monitoring: The system offers real-time monitoring of people and vehicle counts. Authorized personnel can access up-to-date information on occupancy levels, allowing them to respond promptly to any overcrowding or security concerns.

Efficient Resource Management: The system enables better resource management within the secretariat. By having accurate occupancy data, administrators can allocate staff and facilities more effectively, optimizing utilization and avoiding underutilization or overutilization of resources.

Improved Security Measures: The system enhances security measures by detecting anomalies and unusual patterns in people and vehicle counts. It can raise alerts for potential security breaches or suspicious activities, enabling security personnel to take immediate action.

Data-driven Decision Making: The system provides valuable data insights for decision-making. Administrators can analyze historical data, identify trends, and make informed choices regarding operational strategies, resource allocation, and security enhancements.

Optimal Visitor Experience: With accurate people counting, the system helps manage crowd density and flow within the secretariat. This leads to a better visitor experience by reducing congestion, ensuring smoother movement, and maintaining a comfortable environment.

## 8.2 Disadvantages:

Cost of Implementation: Implementing an IoT-based counting system may involve significant upfront costs. The installation of IoT devices, sensors, network infrastructure, and cloud platform can require substantial investment, especially for large-scale secretariats.

Technical Challenges: Setting up and maintaining the IoT infrastructure requires technical expertise. It may involve challenges such as connectivity issues, device compatibility, firmware updates, and troubleshooting technical glitches. Adequate training and technical support may be needed to address these challenges.

Data Privacy and Security Risks: IoT systems involve the collection and transmission of sensitive data, including people and vehicle counts. There is a risk of data breaches, unauthorized access, or misuse of the collected data. Robust security measures, encryption protocols, and adherence to privacy regulations are crucial to mitigate these risks.

Reliance on Internet Connectivity: The system's functionality relies on stable internet connectivity. If there are network disruptions or outages, it can impact real-time monitoring, data transmission, and system reliability. Backup connectivity options or contingency plans may be necessary to ensure uninterrupted operation.

Potential for Inaccuracies: While IoT counting systems strive for accuracy, there is still a possibility of errors in counting. Factors such as sensor malfunction, occlusions, or abnormal

movement patterns may lead to incorrect counts. Regular maintenance and calibration of sensors are essential to minimize inaccuracies.

Integration Challenges: Integrating the IoT counting system with existing infrastructure and systems within the secretariat can be complex. Compatibility issues, data synchronization, and interoperability challenges may arise during integration. Thorough planning and collaboration with relevant stakeholders are crucial for successful integration.

# 9.CONCLUSION

The Intelligent People and Vehicle Counting System for Secretariat Using IoT offers significant benefits in terms of accurate counting, real-time monitoring, efficient resource management, improved security measures, data-driven decision-making, enhanced visitor experience, remote access, scalability, and cost-effectiveness. However, it is essential to consider potential disadvantages such as implementation costs, technical challenges, data privacy and security risks, potential inaccuracies, integration complexities, ethical considerations, limited contextual information, and maintenance requirements.

# 10.Future scope:

Advanced Analytics: The system can evolve to incorporate more advanced analytics techniques, such as machine learning and predictive modeling. By leveraging historical count data, it can generate predictive insights, forecast future visitor patterns, and provide proactive recommendations for resource allocation and security measures.

Integration with Smart Building Systems: The project can be integrated with smart building systems to create a more comprehensive and interconnected infrastructure. This integration can enable automated adjustments in lighting, temperature, and other environmental factors based on occupancy levels, improving energy efficiency and comfort within the secretariat.

Facial Recognition and Identification: By combining people counting with facial recognition technology, the system can provide additional capabilities such as identifying and tracking 24 specific individuals or detecting unauthorized access. This integration can enhance security measures and enable personalized services for authorized personnel.

Crowd Management and Routing: The system can expand to include crowd management and routing capabilities. By analyzing real-time occupancy data and utilizing digital signage or mobile applications, it can guide visitors to less crowded areas, optimize traffic flow, and prevent congestion within the secretariat.

Data-driven Decision Support: The project's future scope includes developing comprehensive dashboards and reporting tools that provide actionable insights based on people and vehicle count data. Decision-makers can access user-friendly interfaces and visualizations to make informed decisions regarding space utilization, staffing requirements, and security strategies. Integration with IoT Ecosystem: As the IoT ecosystem continues to expand, the system can integrate with a wider range of IoT devices and sensors. This integration can enable seamless data sharing between different systems, enhancing overall operational efficiency, security, and visitor experience.

