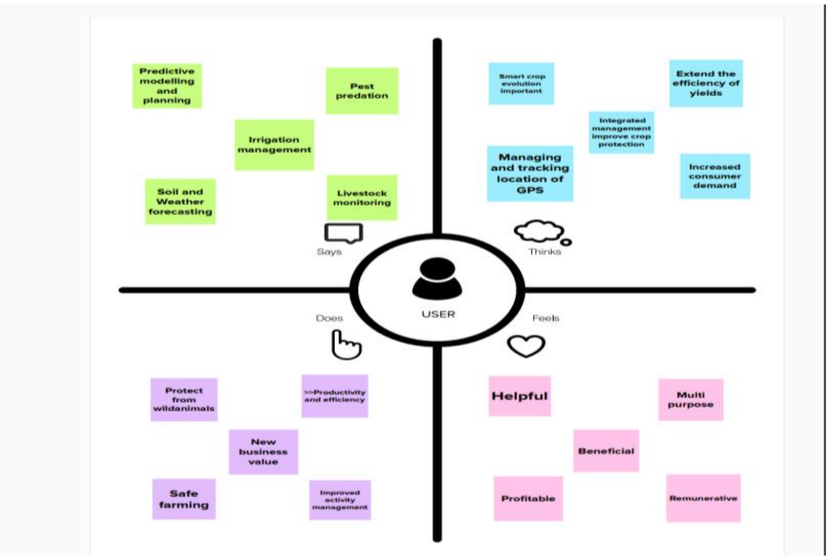
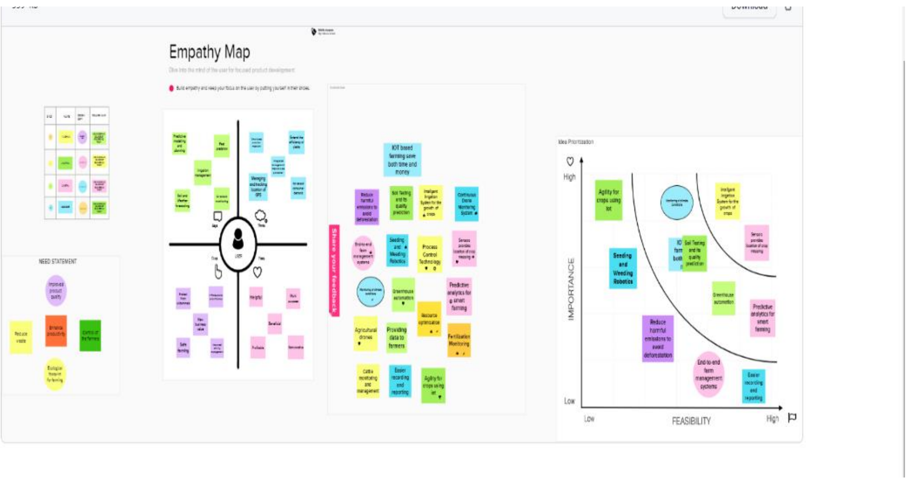
TEAM ID	**PNT2022TMID11096**
PROJECT NAME	**IOT BASED SMART CROP PROTECTION SYSTEM**
IBM ID	**IBM-Project-6301-1658825981**

IDEATION PHASE:

IDEATION:



EMPATHY MAP:



LITERATURE SURVEY:

IOT BASED SMART CROP PROTECTION SYSTEM FOR AGRICULTURE

TEAM LEADER	:	SUBASREE S
TEAM MEMBER1	:	VAISHNAVI V
TEAM MEMBER 2	:	SWETHA G
TEAM MEMBER 3	:	YUTHIKA A

LITERATURE SURVEY:

Nor Adni MatLeh, Zuraida Muhammad, Muhammad Azri Asyri Mohd Hafez, Zakiah Mohd Yusoff, and Shabinar Abd Hamid [1] The term **"Internet of Things"** describes the process of attaching equipment, cars, and other items to a network in order to share data (IoT). The Internet of Things (IoT) is being used more often to link things and gather data. Therefore, the usage of the Internet of Things in agriculture is essential. The project's goal is to build a smart agriculture network that is integrated with the internet of things. To deal with Malaysia's changing weather, technology is coupled with an irrigation system. The **Raspberry Pi 4 Model B** serves as the system's microcontroller. The temperature and humidity in the surrounding region, as well as the moisture level of the soil, are monitored using the **DHT22** and soil moisture sensor. The data will be available on both a smartphone and a computer. As a result, Internet of Things (IoT) and Raspberry Pi-based Smart Agriculture Systems have a significant impact on how farmers work. It will have a good impact on agricultural productivity as well. In Malaysia, employing IoT-based irrigation systems saves roughly 24.44 percent per year when compared to traditional irrigation systems. This would save money on labour expenditures while also preventing water waste in daily needs.

Janani V., Divya J., and Divya M. Both the economy and the existence of the Indian people depend on agriculture. The goal of this project is to develop an embedded-based irrigation and soil monitoring system that will lessen the need for manual field monitoring and deliver data via a mobile app. The technique is designed to assist farmers in boosting agricultural

soil because of excess moisture. A computerised irrigation system is necessary for water conservation and, as a result, agricultural success due to the limited availability of water. Around 85% of the water resources that are globally accessible are used for irrigation. The population is expected to grow in the upcoming years, increasing this need. We must use innovative techniques that reduce the amount of water used in order to satisfy this need.

Amina Saidi, Khaoula DJELLOUT, Mounir BOUHEDDA, and Hamza Benyeza -We all face a global problem with water management, so we must prepare carefully if we want to solve it soon. As a result of the abundance of usable sensors in today's society, systems with water-saving capabilities can be created. The goal of the work presented in this paper is to increase the farmer's ability to use water effectively. It basically uses a soil moisture sensor to determine the amount of moisture in the soil and then connects to the Thing Speaks cloud through ESP8266 Wi-Fi to monitor the soil's status. The proposed system is also equipped with an algorithm that forecasts crop irrigation decisions based on information about soil moisture patterns. If this happens, the device also alerts farmers when the water supply is exhausted. Weather forecasting via internet is another benefit of adopting this approach. The device's energy independence and inexpensive cost make it potentially useful in areas with limited water resources and remote locations. The technology's usefulness is increased by the ease with which farmers may use it. By reducing waste, it also conserves water.

increase crop productivity. The primary goal of this project is to grow crops while utilising the least amount of water feasible.

Dhanashri H. Gawali and Shweta B. Saraf The connecting of a vast number of gadgets to the internet (the "Internet of Things") (IoT). Each item is connected with a distinct identity, enabling data transmission without human intervention. It enables the creation of plans for better natural resource management. According to the IoT concept, smart devices with sensors allow interaction with both the physical and logical worlds. The proposed system in this study leverages real-time input data and is based on the Internet of Things. An Android phone is used by a smart farm irrigation system to remotely monitor and control drips using a wireless sensor network. Zigbee is used for communication between base stations and sensor nodes.

M. Shrihari- Irrigation is a major issue that both scientists and farmers face while trying to automate agricultural production; this idea has been around since the early 1990s. A dynamic system, irrigation is heavily influenced by external factors. In order to create a smart system, the method described in this article uses a specially created mathematical model to manage data from wireless sensors on Google Cloud. a design that can scale up to large farms and is IoT connected. Holistic Agricultural Studies estimate that 35 have been harmed by both animals and people. This smart system can identify humans who are unauthorised visitors to the farm as well as animals depending on their threat level using Tensor flow and deep learning neural networks.

Sujatha and G. Sushanth Since Internet of Things (IoT) sensors may provide information about agricultural area and then act on it based on user input, smart agriculture is a revolutionary notion. The goal of this research is to create a smart agricultural system that uses cutting-edge technologies including wireless sensor networks, the Internet of Things, and Arduino. The goal of this study is to create a system that can use sensors to track temperature, humidity, wetness, and even the movement of animals that could harm crops in agricultural areas. If there is a discrepancy, the system will send an SMS notification to

PROJECT DESIGN PHASE 1:

PROJECT SOLUTION FIT:

Problem-Solution fit canvas 2.0

Purpose / Vision

1. CUSTOMER SEGMENT(S) C

- Crop Management
- Precision Farming.
- Data Analytics
- Agricultural Drones.
- Robotic System.

