**1.INTRODUCTION :-**

**a.Overview:**

To avoid needless mistakes when planning a delivery, it’s important to first distinguish which products are perishable.Most packaged foods and products come with guidelines on how they should be stored, and it’s always worth checking that your transportation or delivery conditions meet these.

Perishable goods are foodstuffs and other products that have short shelf-lives and tend to spoil, decay or deteriorate unless kept under specific temperature and humidity conditions.

For example, food products such as meat, dairy, poultry, fish, vegetables, fruit and pre-cooked foods must be kept at low temperatures, otherwise, mould or harmful bacteria can develop as well as harming the odour, taste and/or visual appeal of the food product, bacterial growth and decay eventually make it unsafe to eat.Other perishable goods include certain chemicals, pharmaceuticals, vaccines, and even flowers.

These products are susceptible to severe deterioration during transportation due to their delicate or volatile nature . Therefore , Perishable goods are problematic to transport over long distances.

**b.Purpose:**

Since perishable goods are known to deteriorate in bad conditions especially during their transport , there is a necessity for continuous monitoring of these goods . The purpose of ***Remote monitoring of perishable goods*** project is to provide access to the state of the perishable goods at any time so that the goods will be in a good condition until they reach their destination.

**2.LITERATURE SURVEY:-**

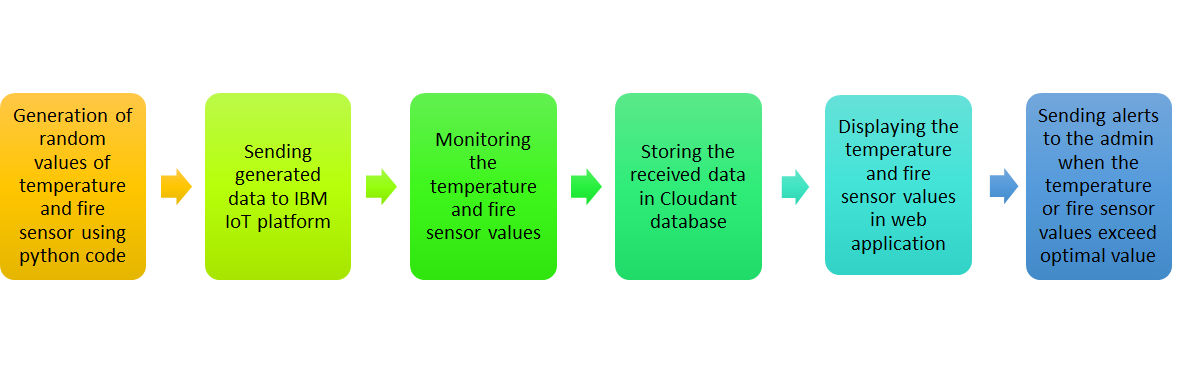
**a.Existing problem:**

The transportation of perishable goods is a hectic task which might result in loss of revenue if the goods are deteriorated . The main problem arises when these goods are to be transported across long distances . The weather conditions change throughout the time so it is important to store the perishable goods in a safe environment where all the parameters are at check .

**b.Proposed solution:**

To avoid the above problem , the ***Remote monitoring of perishable goods*** project is implemented . Some wireless sensors are placed in the container with perishable goods which can send the data regarding the goods such as temperature , humidity or fire sensor values . Through this , the person managing the transport of the perishable goods can continuously monitor the state of the goods.

**3.THEORETICAL ANALYSIS:-**

**a.Block diagram:**

**b.Hardware/Software designing:**

We follow a step-by-step procedure to set up all the interfaces required for our project and develop the code in python to send random sensor data to the cloud. The following software is required:

