

MODERN CROP PROTECTION USING PYTHON

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

by

R.SURYAVAMSI(38110437)

P.Y.S.RAJA(38110444)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by
NAAC JEPPIAAR NAGAR, RAJIV GANDHI
SALAI, CHENNAI – 600119**

MARCH-2022



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai-600119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONA FIDE CERTIFICATE

This is to certify that this project report is the bonafide work **R.SURYAVAMSI(38110437)** and **P.Y.S.RAJA(38110444)** who carried out the project entitled "MODERN CROP PROTECTION" under my super vision from **NOVEMBER 2021** to **MARCH 2022**.

Internal Guide

Dr.L.LAKSHMANAN

Head of the Department

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I R.suryavamsi (38110437) hereby declare that the Project Report entitled **MODERN CROP PROTECTION USING MACHINE LEARNING** done by me under the guidance of **Dr.L.Lakshmanan M.E., (Ph. D)** (Internal) is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering / Technology degree in Computer Science and Engineering.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph. D, Dean**, School of Computing **Dr. L. Lakshmanan M.E., Ph.D.**, and **Dr.S. Vigneshwari M.E., Ph.D. Heads** of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.L.lakshmanan M.E., (Ph. D)** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Surveillance plays a major role in many fields be it at home, hospitals, schools, public places ,farmlands etc. It helps us to monitor a certain area and prevent theft and also provides proof of evidence. In the case of farmlands or agricultural lands surveillance is very important to prevent unauthorized people from gaining access to the area as well as to protect the area from animals .Various methods aim only at surveillance which is mainly for human intruders, but we tend to forget that the main enemies of such farmers are the animals which destroy the crops. This leads to poor yield of crops and significant financial loss to the owners of the farmland. This problem is sopronounced that sometimes the farmers decide to leave the areas barren due to such frequent animal attacks. This system helps us to keep away such wild animals from the farmlands as well as provides surveillance functionality.

LISTOF FIGURES

FIGURE NO	FIGURE NAME	PAGENO
4.1	Raspberry PI	19
4.2	Designer panel	51
4.3	live testing	52
4.4	Mobile application	53
5.1	image capture	55
Output Screenshots		87-88

TABEL OF CONTENTS

CHAPTER	TITLE	PAGE NO
	Abstract	I
	List of figures	II
	List of abbreviations	1
1.	INTRODUCTION	2
2.	LITERATURE SURVEY	3
3	AIM AND SCOPE	4
4	METHODOLAGY AND	
MPLEMENTATION	5- 5 3	
		5
	4.1 TECHNOLOGY USED	5
	4.1.1 SKILLS REQUIRED	5
	4.1.2SOFTWARE REQUIREMENTS	5
	4.1.3 HARDWARE REQUIRMENTS	5
	4.2 INTRODUCTION OF INTERNET OF THINGS	6
	4.3ISSUES RAISED BYINTERNET OF THINGS	11
	4.4 SECURITY ISSUES	11
	4.5 IOT SECURITY QUESTIONS	12
	4.6 RASPBERRY PI	16
	4.7 IBM WATSON CLOUD	20

4.8 IBM IOT PLATFORM	25
4.9 NODE RED CREATION	29
4.10 MIT INVENTOR	50

CHAPTER	TITLE	PAGE NO
5	RESULT AND FUTURE WORKS	54
	5.1 RESULT	
	5.2 FUTURE WORKS	
6	CONCLUSION	67
	REFERENCE	68
	APPENDIX	
	CODE	69-86
	OUTPUT SCREENSHOTES	87-88

LIST OF ABBRETIATIONS

MIT	Massachusetts Institute of Technology
IoT	Internet of Things
USB	Universal Serial Bus
IBM	International Business Machine
DC	Direct current
DHT	Digital Humidity And Temperatures

CHAPTER 1

INTRODUCTION

Detecting the different animal and birds using IBM visual recognition service for protection the crop from damage. There will be live video streaming in raspberry pi and in this video streaming we can detect the animal and birds. Whenever any birds and animals are detected the image will be captured and for keeping away then from the crop we can keep a siren and we can blink the LED's. The camera will be attached to the servomotor it will be rotating for every time interval. The crop is also integrated with the soil moisture sensor and the DHT sensor for temperature and humidity parameters. Whenever there is low soil moisture the admins will be alerted and they can blink the LED's. Then it sounds an alarm to woo the animals away from the fields as well as sends sms to the farmer so that he may know about the issue and come to the spot in case the animals don't turn away from the alarm. This ensures complete safety of crops from animals thus protecting the farmers loss.

CHAPTER 2

LITERATURE SURVEY

Balaji Banu [1] designed a wireless sensor networks to observe the conditions of the farming and increasing the crop yield and quality. Sensors are used to monitor different conditions of environment like water level, humidity, temperature etc., The processors ATMEGA8535 and ICS8817 BS, analog to digital conversion and wireless sensor nodes with wireless transceiver module based on Zig bee protocol are used in the designing the system. Database and web application is used to retrieve and store data. In this experiment the sensor node failure and energy efficiency are managed. Liu Dan [2], Joseph Haule, Kisangiri Michael [3] and Wang Weihong, Cao Shuntian [38] carried out experiments on intelligent agriculture greenhouse monitoring system based on ZigBee technology. The system performs data acquisition, processing, transmission and reception functions. The aim of their experiments is to realize greenhouse environment system, where the of system efficiency to manage the environment area and reduce the money and farming cost and also save energy. IOT technology here is based on the B-S structure and cc2530 used like processing chip to work for wireless sensor node and coordinator. The gateway has Linux operating system and cortex A8 processor act as core. Overall the design realizes remote intelligent monitoring and control of greenhouse and also replaces the traditional wired technology to wireless, also reduces manpower cost. Joseph haule [3], Dragoş Mihai Ofrim, Bogdan Alexandru Ofrim and Dragoş Ioan Săcăleanu [18] have proposed an experiment that explains the use of wsn used in automating irrigation. Irrigation control and rescheduling based on wsn are powerful solutions for optimum water management through automatic communication to know the soil moisture conditions of irrigation design. The process used here is to determine the proper frequency and time of watering are important to ensure the efficient use of water, high quality of crop detection delay throughput and load. Simulation is done for agriculture by OPNET. Another design of wsn is deployed for irrigation system using Zig bee protocol which will impact battery life. There are some drawbacks as wsn is still under development stage with unreliable communication times, fragile, power consumption and communication can be lost in agricultural field. so automate irrigation system and scheduling based on wireless sensor networks are used. WSN uses low power and a low data rate and hence energy efficient technology. All the devices and machines controlled with the help of inputs received via sensors which are mixed with soil. Farmers can analyze whether the system performs in no

CHAPTER 3

AIM AND SCOPE

3.1 AIM

Detecting the different animal and birds using IBM visual recognition service for protection the crop from damage.

