IOT– SMART CROP PROTECTION SYSTEM FOR AGRICULTURE

NALAIYA THIRAN BASED LEARNING

Submitted By

TEAM ID : PNT2022TMID46063

B . SUDHARSHAN – 814719106059

G . SRIDHAR PRASAD – 814719106056

K . VELMURUGAN – 814719106064

V . SELVA VASANTH - 814719106052

Bachelor Of Engineering

In

Electronics And Communication Engineering

SRM TRP ENGINEERING COLLEGE

NH4 , Manachanallur Taluk , Tiruchirapalli District ,

Irungalur , Tamil Nadu 621105

INDEX

1. INTRODUCTION:

1.1 PROJECT OVERVIEW:

The Smart protection system defines that this project help to farmer for the protection of a farm. We have designed this project for the only secure from animals but we this project have the provision to secure from the human begins also. This can achieve by the help of IOT device that we are discuss in this paper. The SCPS work on the battery so that this project can be easily portable and also we are add solar panels and converter modules this can help the battery to charge from solar energy. The IOT device is used to indicate the farmer by a message while someone enter into the farm and we are used SD card module that helps to store a specified sound to fear the animals. This project is smart crop protection system for protect the farm from animals as well as unknown person. This projects contents ardiuno UNO, Nodemcu, LCD display, PIR sensor, flame sensor, sd card module ,solar panel, solar charges converter. This whole project is work on 12v dc supply from battery. We used solar panel to charge the battery.

1.2 PURPOSE:

The project aims at making use of evolving technology i.e. IOT and smart agriculture using automation. Monitoring environmental conditions is the major factor to improve yield of the efficient crops.

2. LITERATURE SURVEY:

2.1 EXISTING PROBLEM:

Iot based smart agriculture monitoring system. Rajalakshmi.P and S. Devi Mahalakshmi, "IOT Based Crop Field Monitoring and Irrigation Automation", 10th International conference on Intelligent systems and control (ISCO), 2016. An IOT Based Crop-field monitoring an irrigation automation system describes how to monitor a crop field. A system is developed by using sensors and according to the decision from a server based on sensed data, the irrigation system is automated. Through wireless transmission the sensed data is forwarded to web server database. If the irrigation is automated then the moisture and temperature fields are decreased below the potential range. The user can monitor and control the system remotely with the help of application which provides a web interface to user. By smart Agriculture monitoring system and one of the oldest ways in agriculture is the manual method of checking the parameters. In this method farmers by themselves verify all the parameter and calculate the reading .The system focuses on developing devices and tool to manage, display and alert the users using the advantages of a wireless sensor network system. It aims at making agriculture smart using automation and IoT technologies. The cloud computing devices are used at the end of the system that can create a whole computing system from sensors to tools that observe data from agriculture field. It proposes a novel methodology for smart farming by including a smart sensing system and smart irrigator system through wireless communication technology. This system is cheap at cost for installation. Here one can access and also control the agriculture system in laptop, cell phone or a computer.

LIMITATIONS

1. There could be a wrong analysis of weather conditions.

2. Devices are to be altered according to the farmers, it will involve equipment which will be expensive.

3. If there are faulty data processing equipment or sensors, then it will lead to a situation where the decisions are taken wrong .

2.2 REFERENCES:

[1] Sinung Suakanto, Ventje J. L. Engel, Maclaurin Hutagalung, Dina Angela, “Sensor networks data acquisition and task management for decision support of smart agriculture,” in 2016 International Conference on Information Technology Systems and Innovation (ICITSI) Bandung – Bali, pp. 24–27, Oct. 2016.

[2] Chetan Dwarkani M, Ganesh Ram R, Jagannathan S, R. Priyatharshini “Smart agriculture system using sensors for agricultural task automation,” in 2015 IEEE International Conference on Technological Innovations in ICT for Agriculture and Rural Development (TIAR 2015).

[3] Nikesh Gondchwar, R. S. Kawitkar, “IOT based smart agriculture,” International journal Of Advanced research in computer and Communication Engineering (IJARCCE), vol. 5, no. 6, Jun. 2016.

[4] Narayut Putjaika, Sasimanee Phusae, Anupong Chen-Im, Phond Phunchongharn and Khajonpong Akkarajit Sakul, “A control system in intelligent agriculture by using arduino technology,” in Fifth ICT International Student Project Conference(ICT-ISPC), 2016.

[5] Tejas Bangera, Akshar Chauhan, Harsh Dedhia, Ritesh Godambe, Manoj Mishra, “IOT based smart village,” International Journal of Engineering Trends and Technology (IJETT), vol. 32, no. 6, Feb. 2016, ISSN: 2231- 5381.

[6] Jeetendra Shenoy, Yogesh Pingle “IOT in agriculture,” 978-9-3805-4421-2/16/, IEEE. 2016.

[7] Rajalakshmi P and S. Devi Mahalakshmi, IOT Based Crop-Field Monitoring and Irrigation Automation.

[8] Abdullah Na, William Isaac, “Developing a human-centric agricultural model in the IOT environment,” in 2016 International Conference on Internet of Things and Applications (IOTA) Maharashtra Institute of Technology, Pune, India 22 Jan - 24 Jan, 2016, 978-1-5090-0044-9/16, 2016 IEEE.

[9] Syed Mubarak and S. Sujatha “International journal of advance research in science and engineering,” IJARSE, vol. 4, no. 01, May 2015, ISSN23198354(E).