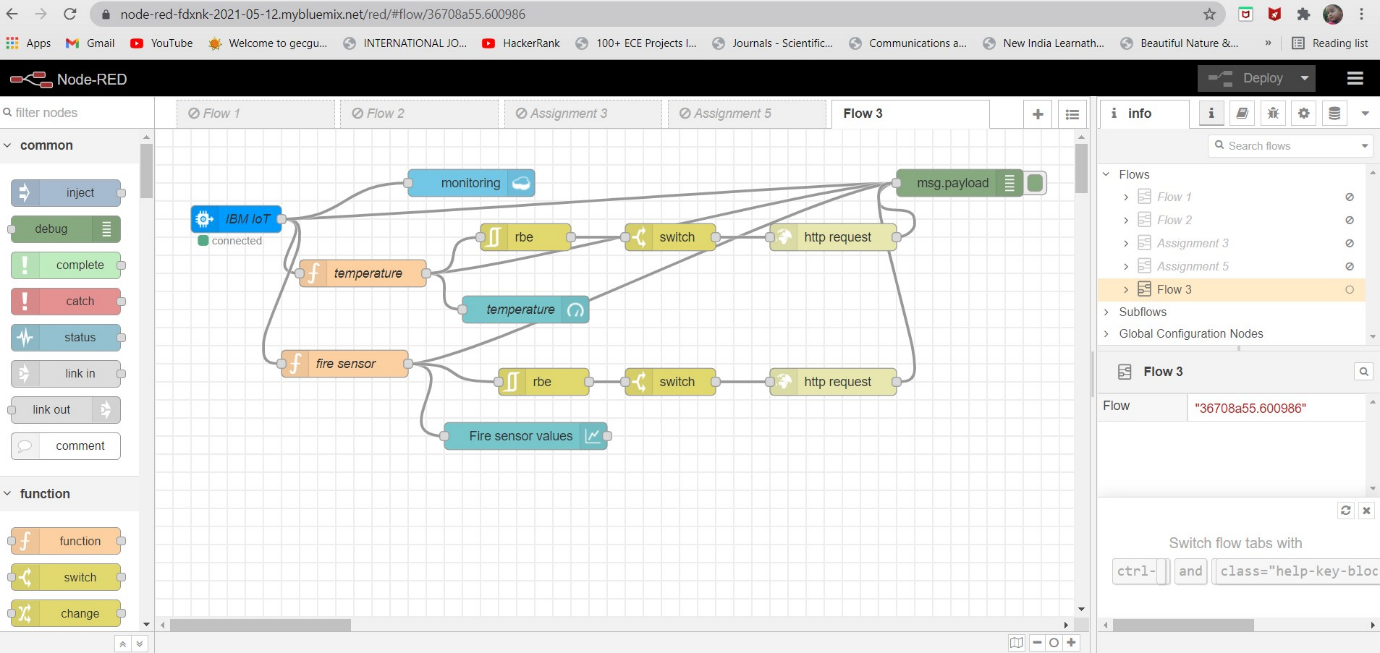
1. Python Idle (with specified packages installed)
2. IBM cloud
3. Node Red service

**4.Experimental Investigations:-**

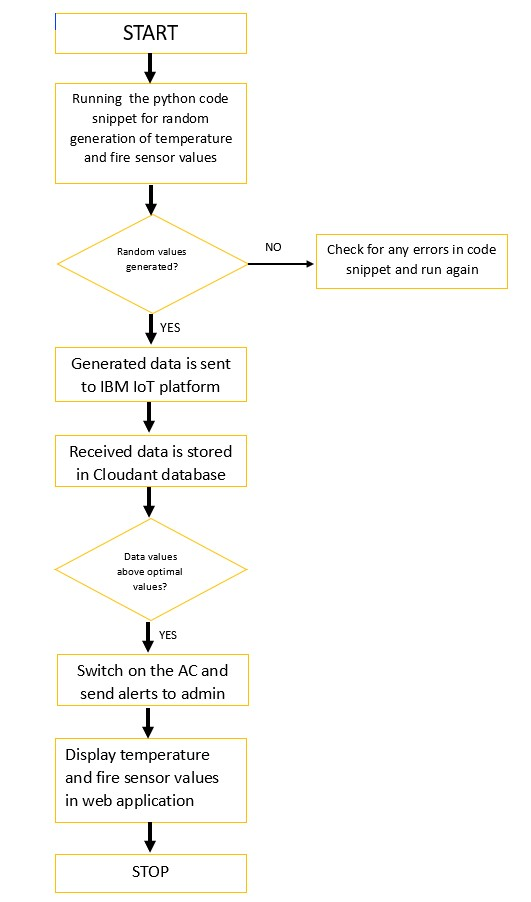
To successfully monitor the state of perishable goods , some sensors are placed in the containers containing the perishable goods such as medicines , fruits.Since hardware is not available ,  a python code snippet is used to generate random values such as temperature and fire sensor. This randomlygenerated data is sent to IBM IoT platform.

Using node red service , a node red flow is created which retrieves the data from IBM IoT platform.This received data is stored in Cloudant database.Since the high values of temperature or fire sensor will result in deterioration of goods , the generated data is always checked whether it exceeds an optimal value or not.

If either temperature or fire sensor values exceed the optimal value , sms alerts are sent to the admin and the AC is automatically switched on.