3.2 SCOPE

This project is used to protect the farmland by using Raspberry Pi this project utilizes recognition service for this purpose. Forest officer and farmer will get the message connecting area which that animals and birds observe.

Detecting the different animal and birds using IBM visual recognition service for protection the crop from damage. There will be live video streaming in raspberry pi and in this video streaming we can detect the animal and birds. Whenever any birds and animals are detected the image will be captured and for keeping away then from the crop we can keep a siren and we can blink the LED's. The camera will be attached to the servomotor it will be rotating for every time interval. The crop is also integrated with the soil moisture sensor and the DHT sensor for temperature and humidity parameters. Whenever there is low soil moisture the admins will be alerted and they can blink the LED's. Then it sounds an alarm to woo the animals away from the fields as well as send SMS to the farmer so that he may know about the issue and come to the spot in case the animals don't turn away from the alarm. This ensures complete safety of crops from animals thus protecting the farmers loss

CHAPTER 4

METHODOLOGY AND IMPLEMENTATION

4.1 TECHNOLOGY USED

4.1.1 SKILLS REQUIRED

- IOT Open Hardware Platforms
- IOT Application Development
- IOT Cloud Platform
- IOT Communication Technologies

4.1.2

SOFTWARE REQUIREMENTS

- Python
- IBM Watson
- Node Red

4.1.3

HARDWARE REQUIREMENTS

- DHT11 Sensor
- Soil Moisture Sensor
- Servo Motor
- 9VDC Motor
- Camera
- Connecting Wires

- Raspberry pi

4.2

INTRODUCTION OF INTERNET OF THINGS:

Internet of Everything or Network of Everything is additionally known as Internet of Things (IoT). When physical objects or things are embedded with physics, sensors and software then the network called IoT is formed. The Internet of Things (IoT) is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.

The Internet of Things is an emerging topic of technical, social, and economic significance. Consumer products, durable goods, cars and trucks, industrial and utility components, sensors, and other everyday objects are being combined with Internet connectivity and powerful data analytic capabilities that promise to transform the way we work, live, and play. Projection for the impact of IoT on the Internet and economy are impressive, with some anticipating as many as 100 billion connected IoT devices and a global economic impact of more than \$11 trillion by 2025.

At the same time, however, the Internet of Things raises significant challenges that could stand in the way of realizing its potential benefits. Attention-grabbing headlines about the hacking of Internet-connected devices, surveillance concerns, and privacy fears already have captured public attention. Technical challenges remain and new policy, legal and development challenges are emerging.

This overview document is designed to help the Internet Society community navigate the dialogues surrounding the Internet of Things in light of the competing predictions about its promises and devices that have existed for decades. The recent confluence of several technology market trends, however, is bringing the Internet of Things closer to widespread reality. These include Ubiquitous Connectivity, Widespread Adoption of IP-based Networking, Computing Economics, Miniaturization, Advances in Data Analytics and the Rise of Cloud Computing.

- **Connectivity Models:**

IoT implementations use different technical communications models, each with its own characteristics. Four common communications models described by the Internet Architecture Board include: Device-to-Device, Device-to-Cloud, Device-to-Gateway, and Back-End Data-Sharing. These models highlight the flexibility in the ways that IoT devices can connect and provide value to the user.

- **Transformational Potential:**

The implications and issues in a world where the most common interaction with the Internet comes from passive engagement with connected objects rather than active engagement with content. The potential realization of this outcome – a “hyper connected world” -- is testament to the general-purpose nature of the Internet architecture itself, which does not place inherent limitations on the applications or services that can make use of the technology.

Five key IoT issue areas are examined to explore some of the most pressing challenges and questions related to the technology. These include security; privacy; interoperability and standards; legal, regulatory, and rights; and emerging economies and development.

- **Security:**

While security considerations are not new in the context of information technology, the attributes of many IoT implementations present new and unique security challenges.

Addressing these challenges and ensuring security in IoT products and services must be fundamental priority.. Poorly secured IoT devices and services can serve as potential entry points for cyber attack and exposure data to theft by leaving data streams inadequately protected.

The interconnected nature of IoT devices means that every poorly secured device that is connected online potentially affects the security and resilience of the Internet globally. This challenge is amplified by other considerations like the mass-scale connect to other devices, and the likelihood of fielding these devices in secure environments.

As a matter of principle, developers and users of IoT devices and systems have a collective obligation to ensure they do not expose users and the Internet itself to potential harm. Accordingly, collaborative approach to security will be needed to develop effective and appropriate solutions to IoT security challenges that are well suited to the scale and complexity of the issues.

Privacy:

The full potential of the Internet of Things depends on strategies that respect individual privacy choices a

specificity afforded by IoT devices can unlock incredible value to IoT users, but concerns about privacy and potential harms might hold back full adoption of the Internet of Things. This means that privacy rights and respect for user privacy expectations are integral to ensuring user trust and confidence in the Internet, connected devices, and related services.

IoT amplifies concerns about the potential for increased surveillance and tracking, difficulty in being able to opt out of certain data collection, and the strength of aggregating IoT data streams to paint a detailed digital portrait for users. While these are important challenges, they are not insurmountable.

- **Interoperability/Standards:**

A fragmented environment of proprietary IoT technical implementations will inhibit value for users and industry. While full interoperability across products and services is not always feasible or necessary, purchasers may be hesitant to buy IoT products and services if there is integration inflexibility, high ownership complexity, and concern over vendor lock-in.