[10] Joaquín Gutiérrez, Juan Francisco Villa-Medina, Alejandra Nieto-Garibay, and Miguel Ángel Porta- Gándara”Automated Irrigation System Using a Wireless Sensor Network and GPRS module” , Ieee Transactions On Instrumentation And Measurement, Vol. 63, No. 1, January 2014.

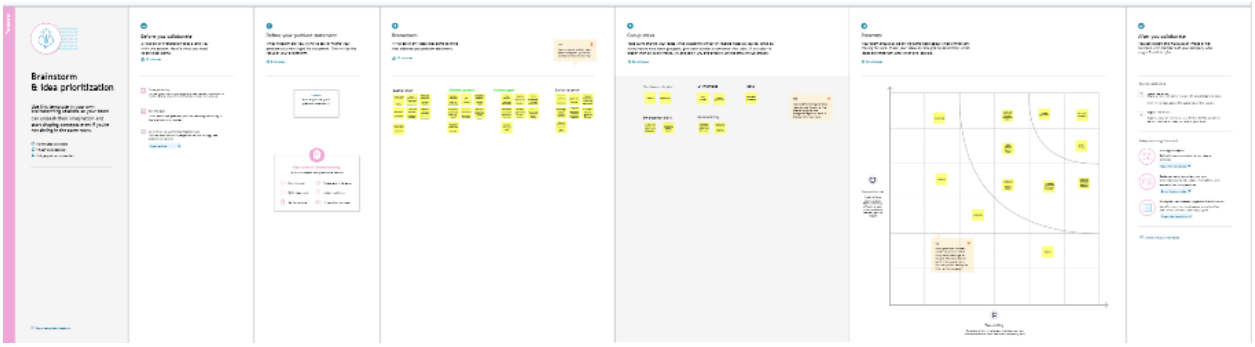
2.3 PROBLEM STATEMENT DEFINITION:

Proposes the automatic irrigation system using Arduino for smart crop field productivity. This system consists of sensor like moisture sensor, temperature sensor, rain sensor and ultrasonic sensor. Moisture sensor used for detecting the moisture content in soil, temperature sensor is used to measure the temperature value and ultrasonic sensor is used to measure the water level. If the excess of water in field is detected, then motor will ON to remove it. The measured parameters are uploaded to IOT through this farmer can monitor anywhere. Based on the command from IOT water motor will ON. The measured parameters are displays on LCD.

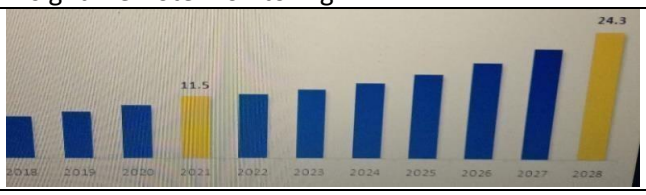
3. IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS:

3.2 IDEATION & BRAINSTORMING:



3.3 PROPOSED SOLUTION:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Usually crops in the fields are protected against birds and other unknown disturbances by humans. This take an enormous amount of time. Creating a smart automatic system will benefit the farmers in many Different ways.
2.	Idea / Solution description	Smart Farming has enabled farmers to reduce waste and enhance productivity with the help of sensors (light, humidity, temperature, soil moisture,etc..) . Further with the help of these sensors, farmers can Monitor the field conditions from anywhere.
3.	Novelty / Uniqueness	Role of SENSORS: IOT smart agriculture products are designed to help monitor crop fields using sensors and by automating irrigation systems. As a result, farmers and associated brands can easily monitor the Field conditions from anywhere without any hassle.
4.	Social Impact / Customer Satisfaction	Water conservation. Saves lot of time. Increased quality of production. Real time data and production Insight. Remote monitoring.
5.	Business Model (Revenue Model)	
6.	Scalability of the Solution	Scalability in smart farming refers to the adaptability of a system to increase the capacity , the number of Technology devices such as sensors and fluctuates.

3.4 PROBLEM SOLUTION FIT:

4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Install the app. Signing up with Gmail or phone number Creating a profile. Understand the guidelines.
FR-2	User Confirmation	Email or phone number verification required via OTP.
FR-3	Accessing datasets	Data's are obtained by cloudant DB.
FR-4	Interface sensor	Connect the sensor and the application When animals enter the field , the alarm is generated.
FR-5	Mobile application	It is used to control motors and field sprinklers.