5. CUSTOMER CONSTRAINTS C

- Automate process reduce labor costs.
- Sensors are used.
- Compact in Size.
- Gives data within a fraction of seconds
- Usually Iot devices are small and light.

8. AVAILABLE SOLUTIONS A

- This technology plays a vital role in Seeding and Weeding Robotics it is very useful to farmers and reduce their labour costs.
- Continuous Drones Monitoring System gives a best result to Farmers.
- Sensors provides location of crop mapping helps the farmers to identify the crops easily

2. JOBS-TO-BE-DONE / PROBLEMS J&P

- To managing and tracking location of GPS using IOT.
- To extend the efficiency of yields.
- To enable Integrated management to improve crop protection.
- To monitor the soil,humidity,pest,insect attacks in agriculture fields.

6. PROBLEM ROOT CAUSE R

- It requires an unlimited or continuous internet connection.So that this problem plays a vital role.
- It should definetly Diminishes Soil Fertility.
- Smart farming based equipments require famers to understand and learn the use of technonlogy.
- Use of New chemicals leads to an annual loss of Soil Quality.

9. BEHAVIOUR B

- To predict the soil Monitoring,Humidity Monitoring,Temperature Monitoring,ph Monitoring,Cattle Monitoring ,Fertilization Monitoring.So many things are Benefical here.
- Easier Recording and Reporting,Providing data to Farmers continuously.
- It reduce the labour cost and user friendly.

3. TRIGGERS T

- Farmers are able to recognise the issues and work without anyone help.
- Customers find easy to use it.Because it is fully digital,faster enough,automatic too.

4. EMOTIONS: BEFORE / AFTER EM

- Before :Anxity,time consumption and unaware of things
- After:aware of things ,less time consumption and pleasure

7. YOUR SOLUTION SL

- The system finds a way for supervising and monitoring the Crops so that quality & Quantity can be maintained.
- To consume less power consumption and to provide in cheaper cost
- The device will be in compact size and user friendly to use

10. CHANNELS of BEHAVIOUR CH

ONLINE

- Data Analytics used to give data to farmers regularly.Storage of data also safe using iot

OFFLINE

- The proposed system includes a number of sensors to test and guarantee the Crop quality based on factors including pH, temperature, conductivity, humidity and ardunio.

AMALTAMA

PROPOSED SOLUTION TEMPLATE:

Project Design Phase-I
Proposed Solution Template

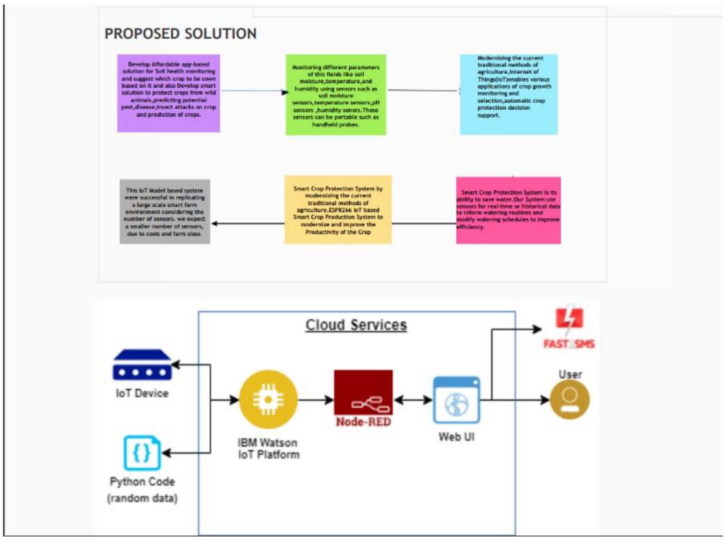
Date	01 October 2022
Team ID	PNT2022TMID11096
Project Name	Project – IoT Based Smart Crop Protection System For Agriculture.
Maximum Marks	2 Marks