In addition, poorly designed and configured IoT devices may have negative consequences for the networking resources they connect to and the broader Internet. Appropriate standards reference models, and best practices also will help curb the proliferation of devices that may act in disrupted ways to the Internet.

- **Legal, Regulatory and Rights:**

The use of IoT devices raises many new regulatory and legal questions as well as amplifies existing legal issues around the Internet. The questions are wide in scope, and the rapid rate of change in IoT technology frequently outpaces the ability of the associated policy, legal, and regulatory structures to adapt.

One set of issues surrounds cross border data flows, which occur when IoT devices collect data about people in one jurisdiction and transmit it to another jurisdiction with different data protection laws for processing. Further, data collected by IoT devices is sometimes susceptible to misuse, potentially causing discriminatory outcomes for some users.

While the legal and regulatory challenges are broad and complex in scope, adopting the guiding Internet Society principles of promoting a user's ability to connect, speak, innovate, share, c

- **Emerging Economy and Development Issues:**

The Internet of Things holds significant promise for delivering social and economic benefits to emerging and developing economies. This includes areas such as sustainable agriculture, water quality and use, healthcare, industrialization, and environmental management, among others. As such, IoT holds promise as a tool in achieving the United Nations Sustainable Development Goals.

The broad scope of IoT challenges will not be unique to industrialized countries. Developing regions also will need to respond to realize the potential benefits of IoT. In addition, the unique needs and challenges of implementation in less-developed regions will need to be addressed, including infrastructure readiness, market and investment incentives, technical skill requirements, and policy resources.

The Internet of things (IoT) is the network of physical devices, vehicles, home appliances and other items embedded with electronics, software, sensors, actuators and network connectivity which enable these objects to connect and exchange data. Each thing is uniquely identifiable through its embedded computing system but is able to inter-operate with the existing Internet infrastructure. Experts estimate that the IoT will consist of about 30 billion objects by 2020. It is also estimated that the global market value of IoT will reach \$7.1 trillion by 2020. The IoT allows objects to be sensed or controlled remotely across the existing network infrastructure, creating opportunities for more direct integration of the physical world into computer-based systems, and resulting in improved efficiency, accuracy and economic benefit in addition to reduced human intervention.

"Things", in the IoT sense, can refer to a wide variety of the devices. Which such as the heart monitoring implants, biochip transponders on the farm animals, cameras streaming live feeds of wild animals in coastal waters, automobiles with built-in sensors, DNA analysis devices for environmental/food/pathogen monitoring, or field operation devices that assist fire fighters in search and rescue.

operations. Legal scholars

suggest regarding "things" as an "inextricable mixture of hardware, software, data and service".

These devices collect useful data with the help of various existing technologies and then autonomously flow the data between other devices. The term "the Internet of things

" was coined by the Kevin Ashton of Procter & Gamble, later MIT's Auto-ID Centre, in 1999.

WHAT IS THE INTERNET OF THINGS?

The term "Internet of Things" (IoT) was first used in 1999 by British technology pioneer Kevin Ashton to describe a system in which objects in the physical world could be connected to the Internet by sensors. Ashton coined the term to illustrate the power of connecting Radio-Frequency Identification (RFID) tags used in corporate supply chains to the Internet in order to count and track goods without the need for human intervention. Today, the Internet of Things has become a popular term for describing scenarios in which Internet connectivity and computing capability extend to a variety of objects, devices, sensors, and everyday items.

While the term "Internet of Things" is relatively new, the concept of combining computers and networks to monitor and control devices has been around for decades. By the late 1970s, for example, systems for remotely monitoring meters on the electrical grid via telephone lines were already in commercial use. In the 1990s, "machine-to-machine" (M2M) enterprise and industrial solutions for equipment monitoring and operation to become widespread. Many of these early M2M solutions, however, were based on closed purpose-built networks and proprietary or industry-specific standards rather than on Internet Protocol (IP)-based networks and Internet standards.

Using IP to connect devices other than computers to the Internet is not a new idea. The first Internet "device" an IP-enabled toaster that could be turned on and off over the Internet was featured at an Internet conference in 1990. Over the next several years, other "things" were IP-enabled, including a soda machine at Carnegie Mellon University in the US and a coffee pot¹⁸

in the Trojan Room at the University of Cambridge in the UK (which remained Internet-connected until 2001). From these whimsical beginnings, a robust field of research and development into “smart object networking” helped create the foundation for today’s Internet of Things. At present, a wide range of industry sectors including automotive, healthcare, manufacturing, home and consumer electronics, and well beyond are considering the potential for incorporating IoT technology into their products, services and operations.

4.3

- **WHAT ISSUES ARE RAISED BY INTERNET OF THINGS?**

It would be impossible to cover the broad scope of issues surrounding the Internet of Things in a single paper. Below, however, we provide an overview of five topics frequently discussed in relation to IOT. These include: security; privacy; interoperability and standards; legal, regulatory and rights; and emerging economies and development.

We begin to examine these issues through the lens of “the Abilities” – the statement of fundamental principles that guide ISOC’s work in terms of the capabilities we believe all Internet users should enjoy that must be protected.

These include the ability to connect, speak, innovate, share, choose, and trust. With these principles as a guide, we present important aspects of each issue and propose several questions for discussion.

4.4

- **SECURITY ISSUES**

As we note in the principles that guide our work, ensuring the security, reliability, resilience, and stability of Internet applications and services is critical to promoting trust and use of the Internet. As users of the Internet, we need to have a high degree of trust that the Internet, its applications, and the devices linked to it are secure enough to do the kinds of activities we want to do online in relation to the risk tolerance associated with those activities. The Internet of Things is no different in this respect, and security in IOT is fundamentally linked to the ability of users to trust their environment. If people don’t believe their connected devices and their information are reasonably

Secure from misuse or harm, the resulting erosion of trust causes a reluctance to use the Inter

speech, and practically every other aspect of online activities. Indeed, ensuring security in IoT products and services should be considered a top priority for the sector.