4.2 NON-FUNCTIONAL REQUIREMENTS:

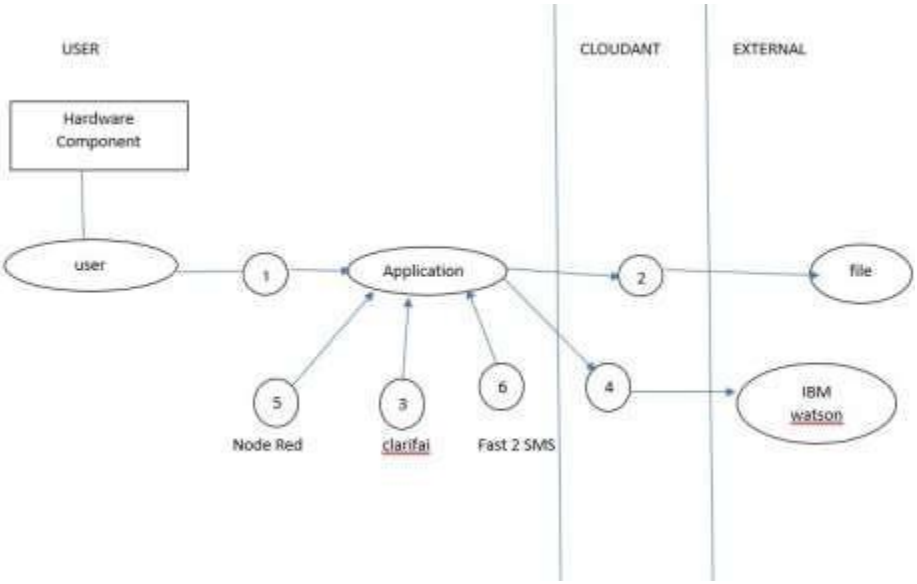
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	This project's contributes the farm protection through the smart protection system.
NFR-2	**Security**	It was created to protect the crops from animals.
NFR-3	**Reliability**	Farmers are able to safeguard their lands by help of this technology. They will also benefits from higher crop yields, which will improve our economic situation.
NFR-4	**Performance**	When animals attempt to enter the field, IOT devices and sensors alert the farmer via message.
NFR-5	**Availability**	We can defend the crops against wild animals by creating and implementing resilient hardware and software.
NFR-6	**Scalability**	This system's integration of computer vision algorithms with IBM cloudant services makes it more efficient to retrieve photos at scale, enhancing scalability.

5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS:

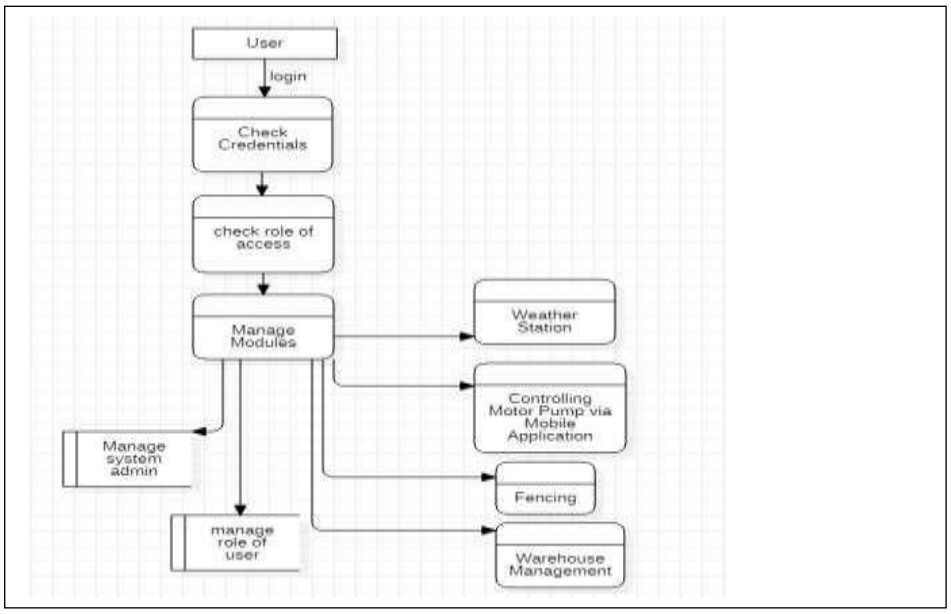
A data-flow diagram is a way of representing a flow of data through a process or a system. The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow — there are no decision rules and no loops. Specific operations based on the data can be represented by a flowchart.



5

S

TECHNICAL ARCHITECTURE:



5.3 USER STORIES:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirmingmy password.	I can access my account / dashboard.	High	Sprint-1
Customer	Registration	USN-2	As a user, I will receive confirmation messageonce I have registered for the application.	I can receive confirmationmessage & click confirm.	High	Sprint-1
Administrator	Login page	USN-3	As a user entering the username and password which is already existing.	Redirecting to useraccount.	Medium	Sprint-1
Weather station	Forecasting thecurrent weather	USN-4	As a user, we can monitor the weather fundamentals like (humidity, wind speed, wind direction and rainfall).	Notified about weatherconditions.	High	Sprint-1
Controlling theMotor Pump	Controlling	USN-5	It is used to control motors and field sprinklers.	Switching on and off the motor pump manually viamobile application	High	Sprint-2
Fencing	Detecting the motion in certain range	USN-6	Fencing system are helpful in providing security against unauthorized access of humanand animal.	I can receive notification; prevention has been taken.	High	Sprint-3
Warehouse management	Collecting database of crops	USN-7	Here famer need to update about expire dateof fertilizer and seeds.	Generate the popup message about expire date and stocks and offers	High	Sprint-4