Proposed Solution Template:

Project team shall fill the following information in proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Develop Affordable app-based solution for Soil health monitoring and suggest which crop to be sown based on it and also Develop smart solution to protect crops from wild animals ,predicting potential pest,disease,insect attacks on crop and prediction of crops.
2.	Idea / Solution description	We are about to propose a solution for monitoring different parameters of his field like soil moisture, temperature, and humidity using sensors such as soil moisture sensors, temperature sensors and a humidity sensor. These sensors can be stationary or portables such as handheld probes. **A temperature sensor is for detecting and measuring the hotness and coolness present in the environment and converts those inputs into an electrical signal. A humidity sensor is to detect and measure the water vapour or water droplets present in the atmospheric air and with those inputs it measures the humidity present in the air.**
3.	Novelty / Uniqueness	• Modernizing the current traditional methods of agriculture • Internet of Things (IoT) enables various applications of **crop growth monitoring** and selection, **automatic crop protection** decision support.
4.	Social Impact / Customer Satisfaction	One of the greatest advantages of a smart crop protection system is its ability to save water. In general, traditional watering methods can waste as much as 50% of the water used due to inefficiencies in irrigation, evaporation and overwatering. Our system use sensors for real-time or historical data to inform watering routines and modify watering schedules to improve efficiency. Users can configure these systems to manage crop protection on demand.

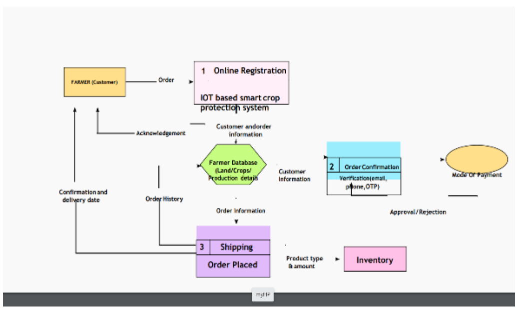
SOLUTION ARTICHECTURE:



PROJECT DESIGN PHASE 2:

CUSTOMER JOURNEY:

DATA FLOW DIAGRAM:



FUNCTIONAL REQUIREMENTS:

Non-functional Requirements:

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

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Have a clear and self-explanatory manual. Easier to use Even an illiterate farmer have to use the product without any difficulties
NFR-2	**Security**	Application has to be secured with 2 step authorisation Passwords and passkeys will be assigned as per the users need.
NFR-3	**Reliability**	Hardware requires a regular checking and service Software may be updated periodically Immediate alert is provided in case of any system failure
NFR-4	**Performance**	The application must have a good user interface It should have a minimal energy requirement It has to save water and energy
NFR-5	**Availability**	All the features will be available when the user requires. It depends on the need of the farmer and the customization the user has done.
NFR-6	**Scalability**	The product has to cover all the space of land irrespective of the size or area of a farm field.

Project Design Phase-II

Solution Requirements (Functional & Non-functional)

Date	03 October 2022
Team ID	PNT2022TMID11096
Project Name	Project - Iot Based Smart Crop Protection System for Agriculture
Maximum Marks	4 Marks