# 11.Appendix:

## 11.1Github link:

https://github.com/naanmudhalvan-SI/IBM--11409-1682326281/tree/main

## 11.2 Code Link:

https://github.com/naanmudhalvan-SI/IBM--11409-1682326281/blob/main/Final%20Deliverables/Code.pdf

## 11.3 Demo link

https://drive.google.com/file/d/1Lvglfit5gKwzw8ARAh5QFlW4RZA-gNTq/view?usp=sharing

## 11.4 Source code:

```
##Contador de personas
##Federico Mejia
import numpy as np
import cv2
import Person
import time
import pyttsx3
import requests
import time
import sys
#import ibmiotf.application
import ibmiotf.device
import random
organization = "lksb08"
deviceType = "sri"
deviceId = "12345"
authMethod = "token"
authToken = "12345678"
engine = pyttsx3.init()
engine.say('Hello')
```

```
engine.runAndWait()

#Contadores de entrada y salida
cnt_up   = 0
cnt_down = 0

#Fuente de video
#cap = cv2.VideoCapture(0)
#cap = cv2.VideoCapture('people.mp4')



#Propiedades del video
##cap.set(3,160) #Width
##cap.set(4,120) #Height

#Imprime las propiedades de captura a consola
cap = cv2.VideoCapture('people.mp4')
#cap = cv2.VideoCapture(0)
for i in range(19):
   print (i, cap.get(i))

w = cap.get(3)
h = cap.get(4)
frameArea = h*w
areaTH = frameArea/250
print ('Area Threshold', areaTH)

#Lineas de entrada/salida
line_up = int(2*(h/5))
line_down   = int(3*(h/5))

up_limit =   int(1*(h/5))
down_limit = int(4*(h/5))
```

```
print ("Red line y:",str(line_down))
print ("Blue line y:", str(line_up))
line_down_color = (255,0,0)
line_up_color = (0,0,255)
pt1 =  [0, line_down];
pt2 =  [w, line_down];
pts_L1 = np.array([pt1,pt2], np.int32)
pts_L1 = pts_L1.reshape((-1,1,2))
pt3 =  [0, line_up];
pt4 =  [w, line_up];
pts_L2 = np.array([pt3,pt4], np.int32)
pts_L2 = pts_L2.reshape((-1,1,2))

pt5 =  [0, up_limit];
pt6 =  [w, up_limit];
pts_L3 = np.array([pt5,pt6], np.int32)
pts_L3 = pts_L3.reshape((-1,1,2))
pt7 =  [0, down_limit];
pt8 =  [w, down_limit];
pts_L4 = np.array([pt7,pt8], np.int32)
pts_L4 = pts_L4.reshape((-1,1,2))

#Substractor de fondo
fgbg = cv2.createBackgroundSubtractorMOG2(detectShadows = True)

#Elementos estructurantes para filtros morfoogicos
kernelOp = np.ones((3,3),np.uint8)
kernelOp2 = np.ones((5,5),np.uint8)
kernelCl = np.ones((11,11),np.uint8)

#Variables
font = cv2.FONT_HERSHEY_SIMPLEX
```

```
persons = []
max_p_age = 5
pid = 1
def ibmwork(cnt_up,cnt_down,deviceCli):
   data = { 'UP' : cnt_up, 'down': cnt_down}
      #print data
   def myOnPublishCallback():
      print ("Published Up People Count = %s" % str(cnt_up), "Down People Count = %s " % str(cnt_down), "to IBM Watson")

   success = deviceCli.publishEvent("PeopleCounter", "json", data, qos=0, on_publish=myOnPublishCallback)
   if not success:
      print("Not connected to IoTF")



   deviceCli.disconnect()



def ibmstart(cnt_up,cnt_down):


   try:
         deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": authMethod, "auth-token": authToken}
         deviceCli = ibmiotf.device.Client(deviceOptions)
         print(type(deviceCli))
         #.....


   except Exception as e:
         print("Caught exception connecting device: %s" % str(e))
         sys.exit()
   deviceCli.connect()
   ibmwork(cnt_up,cnt_down,deviceCli)
```

```
while(cap.isOpened()):
##for image in camera.capture_continuous(rawCapture, format="bgr", use_video_port=True):
   #Lee una imagen de la fuente de video
   ret, frame = cap.read()
##    frame = image.array

   for i in persons:
      i.age_one() #age every person one frame
   #########################
   #   PRE-PROCESAMIENTO   #
   #########################

   #Aplica substraccion de fondo
   fgmask = fgbg.apply(frame)
   fgmask2 = fgbg.apply(frame)

   #Binariazcion para eliminar sombras (color gris)
   try:
      ret,imBin= cv2.threshold(fgmask,200,255,cv2.THRESH_BINARY)
      ret,imBin2 = cv2.threshold(fgmask2,200,255,cv2.THRESH_BINARY)
      #Opening (erode->dilate) para quitar ruido.
      mask = cv2.morphologyEx(imBin, cv2.MORPH_OPEN, kernelOp)
      mask2 = cv2.morphologyEx(imBin2, cv2.MORPH_OPEN, kernelOp)
      #Closing (dilate -> erode) para juntar regiones blancas.
      mask =  cv2.morphologyEx(mask , cv2.MORPH_CLOSE, kernelCl)
      mask2 = cv2.morphologyEx(mask2, cv2.MORPH_CLOSE, kernelCl)
   except:
      print('EOF')
      print ('UP:',cnt_up)
      print ('DOWN:',cnt_down)
      break
```

```
    ##################
    #   CONTORNOS   #
    ##################

    # RETR_EXTERNAL returns only extreme outer flags. All child contours are left behind.