6. PROJECT PLANNING & SCHEDULING

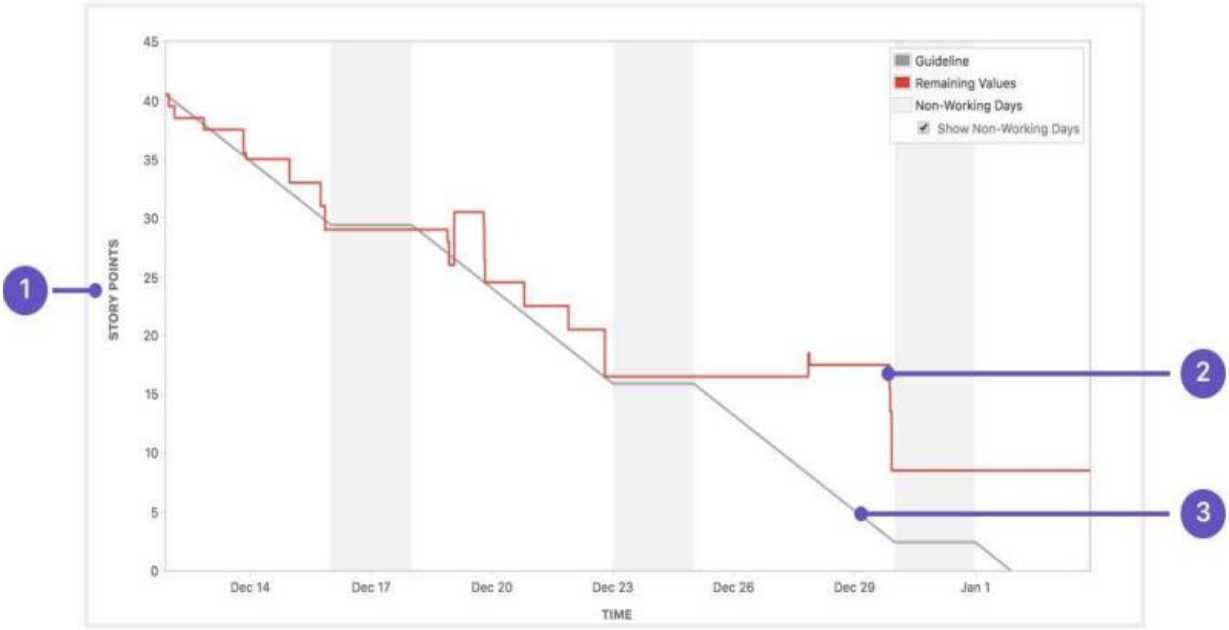
6.1 SPRINT PLANNING & ESTIMATION:

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	B.Sudharshan G.Shridhar prasad K.velmurugan.
Sprint-1	Verification	USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	B.Sudharshan G.Shridhar prasad K.velmurugan
Sprint-2		USN-3	As a user, I can register for the application through Facebook	2	Low	V.Selva Vasanth.
Sprint-1	Login	USN-4	As a user, I can register for the application through Gmail	2	Medium	B.Sudharshan G.Shridhar prasad
Sprint-1	Dashboard	USN-5	As a user, I can log into the application by entering email & password	1	High	B.Sudharshan G.Shridhar prasad K.velmurugan

6.2 SPRINT DELIVERY SCHEDULE:

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	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

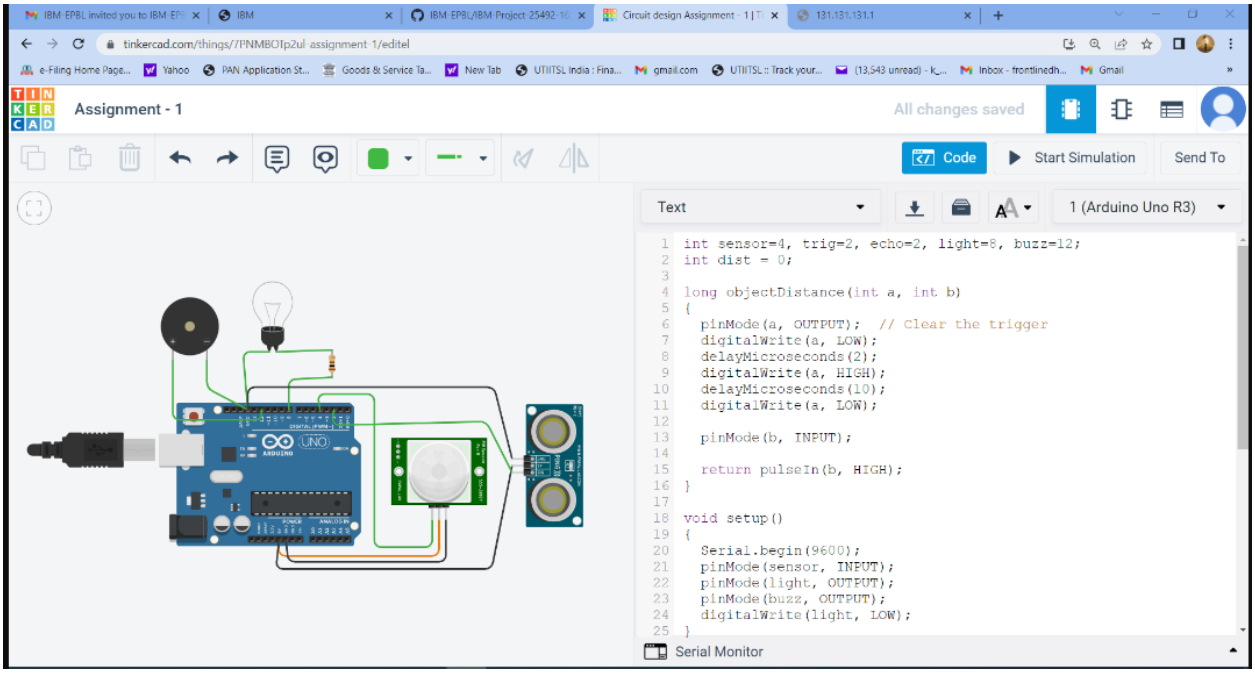
6.3 Reports from JIRA:

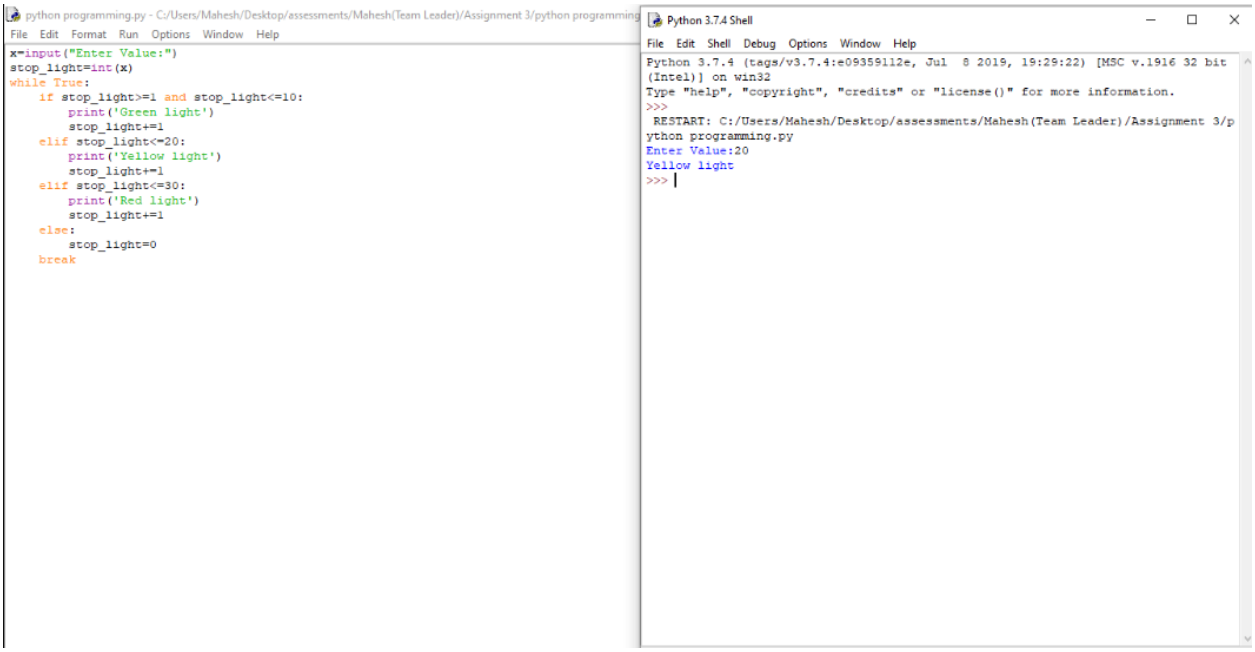


7. CODING & SOLUTIONING:

```
int
sensor=4,
trig=2,
echo=2,
light=8,
buzz=12;

                int dist = 0;


                long objectDistance(int a, int b)

                {

                  pinMode(a, OUTPUT);  // Clear the trigger

                  digitalWrite(a, LOW);

                  delayMicroseconds(2);

                  digitalWrite(a, HIGH);

                  delayMicroseconds(10);

                  digitalWrite(a, LOW);


                  pinMode(b, INPUT);


                  return pulseIn(b, HIGH);

                }

                void setup()

                {

                  Serial.begin(9600);

                  pinMode(sensor, INPUT);
```

```
  pinMode(light, OUTPUT);
  pinMode(buzz, OUTPUT);
  digitalWrite(light, LOW);
}
void loop()
{
  //readUltrasonicDistance(7, 7)
  dist = 0.01723 * objectDistance(trig, echo);
  Serial.print("Distance is ");
  Serial.print(dist);
  Serial.println("cm");
  if(dist>50 && dist<100)
  {
    tone(buzz, 50);
    delay(2000);
    noTone(buzz);
    //delay(1000);
    if(digitalRead(sensor))
    {
      digitalWrite(light, HIGH);
      delay(2000);
    }
```


Assignment - 1
All changes saved
Code
Start Simulation
Send To
Text
1 (Arduino Uno R3)
int sensor=4, trig=2, echo=2, light=8, buzz=12;
int dist = 0;
long objectDistance(int a, int b)
{
pinMode(a, OUTPUT); // Clear the trigger
digitalWrite(a, LOW);
delayMicroseconds(2);
digitalWrite(a, HIGH);
delayMicroseconds(10);
digitalWrite(a, LOW);
pinMode(b, INPUT);
return pulseIn(b, HIGH);
}
void setup()
{
Serial.begin(9600);
pinMode(sensor, INPUT);
pinMode(light, OUTPUT);
pinMode(buzz, OUTPUT);
digitalWrite(light, LOW);
}
Serial Monitor


python programming.py - C:/Users/Mahesh/Desktop/assessments/Mahesh(Team Leader)/Assignment 3/python programming.py
File Edit Format Run Options Window Help
x=input("Enter Value:")
stop_light=int(x)
while True:
if stop_light>=1 and stop_light<=10:
print('Green light')
stop_light+=1
elif stop_light<=20:
print('Yellow light')
stop_light+=1
elif stop_light<=30:
print('Red light')
stop_light+=1
else:
stop_light=0
break
Python 3.7.4 Shell
File Edit Shell Debug Options Window Help
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 19:29:22) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Users/Mahesh/Desktop/assessments/Mahesh(Team Leader)/Assignment 3/python programming.py
Enter Value:20
Yellow light
>>>