As we increasingly connect devices to the Internet, new opportunities to exploit potential security vulnerabilities expose user data to theft by leaving data streams inadequately protected. Failures in malfunctioning devices also can create security vulnerabilities. These problems are just as large or larger for the small, cheap, and ubiquitous smart devices in the Internet of Things as they are for the computers that have traditionally been the endpoints of Internet connectivity. Competitive cost and technical constraints on IoT devices challenge manufacturers to adequately design security features into these devices, potentially creating security and long-term maintainability vulnerabilities greater than their traditional computer counterparts.

Along with potential security design deficiencies, the sheer increase in the number and nature of IoT devices could increase the opportunities of attack. When coupled with the highly interconnected nature of IoT devices, every poorly secured device that is connected online potentially affects the security and resilience of the Internet globally, not just locally. For example, an unprotected refrigerator or television in the US that is infected with malware might send thousands of harmful spam emails to recipients worldwide using the owner's home Wi-Fi Internet connection.

To complicate matters, our ability to function in our daily activities without using devices or systems that are Internet-enabled is likely to decrease in a hyper connected world. In fact, it is increasingly difficult to purchase some devices that are not Internet-connected because certain vendors only make connected products. Day by day, we become more connected and dependent on IoT devices for essential services, and we need the devices to be secure, while recognizing that no device can be absolutely secure. This increasing level of dependence on IoT devices and the Internet services they interact with also increases the pathways for wrongdoers to gain access to devices. Perhaps we could unplug our Internet-connected TVs if they get compromised in a cyber attack, but we can't so easily turn off a smart utility power meter or a traffic control system or a person's implanted pacemaker if they fall victim to malicious behavior. This is why security of IoT devices and services is a major discussion point and should be considered a critical issue. We increasingly depend on these devices for essential services, and their behavior may have global reach and impact.

4.5

IoT Security Questions

A number of questions have been raised regarding security challenges posed by Internet of Things devices. Many of these questions existed prior to the growth of IoT,

Good Design Practices:

What are the sets of best practices for engineers and developers to use to design IoT devices to make them more secure?

How do lessons learned from Internet of Things security problems get captured and conveyed to development communities to improve future generations of devices? What training and educational resources are available to teach engineers and developers more secure IoT design?

- **Cost vs. Security Trade-Offs:**

How do stakeholders make informed cost-benefit analysis decisions with respect to Internet of Things devices? How do we accurately quantify and assess the security risks? What will motivate device designers and manufacturers to accept additional product design cost to make devices more secure, and, in particular, to take responsibility for the impact of any negative externalities resulting from their security decisions? How will in-between functionality and usability be reconciled with security? How do we ensure IoT security solutions support opportunities for IoT innovation, social and economic growth?

- **Standards and Metrics:**

What is the role of technical and operational standards for the development and deployment of secure, well-behaving IoT devices? How do we effectively identify and measure characteristics of IoT device security?

How do we measure the effectiveness of Internet of Things security initiatives and countermeasures? How do we ensure security best solution to prevent eavesdropping and hijacking attacks of the data streams these devices produce? Which encryption and authentication technologies could be adapted for the Internet of Things, and how could they be implemented.

- **Field-Upgradeability:**

With an extended service life expected for many IoT devices, should devices be designed for maintainability and upgradeability in the field to adapt to evolving security threats?

New software and parameter settings could be installed in a fielded IoT device by a centralized security management system if each device had an integrated device management agent. But management systems add cost and complexity; could other approaches to upgrading device software be more compatible with widespread use of IoT devices? Are there any classes of IoT devices that are low-risk and therefore don't warrant these kinds of features? In general, are the user interfaces IoT devices expose (usually intentionally minimal) being properly scrutinized with consideration for device management (by anyone, including the user)?

- **Regulation:**

Should device manufacturers be penalized for selling software or hardware with known or unknown security flaws? How might product liability and consumer protection laws be adapted or extended to cover any negative externalities related to the Internet of Things and would this operation across-border environment?

Would it be possible for regulation to keep pace and be effective in light of evolving IoT technology and evolving security threats? How should regulation be balanced against the needs of permission-less innovation, Internet freedom, and freedom of expression?

Device Obsolescence:

What is the right approach to take with obsolete IoT devices as the Internet evolves and security threats change? Should IoT devices be required to have a built-in end-of-life expiration feature that disables them? Such a requirement could force older, non-interoperable devices out of service and replace them with more secure and interoperable devices in the future. Certainly, this would be very challenging in the open marketplace. What are the implications of automatic decommissioning IoT devices?

The breadth of these questions is representative of the wide-ranging security considerations associated with Internet of Things devices. However, it's important to remember that when a device is on the Internet, it is also part of the Internet which means that effective and appropriate security solutions can be achieved only if the participants involved

4.6

RASPBERRY PI

The Raspberry Pi is a credit card-sized computer that plugs into your TV and a keyboard. It is a capable little computer which can be used in electronics projects, and for many of the things that your desktop PC does, like spreadsheets, word processing, browsing the internet, and playing games. It also plays high-definition video. We want to see it being used by kids all over the world to learn programming.

The Raspberry Pi 3 is the third generation Raspberry Pi. It replaced the Raspberry Pi 2 Model B in February 2016. Compared to the Raspberry Pi 2 it has:

- A 1.2GHz 64-bit quad-core ARMv8 CPU
- 802.11n Wireless LAN
- Bluetooth 4.1
- Bluetooth Low Energy (BLE)
- Like the Pi2, it also has:
- 1GB RAM
- 4 USB ports
- 40 GPIO pins
- Full HDMI port
- Ethernet port
- Combined 3.5mm audio jack and composite video
- Camera interface (CSI)
- Display interface (DSI)
- MicroSD card slot (now push-pull rather than push-push)
- VideoCore IV 3D graphics core

The Raspberry Pi 3 has an identical form factor to the previous Pi 2 (and Pi 1 Model B+) and has complete compatibility with Raspberry Pi 1 and 2.

it's the one most people will want to start with. Raspbian is a version of Linux built specifically for the Raspberry Pi. It comes packed with all the software you'll need for every basic task with a computer. Basic examples to help you get started with some of the software available in Rasp \.