    contours0, hierarchy = cv2.findContours(mask2,cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE)
    for cnt in contours0:
        area = cv2.contourArea(cnt)
        if area > areaTH:
            ##################
            #   TRACKING    #
            ##################

            #Falta agregar condiciones para multipersonas, salidas y entradas de pantalla.

            M = cv2.moments(cnt)
            cx = int(M['m10']/M['m00'])
            cy = int(M['m01']/M['m00'])
            x,y,w,h = cv2.boundingRect(cnt)

            new = True
            if cy in range(up_limit,down_limit):
                for i in persons:
                    if abs(cx-i.getX()) <= w and abs(cy-i.getY()) <= h:
                        # el objeto esta cerca de uno que ya se detecto antes
                        new = False
                        i.updateCoords(cx,cy)   #actualiza coordenadas en el objeto and resets age
                        if i.going_UP(line_down,line_up) == True:
                            cnt_up += 1;
                            print ("ID:",i.getId(),'crossed going up at',time.strftime("%c"))
                            engine.say('A Person is Going UP ')
                            engine.runAndWait()
```

```
            elif i.going_DOWN(line_down,line_up) == True:
                cnt_down += 1;
                print ("ID:",i.getId(),'crossed going down at',time.strftime("%c"))
                engine.say('A Person is Going Down')
                engine.runAndWait()
            break
        if i.getState() == '1':
            if i.getDir() == 'down' and i.getY() > down_limit:
                i.setDone()
            elif i.getDir() == 'up' and i.getY() < up_limit:
                i.setDone()
        if i.timedOut():
            #sacar i de la lista persons
            index = persons.index(i)
            persons.pop(index)
            del i     #liberar la memoria de i
    if new == True:
        p = Person.MyPerson(pid,cx,cy, max_p_age)
        persons.append(p)
        pid += 1
    ##################
    #   DIBUJOS     #
    ##################
    cv2.circle(frame,(cx,cy), 5, (0,0,255), -1)
    img = cv2.rectangle(frame,(x,y),(x+w,y+h),(0,255,0),2)
    #cv2.drawContours(frame, cnt, -1, (0,255,0), 3)

#END for cnt in contours0

#########################
# DIBUJAR TRAYECTORIAS  #
#########################
for i in persons:
```

```
##       if len(i.getTracks()) >= 2:
##           pts = np.array(i.getTracks(), np.int32)
##           pts = pts.reshape((-1,1,2))
##           frame = cv2.polylines(frame,[pts],False,i.getRGB())
##       if i.getId() == 9:
##           print str(i.getX()), ',', str(i.getY())
        cv2.putText(frame, str(i.getId()),(i.getX(),i.getY()),font,0.3,i.getRGB(),1,cv2.LINE_AA)


    ##################
    #   IMAGANES    #
    ##################
    str_up = 'UP: '+ str(cnt_up)
    str_down = 'DOWN: '+ str(cnt_down)
    print('----------------------------------------')
    print ('UP:',cnt_up)
    print ('DOWN:',cnt_down)


    #r1                                                                  =
requests.get('https://api.thingspeak.com/update?api_key=4BGMGGBRLQM3VRHO&field1='+st
r(cnt_up))
   #                                                  r2                 =
requests.get('https://api.thingspeak.com/update?api_key=4BGMGGBRLQM3VRHO&field2='+st
r(cnt_down))
   # print(r1.status_code)
   # print(r2.status_code)
    frame = cv2.polylines(frame,[pts_L1],False,line_down_color,thickness=2)
    frame = cv2.polylines(frame,[pts_L2],False,line_up_color,thickness=2)
    frame = cv2.polylines(frame,[pts_L3],False,(255,255,255),thickness=1)
    frame = cv2.polylines(frame,[pts_L4],False,(255,255,255),thickness=1)
    cv2.putText(frame, str_up ,(10,40),font,0.5,(255,255,255),2,cv2.LINE_AA)
    cv2.putText(frame, str_up ,(10,40),font,0.5,(0,0,255),1,cv2.LINE_AA)
    cv2.putText(frame, str_down ,(10,90),font,0.5,(255,255,255),2,cv2.LINE_AA)
    cv2.putText(frame, str_down ,(10,90),font,0.5,(255,0,0),1,cv2.LINE_AA)
```

```
    cv2.imshow('Frame',frame)
    #cv2.imshow('Mask',mask)

    #preisonar ESC para salir

    ibmstart(cnt_up,cnt_down)




# Disconnect the device and application from the cloud

    k = cv2.waitKey(30) & 0xff
    if k == 27:
        break
#END while(cap.isOpened())

##################
#   LIMPIEZA    #
##################

cap.release()
cv2.destroyAllWindows()
```