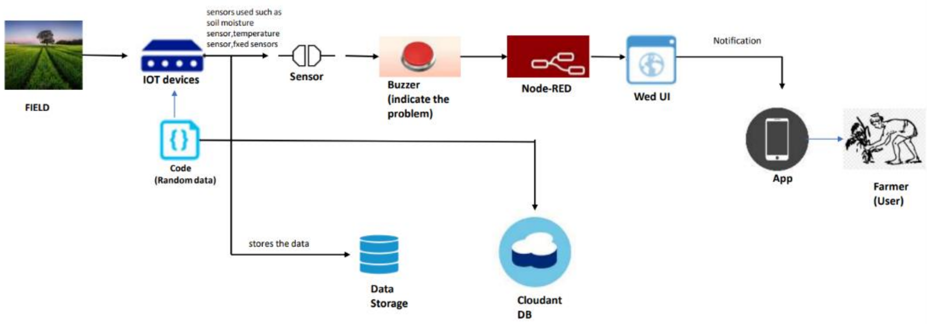
Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	**User Requirements**	Crop Protection System Automatic Sprinkler System Monitors Soil Moisture and ph ,Humidity and Temperature.
FR-2	**User Registration**	Manual Registration Registration through webpage Registration through Form Registration through Gmail
FR-3	**User Confirmation**	Confirmation via Phone Confirmation via Email Confirmation via OTP
FR-4	**Payment Options**	Cash on Delivery Net Banking/UPI Credit/Debit/ATM Card
FR-5	**Product Delivery and Installation**	Door Step delivery Take away Free Installation and 1 year Warranty
FR-6	**Product Feedback**	Through Webpage Through Phone calls Through Google forms

TECHNOLOGY ARCHITECTURE:

Solution Architecture Diagram:



Architecture and data flow of the IoT Based Smart Crop Protection System For Agriculture

PROJECT PLANNING PHASE:

PROJECT MILESTONE:

PROJECT PLANNING PHASE

PROJECT MILESTONE

Date	21 October 2022
Team ID	PNT2022TMID11096
Project Name	IOT BASED SMART CROP PROTECTION SYSTEM
Maximum Marks	4 marks

S.NO	ACTIVITY TITLE	ACTIVITY DESCRIPTION	DURATION
1	Understanding the project requirement	Assign the team members and create repository in the Github, Assign the task to each members and teach how to use and open and class the Github and IBM career education	1 WEEK
2	Starting of project	Advice students to attend classes of IBM portal create and develop an rough diagram based on project description and gather of information on IOT and IBM project and team leader assign task to each member of the project	1 WEEK
3	Attend class	Team members and team lead must watch and learn from classes provided by IBM and NALAYATHIRAN and must gain	4 WEEK

SPRINT DELIVERY PLAN:

Project Planning Phase

Sprint Delivery Plan

Date	21 October 2022
Team ID	PNT2022TMID11096
Project Name	IOT BASED SMART CROP PROTECTION SYSTEM
Maximum Marks	8 Marks

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Subasree S
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Vaishnavi V
Sprint-2		USN-3	As a user, I can register for the application through Facebook	2	Low	Yukthika A
Sprint-1		USN-4	As a user, I can register for the application through Gmail	2	Medium	Swetha
Sprint-1	Login	USN-5	As a user, I can log into the application by Entering email & password	1	High	Subasree

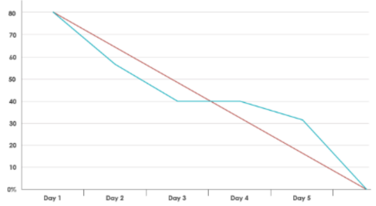
Project Tracker, Velocity & Burndown Chart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	30	30 Oct 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	49	06 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	50	07 Nov 2022

Velocity:

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

Burndown Chart:



PROJECT DEVELOPMENT PHASE:

SPRINT 1:

(PNT2022TMID11096)