Contents

- **Scratch**
 - Scratch is a visual programming tool allowing the user to create animations and games with a drag-and-drop interface
- **Python**
 - Python is a general purpose programming language that is easy to pick up and also very powerful
- **SonicPi**
 - Write code to make music with SonicPi
- **Terminal**
 - The Linux terminal is a powerful tool for getting around the file system and issuing commands
- **GPIO: Raspberry Pi Models A and B**
 - The General Purpose Input/Output pins allow you to control and interact with the real world from Python, Scratch or other programming environments
- **GPIO: Models A+, B+, Raspberry Pi 2B and Raspberry Pi 3B**
 - The General Purpose Input/Output pins allow you to control and interact with the real world from Python, Scratch or other programming environments
- ❓ **Minecraft**
 - A free version of Minecraft is available for the Pi, and this is the only edition that has a programming interface, meaning you can control Minecraft with Python code - and even interact with the real world through GPIO
- ❓ **Python Games**
 - Raspbian ships with some ready-made Python games for you to play on your Raspberry Pi - and why not browse the code, hack the games and learn to make your own?
- ❓ **Word Press**

- Setup a web server on your Raspberry Pi and install Wordpress, a content management and blogging system
- *Mathematica*
 - Mathematica is an industry leading computational platform - available for free on Raspberry Pi
- *Camera Module*
 - The Raspberry Pi camera module is capable of taking full HD 1080p photo and video and can be controlled programmatically
 - Libraries are available for:
 - Bash (Linux command line)
 - Python
- *Webcams*
 - Using a standard USB webcam instead of the Raspberry Pi camera module
- *Kodi*
 - Installing media centre software on your Raspberry Pi
- *Playing audio*
 - Playing audio on your Raspberry Pi
- *Playing video*
 - Playing video on your Raspberry Pi
- *Demo programs*
 - A selection of examples to demonstrate the Pi's capabilities

4.1

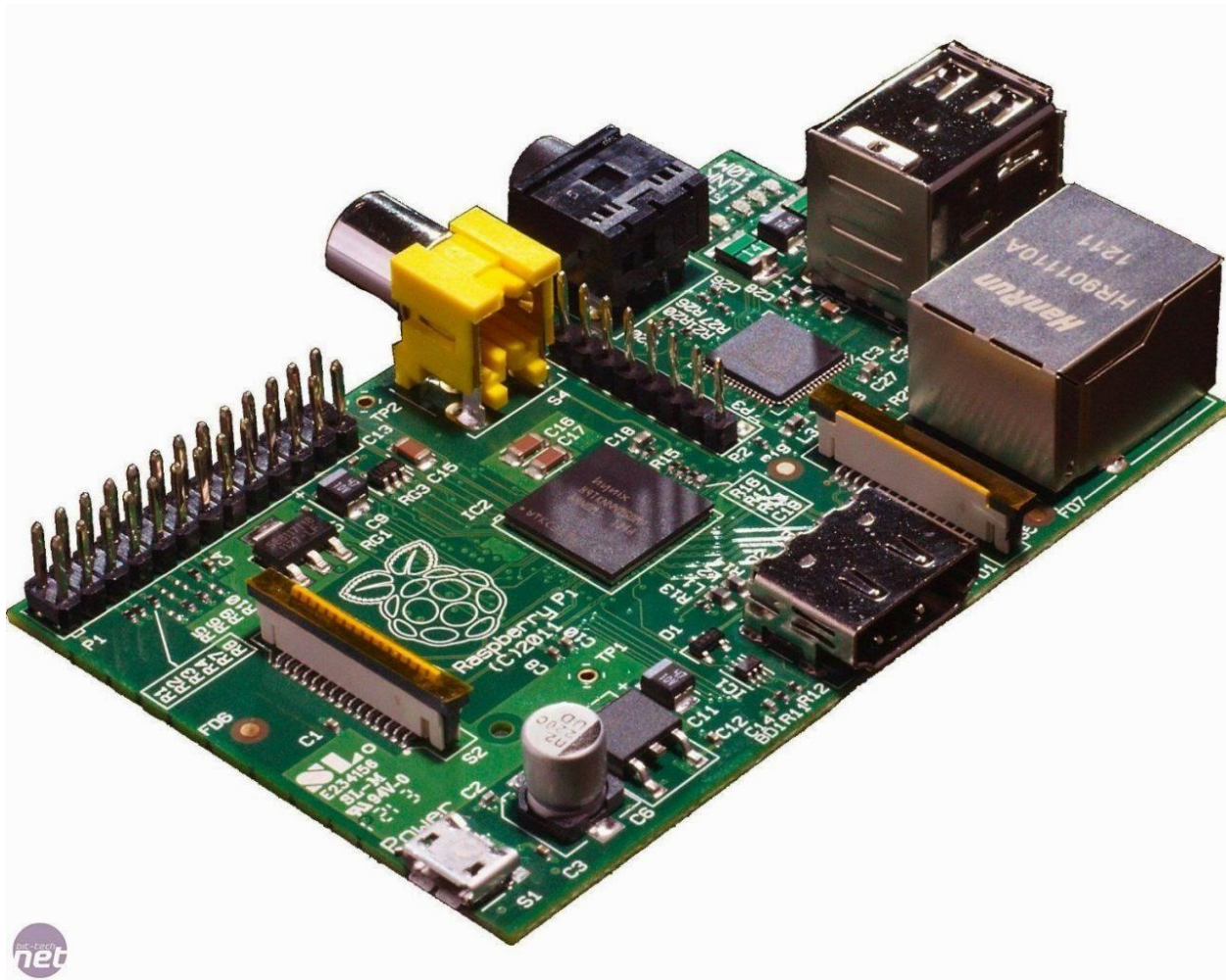


Fig: Raspberry pi

4.7

IBM Watson Cloud

- **Creation of IBM account:**
 - Type <https://cloud.ibm.com/login> in Browser.
 - Click on sign up icon
 - Sign up email Id and password.
 - Click on Next and Verification sent to your mail.
 - If you click on verify account.
 - Next it ask's you to enter some personal details
 - It re direct to IBM account main page.