8 TESTING

8. SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

8.1 TYPES OF TESTS

8.1.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

8.1.2 Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

8.1.3 Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

8.1.4 System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

8.1.5 White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

8.1.6 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

8.2 Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

8.2.1 Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

8.2.2 Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

8.2.3 Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

8.3 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

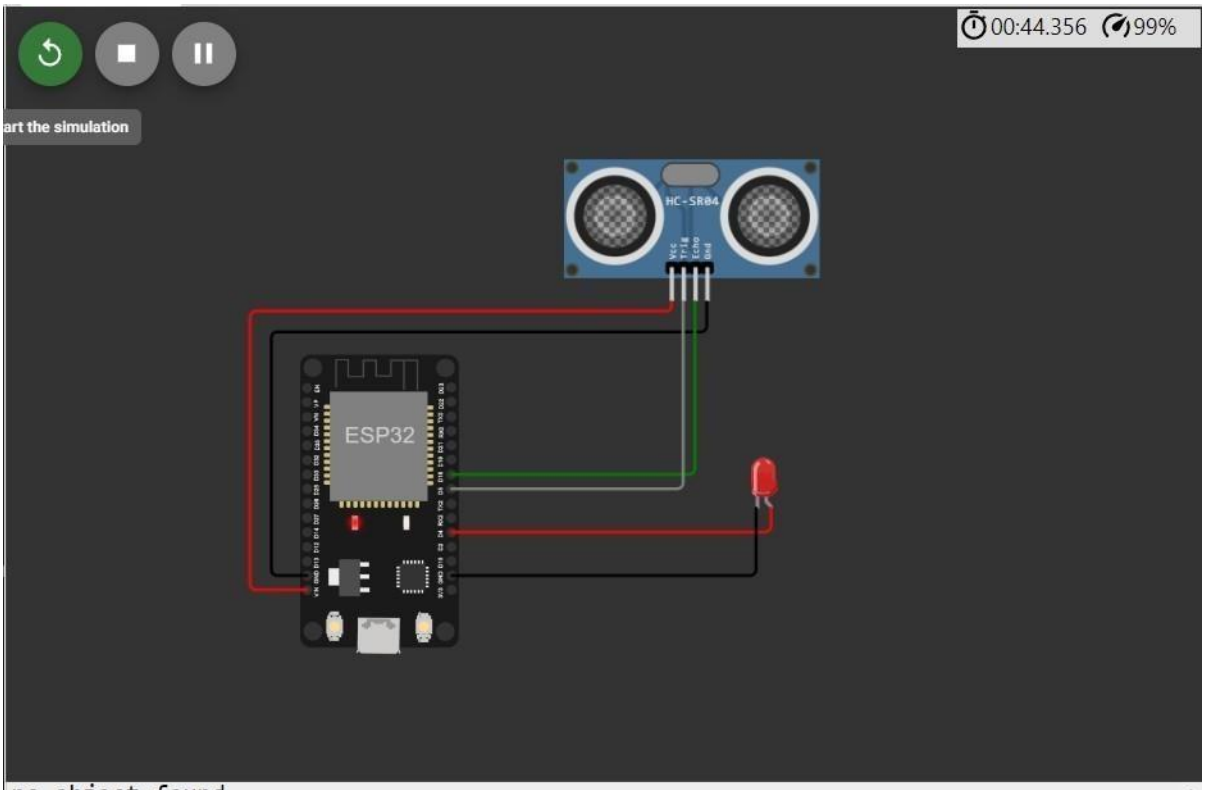
Test Results: All the test cases mentioned above passed successfully. No defects encountered.