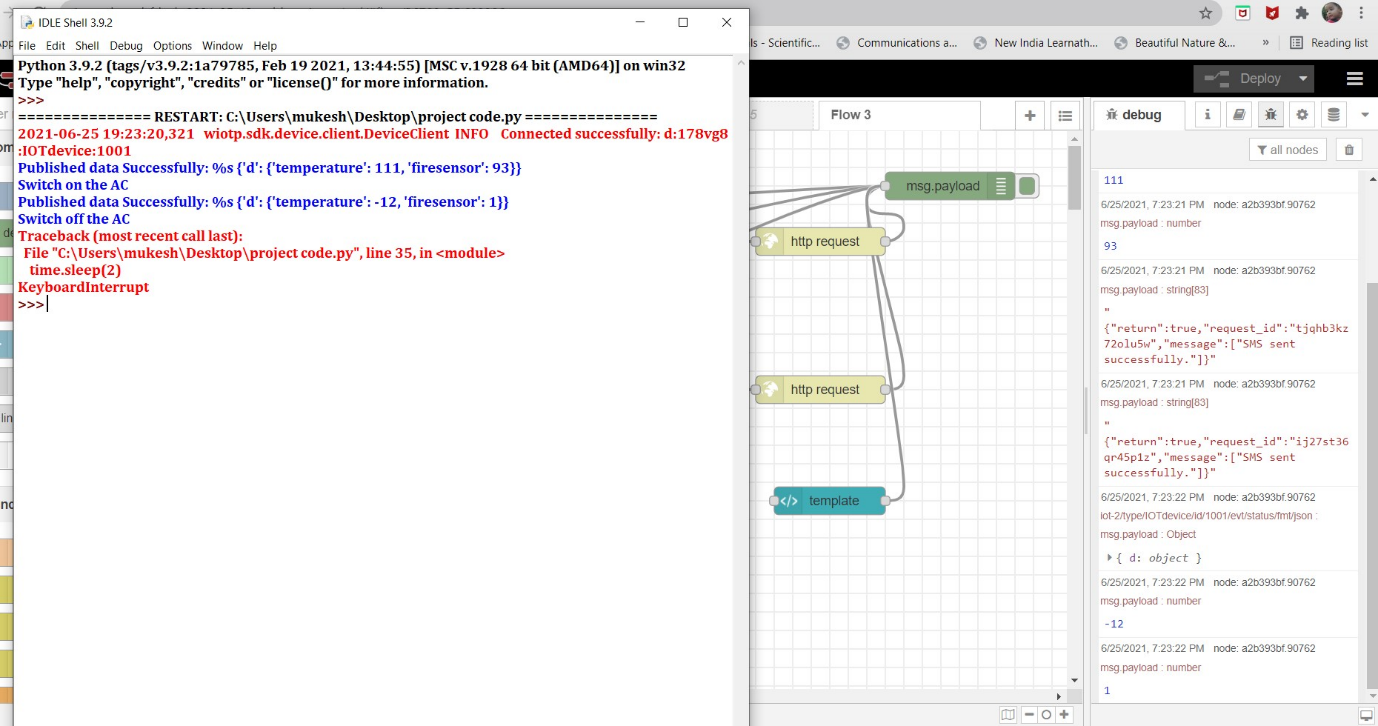


**5.FLOWCHART:-**



**6.RESULT:-**

The result of this project fulfills our purpose of continuous monitoring of state of perishable goods.This reduces the loss of revenue due to deterioration of goods during transport.



**7.ADVANTAGES & DISADVANTAGES:-**

**Advantages:**

 Easier maintenance of perishable goods

 Reduces loss of spoilage of goods such as fruits due to temperature deviations during transportation

 Reduces loss of revenue for the goods sellers since there is less spoilage

**Disadvantages:**

⭙ Consumes much revenue since installing the sensors is somewhat expensive

⭙ The maintenance of sensors is expensive

⭙ It is difficult if the sensors get damaged during transportation

**8.APPLICATIONS:**

➤ In transport of pharmaceutical goods such as vaccines and medicines.

➤ In transport of food products such as fish , meat , fruits and dairy products and much more.

➤ In monitoring the quality of food products in super markets.

**9.CONCLUSION:-**

Food safety plays an important role in a country's health and economy.Hence it is necessary to maintain a good quality of food products.Pharmaceutical products such as vaccines also play a crucial role in everyone’s life due to the pandemic situation.

Sometimes people may consume the spoiled goods and may get affected with severe problems or it may lead to death too.Hence maintaing the perishable goods and avoiding their damage is a necessity to prevent loss of revenue or a human life.

10.FUTURE SCOPE:-

This project can be extended beyond by using it at the automated good production units.

11.Bibliography:-

**♢** https://partheniumprojects.com/remote-monitoring-of-perishable-goods

**♢** https://inspirationfeed.com/how-to-keep-perishable-goods-in-optimal-condition-with-remote-temperature-monitoring

**12.APPENDIX:-**

**a.Source code:**

import wiotp.sdk.device

import time

import random

myConfig = {

    "identity": {

        "orgId": "178vg8",

        "typeId": "IOTdevice",

        "deviceId":"1001"

    },

    "auth": {

        "token": "1234567890"

    }

}

def myCommandCallback(cmd):

    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])

    m=cmd.data['command']

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)

client.connect()

T=24    #Threshold temperature

while True:

    temp=random.randint(-20,125)

    fs=random.randint(0,100)

    myData={"d":{'temperature':temp, 'firesensor':fs}}

    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)

    print("Published data Successfully: %s", myData)

    if temp > T:

        print("Switch on the AC")

    else:

  print("Switch off the AC")

    client.commandCallback = myCommandCallback

    time.sleep(2)

client.disconnect()

**b.UI OUTPUT:-**