Already have an IBM Cloud account? [Log in](#)

Create an account

1. Account information

Email

Password

Next



Already have an IBM Cloud account? [Log in](#)

Create an account

1. Account information ✓

Edit

~~XXXXXXXXXXXX@EXAMPLE.COM~~
.....

2. Verify email

We sent a 7-digit verification code to ~~XXXXXXXXXXXX@EXAMPLE.COM~~. This code is valid for 30 minutes. Want to create this account with a different email? [Edit email](#).

Verification code

Verification code is required.

Next



[Resend code](#)

3. Personal information

Create account



Create an account

1. Account information ✓

Edit 

~~pathi.ritika@gmail.com~~
.....

2. Verify email ✓

Edit 

Your email is verified.

3. Personal information

Lite plan services are always free, guaranteed. You'll be charged only if you use any billable plans. [Learn more](#)

First name

Last name

Country or region

India



Next



Create account



About your IBMid Account Privacy

This notice provides information about accessing your IBMid user account (Account). If you have previously been presented with a version of this notice, please refer to "Changes since the previous version of this notice" below for information about the new updates. Updates to the IBM Privacy Statement since this notice was originally published provide additional information about how your personal information is processed by IBM.

- + Changes since the previous version of this notice
- + What data does IBM collect?
- + Why IBM needs your data
- + How your data was obtained
- + How IBM uses your data
- + How IBM protects your data
- + How long we keep your data

Your rights

Our Privacy Statement provides more information about your personal data rights. It also provides contact information if you have questions or concerns regarding our handling of your personal data.

Acknowledgement

I acknowledge that I understand how IBM is using my Basic Personal Data and I am at least 16 years of age.

[Proceed](#)

[Cancel Sign In](#)

This document was last updated on 2020-03-05

IBM cloud account main page.

4.8

IBM IOT PLATFORM

4.8.1

Introduction:

The IBM cloud platform combines platform as a service (PaaS) with infrastructure as a service (IaaS) to provide an integrated experience. The platform scales and supports both small development teams and organizations, and large enterprise businesses. Globally deployed across data centers around the world, the solution you build on IBM Cloud spins up fast and performs reliably in a tested and supported environment you can trust. The platform is built to support your needs whether it's working only in the public cloud or taking advantage of a multi cloud deployment model. With open source technologies, such as Kubernetes, Red Hat OpenShift, and a full range of compute options, including virtual machines, containers, bare metal, and serverless, you have as much control and flexibility as you need to support workloads in your hybrid environment. You can deploy cloud-native apps while also ensuring workload portability.

4.8.2

Setting up your account:

- User access groups for organizing users and service IDs into one entity to make assigning access a streamlined process.
- The enterprise account is the parent account to all other accounts in the enterprise. Billing for the enterprise is managed at the enterprise account level.
- Account groups provide a way to organize related accounts. And, you get a unified view of resource usage costs across all accounts that are included in an account group.
- Resource groups for organizing your resources to make assigning access to a set of resources quick and easy.
- Access policies for your access groups or individual developers who need IAM access policies or Cloud Foundry org and space roles.

4.8.3

IBM Cloud catalog:

IBM Cloud provides a full-stack, public cloud platform with a variety of products in the catalog, including compute, storage, and networking options, end-to-end developer solutions for app development, testing and deployment, security management services, traditional and open-source databases, and cloud-native services. You can find all of these services on the Services tab in the catalog. The lifecycle and operation of these services are the responsibility of IBM.

The Software tab includes a growing catalog of software products, including, cloudpacks, starter kits, Terraform-based templates, and Helm charts. Even though you're responsible for the lifecycle management, deployment, and configuration of these software products on your own compute resources, you can take advantage of a simplified installation process that helps you to get up and running quickly. The catalog supports command-line interfaces (CLIs) and a REST API for users to retrieve information about existing products, and create, manage, and delete their resources.

4.8.4

Creating resources

The resource controller is the next-generation IBM Cloud platform provisioning layer that manages the lifecycle of IBM Cloud resources in your account. Resources are provisioned globally in an account scope. The resource controller supports both synchronous and asynchronous provisioning of resources. Examples of resources include databases, accounts, processors, memory, and storage limits. In general, resources that are tracked by the provisioning layer are intended to associate usage metrics and billing, but that isn't always the case. In some cases, the resource might be associated with the provisioning layer to ensure that the resource lifecycle can be managed along with the account lifecycle. The resource controller uses IBM Cloud Identity and Access Management (IAM) for authentication and authorization of actions that are taken against the provisioning layer.

Managing your resources:

A collection of resources is managed by resource groups. A resource group is associated with your account. All IBM Cloud resources must be assigned to a resource group. When you create an account, a default resource group is created for you. All IBM Cloud IAM-enabled resources must be provisioned within a resource group. If you have a Lite account, you can have only one resource group. If you have a Pay-As-You-Go or Subscription account, you can create more than one resource group. If an account is suspended, the corresponding resource group is suspended as well, and all resources within the resource group are suspended.

Searching and tagging resources:

- The search service is a global and shared resource properties repository that is integrated within the IBM Cloud platform. It is used for storing and searching a cloud resource's attributes, and it categorizes and classifies resources.
- Resources are uniquely identified by a cloud resource name (CRN) identifier. The properties of a resource include tags and system properties. Both properties are defined within an IBM Cloud billing account, and span across many regions.
- This service also manages tags that are associated with a resource. You can create, delete, search, attach, or detach tags with the Tagging API. Tags are uniquely identified by a CRN identifier. Tags have a name, which must be unique within a billing account. You can create tags in key-value pairs or label format.

4.8.5

Monitoring your resources:

Observability offers a single location where you can monitor and observe your applications and services in IBM Cloud.

With the IBM® Log Analysis with Log DNA service, you can add log management capabilities to your IBM Cloud architecture and you can manage system and application logs. It offers advanced features to monitor and troubleshoot, define alerts, and design custom dashboards. IBM Log Analysis with Log DNA is operated by Log DNA in partnership with IBM.