Sprint 1

```
import
random
        import ibmiotf.application
        import ibmiotf.device
        from time import sleep
        import sys

        #IBM Watson Device Credentials...
        organization = "tw9ckq"
        deviceType = "jade"
        deviceId = "7010"
        authMethod = "token"
        authToken = "9944893843"

        def myCommandCallback(cmd):
            print("Command received: %s" % cmd.data['command'])
            status=cmd.data['command']
            if status=="sprinkler_on":
                print ("sprinkler is turning ON")
            else :
                print ("sprinkler is turning OFF")

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

        except Exception as e:
            print("Exception detected in connecting device: %s" % str(e))
            sys.exit()

        #Connecting to IBM watson...
        deviceCli.connect()

        while True:
            #Getting values from sensors...
            temp_sensor = round( random.uniform(0,80),2)
```

```
    PH_sensor = round(random.uniform(1,14),3)
    camera = ["Detected","Not Detected","Not Detected","Not Detected","Not
Detected","Not Detected",]
    camera_reading = random.choice(camera)
    flame = ["Detected","Not Detected","Not Detected","Not Detected","Not
Detected","Not Detected",]
    flame_reading = random.choice(flame)
    moist_level = round(random.uniform(0,100),2)
    water_level = round(random.uniform(0,30),2)

    #storing the sensor data to send in json format to cloud.
    temp_data = { 'Temp' : temp_sensor }
    PH_data = { 'PH value' : PH_sensor }
    camera_data = { 'Animal attack' : camera_reading}
    flame_data = { 'Flame' : flame_reading }
    moist_data = { 'Moisture level' : moist_level}
    water_data = { 'Water level' : water_level}

    # publishing Sensor datas to IBM Watson for every 5-10 seconds.
    success = deviceCli.publishEvent("Temperature sensor", "json", temp_data,
qos=0)
    sleep(1)

    if success:
        print ("... ...publish ok... ... ...")
        print ("Published Temp = %s C" % temp_sensor,  "to IBM Watson")
        success = deviceCli.publishEvent("PH sensor", "json", PH_data, qos=0)
        sleep(1)

    if success:
        print ("Published PH value = %s" % PH_sensor,  "to IBM Watson")
        success = deviceCli.publishEvent("camera", "json", camera_data, qos=0)
        sleep(1)

    if success:
        print ("Published Animal attack %s " % camera_reading,  "to IBM
Watson")
        success = deviceCli.publishEvent("Flame sensor", "json", flame_data,
qos=0)
        sleep(1)

    if success:
        print ("Published Flame %s " % flame_reading,  "to IBM Watson")
        success = deviceCli.publishEvent("Moisture sensor", "json", moist_data,
qos=0)
        sleep(1)
```

SPRINT 2:

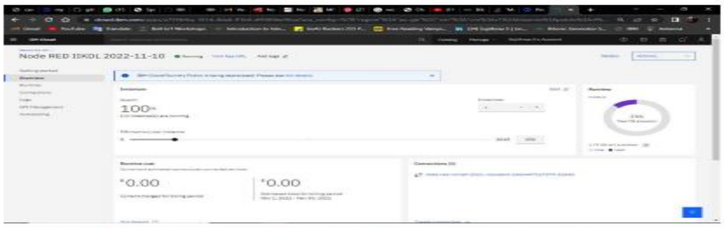
Sprint – 2
Software creation

Date	17 November 2022
Team ID	(PNT2022TMID11096)
Project Name	IoT Based Smart Crop Protection System For Agriculture.

Description:

According to our sprint-2 planning we've created the accounts in the required platforms (Node-red, Cloudant, Cloud object storage, Watson) and linked with our project.

Node-Red:



SPRINT 3:

Sprint – 3
Mobile App Creation

Date	17 November 2022
Team ID	(PNT2022TMID11096)
Project Name	IoT Based Smart Crop Protection System For Agriculture.

Description:

According to our sprint-3 planning we've developed the mobile application to monitor the field and we've linked it with the python coding. This app consists of login credentials in which only the registered user can access. After logging in, the value of Temperature, Humidity, Moisture and Animal Invasion will be displayed in the mobile screen. We can also control the sprinkler using the controller.

Controller using mobile application for monitoring the field will be a more user friendly way for the farmers. This will reduce the work load of the farmers. The values generated from the python using random will be sent to the mobile application through the node-red.

MIT App Inventor:

SPRINT 4:

Sprint – 4

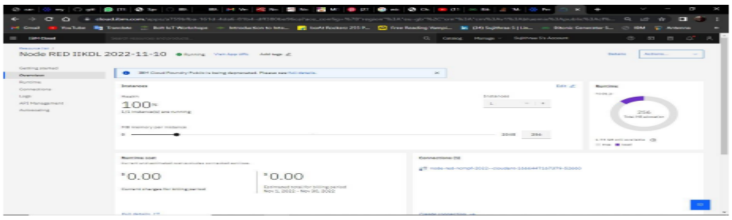
Web UI creation

Date	17 November 2022
Team ID	(PNT2022TMID11096)
Project Name	IoT Based Smart Crop Protection System For Agriculture.

