PUBLIC TRANSPORT OPTIMIZATION

For the Optimization of public transport for the safety and comfortable journey for the passengers, the following devices are required.

Reverse Camera :

Set at the back of the Vehicle or Public bus or taxi or any public transport. The camera is always turned on for the drivers to take reverse or to use breaks safely.

Smoke Detector :

The Smoke detector is connected to the internet through a stable Wi-Fi to read the level of Smoke content. The Water from a small tank at the top of Bus is released to get sprinkled inside the Bus. If the level exceeds all the normal value or the limit set by officials, the Detector sends information with Location to the Fire Department.

Alarm with Siren System :

The Alarm is under the control of either the Driver or Conductor in Bus, the Driver in the Taxi. But its not fully manual, as if there is a severe emergency situation (an Accident, Medical Emergency and more …) and if no actions are taken, then the Alarm is issued to call all the Medical, Fire, Police Departments with the photos captured through cameras, audio recorded by microphone near the camera, location and other information is sent to each department separately. Further, for an immediate help from nearby people Alarm Siren is started loudly.

Camera with Microphone :

In any emergency situations, in order to capture photos, videos and record audio, two cameras are to be installed in the Bus, one at the front near the driver's seat, and the other at back.

In order to make it possible, the components required are as follows,

IoT Device :

Smoke Detector - High quality smoke sensor to read smoke levels, Micro-controller (ESP8266 OR ESP32) with Wi-Fi capabilities, Water sprinkling valves, pipes, tanks and more.

Camera with Microphone - High quality Microphones, Cameras to capture everything clearly, Integration with Wi-Fi, Power supply from the Transport vehicle, Motion and Audio detecting sensors, and more.

Reverse Camera - A High quality camera that is connected to the driver, Power supply is a separate battery in the vehicle.

Alarm with Siren System - A Micro-controller embedded with all device to suppor the internet, Power supply from the Battery, A loud attention-grabbing Siren, Integration with location detector, Data transfer protocol to send data to all service departments.

Internet Connection :

All the IoT device must be connected with a stable Wi-Fi connection in the Transport vehicle to support all the devices.

IoT Platform :

The IoT platforms like Azure, Google, AWS IoT Clouds can be used to integrate with devices to send and receive the data and store them each time when it is issued in the servers at the Service departments.

Python Scripting :

The Python scripts are made separately for each devices in the vehicle that are integrated with cameras, microphones, sensors, and all the other components.
Further, a python script is needed to integrate all the devices together to make it a single unit to send all the information as a single message.

The Algorithm or Steps :

Step - 1 : For Python scripting, all the important libraries required are imported to work with all the sensors, cameras and other devices in the Vehicles.

Step - 2 : All the components taken are of good quality that is tested by many experts before the installation in the vehicle.

Step - 3 : A perfect platform is selected to transfer of data from the vehicle to the service departments without interfering in other signals.

Step - 4 : A Small water tank that must be always full to use it for the emergency situations. It is loaded to the top of the Transport vehicle and it must be connected with the Smoke detector.

Step - 5 : Each time before the start of the journey or travel it the responsibility of the Drivers and Conductors to check if all the devices are at working conditions.

Step - 6 : GPS must be set in the vehicle to detect location.

Step - 8 : A Siren is set at the head of the vehicle to sound loud to seek attention of the people nearby when the vehicle is met with an accident.

Step - 9 : The Wire connections must be properly set at the Reverse Camera to show the full view to the driver while taking reverse to avoid accidents or damages.

Step - 10 : To Collect all the data during a serious or emergency situation as single package and sends it to separate service departments,

(i) Capture all the photos and videos with audio from both cameras inside the vehicle.

(ii) Locate the location using GPS.

(iii) Read the Air components using the Smoke detectors.

(iv) Each data is in separate files and they are further combined as a single package.

(v) Uses the Internet to send the package to each departments.

(vi) The Service departments identifies all the details and takes equipments required to the spot to save lives.

(vii) If the Package is found to be fake situation or the data collected are fake then the components are reviewed before the next journey of the vehicle. Or if it was a Human error, then a strict action is taken on the particular person.

The Python Programs :

1 - Cameras and Microphones to auto capture

```
import cv2  # OpenCV for image and video processing
import sounddevice as sd  # for audio recording
import datetime

# Function to capture photos and record video with audio
def capture_photos_and_videos():
    # Initialize the camera
    camera = cv2.VideoCapture(0)  # Use the appropriate camera device index

    while True:
        ret, frame = camera.read()
        if not ret:
            break

        # Capture photos
        photo_filename = f"emergency_photo_{datetime.datetime.now().strftime('%Y-%m-%d_%H-
%M-%S')}.jpg"
        cv2.imwrite(photo_filename, frame)

        # Record video with audio
        video_filename = f"emergency_video_{datetime.datetime.now().strftime('%Y-%m-%d_%H-
%M-%S')}.avi"
        fourcc = cv2.VideoWriter_fourcc(*'XVID')
        video_writer = cv2.VideoWriter(video_filename, fourcc, 20.0, (640, 480))
        audio_data = sd.rec(int(10 * 44100), channels=2)
        sd.wait()
        video_writer.write(frame)
        video_writer.release()

        print(f"Photo captured: {photo_filename}")
        print(f"Video recorded: {video_filename}")

    camera.release()

# Call the function to start capturing photos and videos with audio
capture_photos_and_videos()
```

2 - Smoke Detector

```
import RPi.GPIO as GPIO
import time

# GPIO pins
SMOKE_SENSOR_PIN = 17
SPRINKLER_PIN = 18

# Initialize GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setup(SMOKE_SENSOR_PIN, GPIO.IN)
GPIO.setup(SPRINKLER_PIN, GPIO.OUT)

# Function to simulate smoke detection
def detect_smoke():
    return GPIO.input(SMOKE_SENSOR_PIN) == GPIO.HIGH

# Function to activate the sprinkler
def activate_sprinkler():
    print("Smoke detected! Activating sprinkler.")
    GPIO.output(SPRINKLER_PIN, GPIO.HIGH)
    time.sleep(5)  # Simulate the sprinkler running for 5 seconds
    GPIO.output(SPRINKLER_PIN, GPIO.LOW)
    print("Sprinkler deactivated.")

try:
    while True:
        if detect_smoke():
            activate_sprinkler()
        time.sleep(1)  # Check for smoke every 1 second

except KeyboardInterrupt:
    GPIO.cleanup()
```

3 - Send Emergency message package and start the loud siren

```
from twilio.rest import Client

# Your Twilio Account SID and Auth Token
account_sid = 'your_account_sid'
auth_token = 'your_auth_token'

# Create a Twilio client
client = Client(account_sid, auth_token)

# Function to send an emergency alert
def send_emergency_alert(location, message):
    recipients = ['police_number', 'fire_number', 'medical_number']  # Replace with actual
numbers

    for recipient in recipients:
        message = client.messages.create(
            body=f"Emergency at {location}: {message}",
            from_="your_twilio_phone_number",
            to=recipient
        )
        print(f"Emergency alert sent to {recipient}: {message.sid}")

# In case of an emergency, call the send_emergency_alert function
location = "Your Location"
emergency_message = "Emergency situation detected."
send_emergency_alert(location, emergency_message)

# Optionally, sound a loud siren using a hardware interface.
```