The IBM Cloud Monitoring with Sysdig service, allows you to gain operational visibility into the performance and health of your applications, services, and platforms. It offers a full stack telemetry with advanced features to monitor and troubleshoot, define alerts, and design custom dashboards. IBM Cloud Monitoring with sysdig is operated by Sysdig in partnership with IBM.

4.9

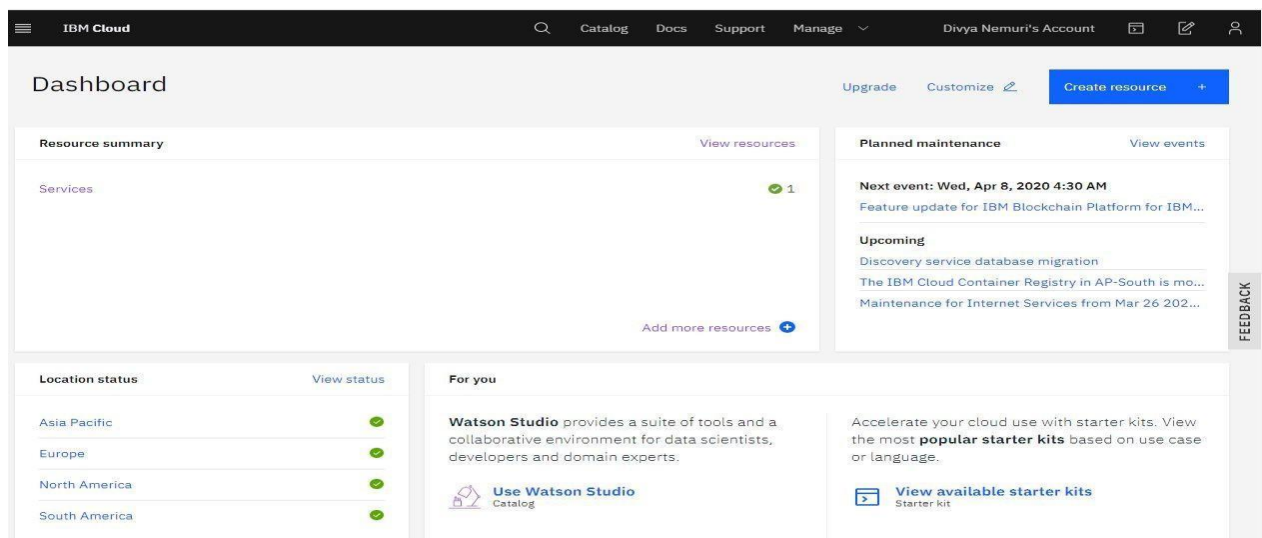
Node-red Creation

Brief: This tutorial will guide you in creating a device in the IBM IoT platform. Outcome: In this tutorial, you will learn the following activities and tasks.

- Creating a node-red application
- Launch and configure the node-red application
- Basic flow on how to use node-red application.
- Installing dashboard nodes for creating a web UI

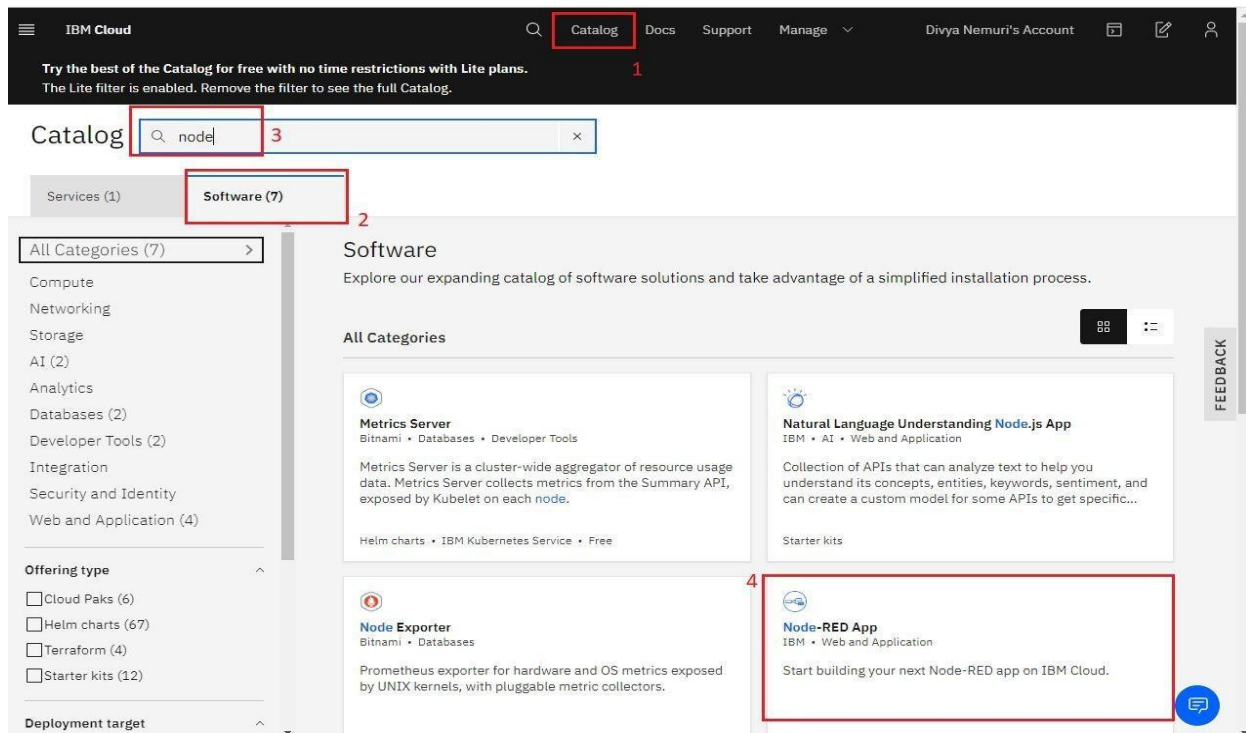
Activity:

Log in to Your IBM account using <https://cloud.ibm.com/>. Upon successful login to the cloud, you will be on the dashboard page. If you are a new user you won't have any resources listed. For a returning user, you will have your previous resources listed.