Description:

According to our sprint-3 planning we've developed the web UI to monitor the field and we've linked it with the python coding. This web UI consists of pictorial representations of the humidity & temperature. If the animal is invaded on the field, the alert will be sent with the picture of the animal. We can also control the pump using this web UI.

Web UI interface:



DEVELOP A PYTHON SCRIPT:

IOT Based Smart Crop Protection System for Agriculture .

Team ID - PNT2022TMID11096

DEVELOPING PYTHON SCRIPT

LOCATION DATA:

```
import wiotp.sdk.device
import time import
random myConfig={
"identity": (
"orgId": "gagtey",
"typeId": "GPS",
"deviceId":"12345"},
"auth": {
"token": "12345678"
}}
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() def pub (data):
```

```
client.publishEvent (eventId="status", msgFormat="json", data=myData, qos=0, print("Published data Successfully: %s", myData) while True:
myData={'name': 'Train1', 'lat': 17.6387448, 'lon': 78.4754336) pub
(myData)
time.sleep (3)
#myData('name': 'Train2', 'lat': 17.6387448, 'lon': 78.4754336)
#pub (myData) #time.sleep (3) myData={'name':
'Train1', 'lat': 17.6341908, 'lon':
78.4744722) pub
(myData)
time.sleep(3)
myData={'name': 'Trainl', 'lat': 17.6340889, lon': 78.4745052)
pub (myData) time.sleep(3)
myData={'name': 'Trainl', 'lat': 17.6248626, 'lon': 78.4720259)
pub (myData) time.sleep (3)
myData={'name': 'Trainl', 'lat': 17.6188577, 'lon': 78.4698726)
pub (myData) time.sleep (3) myData={'name': 'Train1', 'lat':
17.6132382, 'lon':
```

```
78.4707318) pub
(myData)
time.sleep (3)
client.commandCallback = myCommandCallback
client.disconnect()
```

QR SCANNER CODE:

```
Import cv2 import
numpy as np import
time
Import pyzbar.pyzbar as pyzbar
from ibmcloudant.cloudant_v1 import CloudantV1 from
ibmcloudant import CouchDbSessionAuthenticator
from ibm_cloud_ sdk_core.authenticators import
BasicAuthenticator
authenticator= BasicAuthenticator ('apikey-v2-
16u3crmdpkghhxefdikvpssoh5fwezrmuup5fv5g3ubz',
'b0ab119f45d3e6255eabb978
service Cloudant V1 (authenticator-authenticator)
service.set_service_url('https://apikey-
v216u3crmdpkghhxefdikvpssoh5fwezrmuup5fv5g3ubz:b0ab
119 f45d3e6255eabb978e7e2f0 cap= cv2.VideoCapture (0)
font cv2.FONT HERSHEY PLAIN while True:
frame cap.read()
```

```
decodedobjects pyzbar.decode (frame)
for obj in decodedObjects: #print
("Data", obj.data) a-
obj.data.decode('UTF-8')
cv2.putText (frame, "Ticket", (50, 50), font, 2,
(255, 0, 0), 3) #print
(a)
try: response = service.get_document ( db='booking,
doc_id = a
).get_result()
print (response) time.sleep(5) except
Exception as e:
print ("Not a Valid Ticket")
time.sleep (5) cv2.imshow("Frame",
frame) if cv2.waitKey(1) &
0xFF==ord('q'):
break cap.release()
cv2.destroyAllWindows ()
client.disconnect()
```

CREATE AND CONFIGURE IBM CLOUD SERVICE:

IBM WATSON IOT PLATFORM DEVICE:

IOT Based Smart Crop Protection System for Agriculture.

Team ID – PNT2022TMID11096

Create IBM Watson IOT Platform and Device