8.4 Acceptance Testing

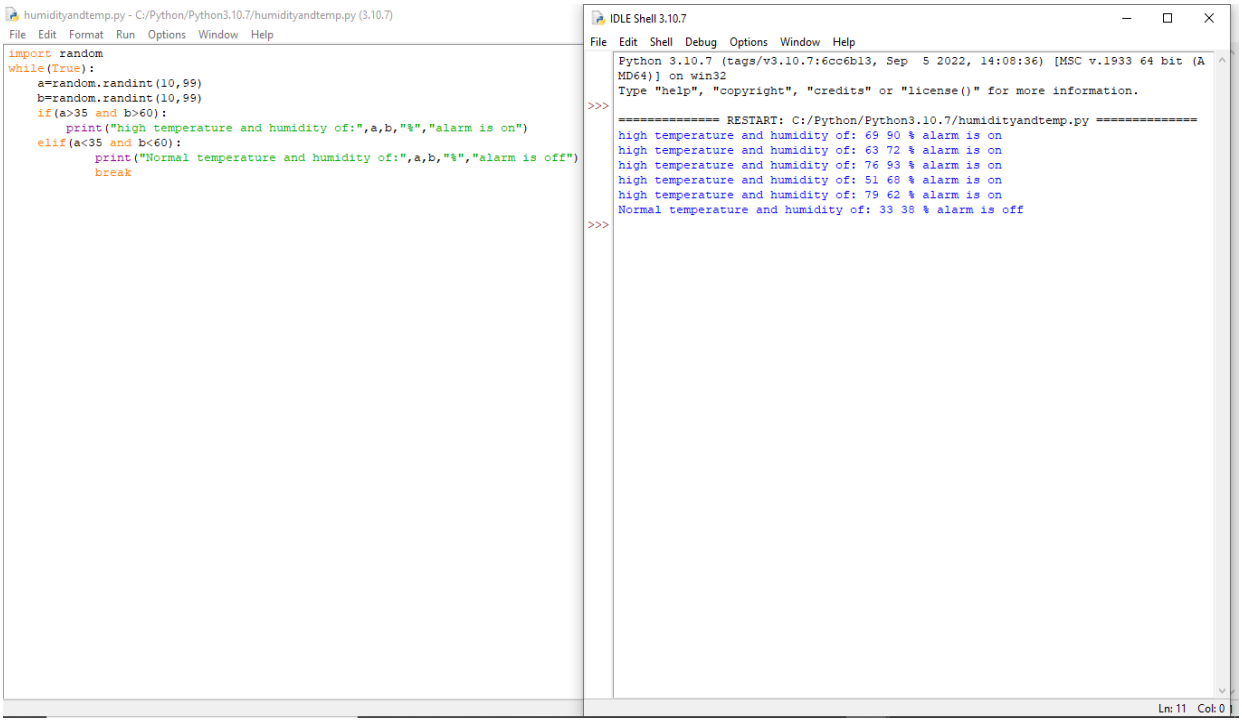
User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

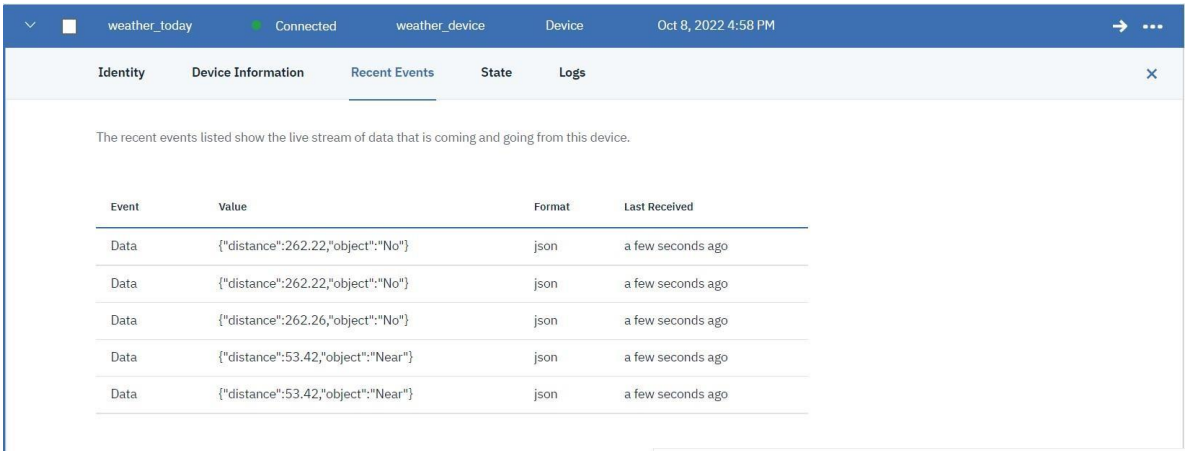
Test Results: All the test cases mentioned above passed successfully. No defects encountered.

9 RESULTS




```
no object found
Sending payload: {"distance":335.90,"object":"No"}
Publish ok
Distancein cm335.90
no object found
Sending payload: {"distance":335.90,"object":"No"}
Publish ok
```


humidityandtemp.py - C:/Python/Python3.10.7/humidityandtemp.py (3.10.7)
File Edit Format Run Options Window Help
import random
while(True):
a=random.randint(10,99)
b=random.randint(10,99)
if(a>35 and b>60):
print("high temperature and humidity of:",a,b,"%","alarm is on")
elif(a<35 and b<60):
print("Normal temperature and humidity of:",a,b,"%","alarm is off")
break
IDLE Shell 3.10.7
File Edit Shell Debug Options Window Help
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:/Python/Python3.10.7/humidityandtemp.py ==========
high temperature and humidity of: 69 90 % alarm is on
high temperature and humidity of: 63 72 % alarm is on
high temperature and humidity of: 76 93 % alarm is on
high temperature and humidity of: 51 68 % alarm is on
high temperature and humidity of: 79 62 % alarm is on
Normal temperature and humidity of: 33 38 % alarm is off
>>>
Ln: 11 Col: 0


weather_today
Connected
weather_device
Device
Oct 8, 2022 4:58 PM
Identity
Device Information
Recent Events
State
Logs
The recent events listed show the live stream of data that is coming and going from this device.
Event
Value
Format
Last Received
Data
{"distance":262.22,"object":"No"}
json
a few seconds ago
Data
{"distance":262.22,"object":"No"}
json
a few seconds ago
Data
{"distance":262.26,"object":"No"}
json
a few seconds ago
Data
{"distance":53.42,"object":"Near"}
json
a few seconds ago
Data
{"distance":53.42,"object":"Near"}
json
a few seconds ago

10 ADVANTAGES & DISADVANTAGES:

ADVANTAGES:

- Lower operating costs
- Increased productivity and workplace safety
- Better customer experiences
- High efficient
- User friendly
- Easy to install

DISADVANTAGES:

- Determining the right process.
- Feeling constrained.
- Spending more than necessary.
- Less efficient
- Covers short area.

11 CONCLUSIONS

IOT based smart agriculture system can prove to be very helpful for farmers since over as well as less irrigation is not good for agriculture. Threshold values for climatic conditions like humidity, temperature, moisture can be fixed based on the environmental conditions of that particular region. The system also senses the invasion of animals which is a primary reason for reduction in crops. This system generates irrigation schedule based on the sensed real time data from field and data from the weather repository. This system can recommend farmer whether or not, is there a need for irrigation. Continuous internet connectivity is required.

12 FUTURE SCOPES:

Indian agriculture is diverse ranging from impoverished farm villages to developed farms utilizing agricultural technologies. Promoting application of modern information technology in agriculture will solve a series of problems facing by farmers. Lack of exact information and communication leads to the loss in production. Our paper is designed to overcome these problems. This system provides an intelligent monitoring platform framework and system structure for facility agriculture ecosystem based on IOT.

13 APPENDIXES:

SOURCE CODE:

```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);

//-------credentials of IBM Accounts------

#define ORG "5qpnhq"//IBM ORGANITION ID
#define DEVICE_TYPE "weather"//Device type mentioned in ibm
watson IOT Platform#define DEVICE_ID " weather1"//Device ID
mentioned in ibm watson IOT Platform #define TOKEN
"?4I@lLnUpZttANL9MsJ&M"
//Token String data3; float dist;

//-------- Customise the above values  char server[] = ORG
".messaging.internetofthings.ibmcloud.com";// Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of event
perform and formatin which data to be send
char subscribetopic[] = "iot-2/cmd/test/fmt/String";// cmd REPRESENT
command type ANDCOMMAND IS TEST OF FORMAT STRING char
authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN; char clientId[] = "d:" ORG
":" DEVICE_TYPE":" DEVICE_ID;//client id
```

```
//.....

WiFiClient wifiClient; // creating the instance for wificlient

PubSubClient client(server, 1883, callback ,wifiClient); //calling the
predefined client id by passingparameter like server id,portand wificredential


int LED =

4; int trig

= 5; int

echo = 18;

void

setup()

{

Serial.begin(11520

0);

pinMode(trig,OUT

PUT);

pinMode(echo,INP

UT);

pinMode(LED,

OUTPUT);

delay(10);

wificonnect();

mqttconnect();

}

void loop()// Recursive Function

{

 digitalWrite(trig,LOW);

digitalWrite(trig,HIGH);
```

```
delayMicroseconds(10);
digitalWrite(trig,LOW);
float dur =
pulseIn(echo,HIGH);
floatdist = (dur *
0.0343)/2; Serial.print
("Distancein cm");
Serial.println(dist);
PublishData(dist);
delay(1000)
;              if
(!client.loop
())            {
mqttconnect
()
}
/*.....retrieving to Cloud .....*/

void PublishData(float dist) {
mqttconnect();//function call for
connecting to ibm
 /*    creating the String in in form JSon to update
the data to ibmcloud
 */
String
object;
if (dist
<100)
```

```
 {
  digitalWrite(LED,HI
GH);
Serial.println("object is
near");object = "Near";
 }
 else
 {
  digitalWrite(LED,LOW
); Serial.println("no object
found");object = "No";
 }
 String payload =
"{\"distance\":";payload
+= dist; payload += ","
"\"object\":\""; payload +=
object; payload += "\"}";
Serial.print("Sending
payload: ");
Serial.println(payload); if
(client.publish(publishTopic,(char*)
payload.c_str())) {
  Serial.println("Publish ok");// if it sucessfully upload data on the cloud then it will print publish ok in Serial monitor or else it will print publish failed
 } else {
  Serial.println("Publish failed");
 }

}
void mqttconnect()
```

```
{ if
(!client.connected()
) {
  Serial.print("Reconnecting client to ");
Serial.println(server);                    while
(!!!client.connect(clientId, authMethod,
token)) {                  Serial.print(".");
                          delay(500);
  }
   initManagedDev
   ice();
   Serial.println();
 }
}
void wificonnect() //function defination for wificonnect
{
 Serial.println();
 Serial.print("Connectin
 g to ");
 WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to
establish the connectionwhile (WiFi.status() != WL_CONNECTED) {
                                          delay(500);
  Serial.print(".");
 }
 Serial.println("");
 Serial.println("WiFi
 connected");
 Serial.println("IP address:
```

```
  ");
  Serial.println(WiFi.localI
  P());
 }
 void initManagedDevice() { if
 (client.subscribe(subscribetopic)) {
 Serial.println((subscribetopic));
   Serial.println("subscribe to cmd OK");
else {
   Serial.println("subscribe to cmd FAILED");
  }

 }
 void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
 {
  Serial.print("callback invoked
 for topic: ");
 Serial.println(subscribetopic);
 for (int i = 0; i < payloadLength;
 i++) {
 //Serial.print((char)payload[i]);
                                  data3
 +=(char)payload[i];
  }
 data3="";
 }
```

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-49745-1660837504

YOUTUBE LINK:

https://www.youtube.com/channel/UCJOKF-hlxQi5uSb3fvg4DrQ