Task-1:CreateNode-redapplication

- Click on catalogon the dashboard
- Click on the software to find the application
- Search for node- redintheseachbar provided
- Click on the node-red application to create



- After clicking on the node-red app you will be redirected to the node-red application starter kit page. Click on create

The screenshot displays the IBM Cloud interface for the Node-RED starter kit. At the top, the navigation bar includes 'IBM Cloud', a search icon, and links for 'Catalog', 'Docs', 'Support', and 'Manage'. The user's account, 'Divya Nemuri's Account', is also visible. Below the navigation bar, the breadcrumb 'Catalog / Starter kit details' is shown. The main header features the Node-RED logo and the text 'Node-RED Code Pattern • Lite'. A blue 'Create app' button is prominently displayed and highlighted with a red rectangular box. The left sidebar contains sections for 'Details' (listing author as IBM, updated on 2/11/2020, and type as Starter kit), 'Source code' (with a GitHub link), and 'Helpful links' (with a Tutorial link). The main content area includes an 'Overview' section describing the starter kit's purpose, a 'This starter kit will help you' section with three bullet points, a 'What's included?' section, and a 'Services (1)' section featuring the Cloudant service with links for 'Free to start', 'View pricing', and 'View docs'. A vertical sidebar on the right contains 'FEEDBACK' and 'ASK A QUESTION' links.

To create the node-red application give the App name. The name should be unique. One may use the default name provided or can be of their choice. Select the region as Dallas for avoiding the conflict on installing the nodes in the future. Then click on create

- To create the app it will take some time while the cloudant service will be created

IBM Cloud

Catalog / Starter kit details / Create app

Node-RED

App details

App name
Accept the default name, or enter a value up to 128 characters.

Node RED BRZHK

Resource group
Default

Tags
Examples: env:dev, version:1

Platform
☒ Node.js

Region
Chennai
Washington DC
Seoul
Dallas
Sydney
Frankfurt
Tokyo
London
Chennai

Resource group
Default

Getting started with apps

[View source code](#)

After you create your app, you can add services and use a DevOps toolchain to set up Continuous Delivery for deploying your app to IBM Cloud.

FEEDBACK

ASK A QUESTION

cloud.ibm.com/developer/appservice/apps/688137d6-7271-40bf-bf90-016cae81a9f5

Apps Gmail YouTube Maps Translate

IBM Cloud Catalog Docs Support Manage Divya Nemuri's Account

Resource list / App details

Node RED BRZHK Add tags Resource group: Default Download code

App details Developer resources

Services (1) Connect existing services Create service

Name	Resources	Actions
Cloudant	Provisioning service credentials	

Configure Continuous Delivery

Continuous Delivery is not enabled for this app. Enable Continuous Delivery to automate builds, tests, and deployments through Delivery Pipeline, GitLab, and more.

Deploy your app

Credentials Show

Knowledge Guide

Next steps

To enable Continuous Delivery and deploy to IBM Cloud:

1. Click **Deploy your app**.
2. Select a deployment target, select the toolchain settings, and click **Create**. The deployment starts automatically.
3. Check the deployment status in the **Delivery Pipeline** section.

Download your app for local development:

1. Install the IBM Cloud CLI. [Learn more](#)
2. Run the `ibmcloud dev code <APPNAME>` command to download your app. If you are unsure of the app name or would like to download another app, you can list all the IBM Cloud apps that are in a space by running the `ibmcloud dev list` command.

[Learn more about deploying your app](#)

FEEDBACK ASK A QUESTION

- After the successful creation of cloudant service, click on deploy your app button

IBM Cloud Catalog Docs Support Manage Divya Nemuri's Account

Resource list / App details

Node RED BRZHK Add tags Resource group: Default Download code

App details Developer resources

Services (1) Connect existing services Create service

Name	Resources	Actions
Cloudant	Documentation	

Configure Continuous Delivery

Continuous Delivery is not enabled for this app. Enable Continuous Delivery to automate builds, tests, and deployments through Delivery Pipeline, GitLab, and more.

Deploy your app

Credentials Show

Knowledge Guide

Next steps

To enable Continuous Delivery and deploy to IBM Cloud:

1. Click **Deploy your app**.
2. Select a deployment target, select the toolchain settings, and click **Create**. The deployment starts automatically.
3. Check the deployment status in the **Delivery Pipeline** section.

Download your app for local development:

1. Install the IBM Cloud CLI. [Learn more](#)
2. Run the `ibmcloud dev code <APPNAME>` command to download your app. If you are unsure of the app name or would like to download another app, you can list all the IBM Cloud apps that are in a space by running the `ibmcloud dev list` command.

[Learn more about deploying your app](#)

FEEDBACK ASK A QUESTION

- In the process of deploying you will be prompted for the IBM cloud API key. Click on new to generate the one.

Resource list / App details

Node RED BRZHK

[Cancel](#) [Create](#)

Deploy your app

Select your deployment target and configure your DevOps toolchain. After you click **Create**, the toolchain is created, and the deployment process is started automatically.

Deployment target

IBM Cloud Foundry
Deploy and run your applications without managing servers or clusters. A Lite plan is available for quick and easy deployment.

IBM Cloud API key

IBM Cloud API key [New](#) [+](#)

The value is required.

Number of instances

1

Memory allocation per instance

64 MB 2000 MB 128

Step 1. Select the deployment target

Select your deployment target, and then provide the configuration information.

IBM Cloud Foundry

Cloud Foundry is the premier industry standard Platform-as-a-Service (PaaS) that ensures fast, easy, and reliable deployment of cloud-native apps. Cloud Foundry ensures that the build and deploy aspects of coding remain carefully coordinated with any attached services — resulting in quick, consistent and reliable iterating of applications. Cloud Foundry has a Lite plan that allows quick deployments for testing purposes.

Before you begin

- If your account doesn't have a Cloud Foundry org, you must create one. [Create org.](#)

Steps

1. Select the number of instances, memory allocation, region, org, and space.
2. Select the domain and provide a host name.

Resource list / App details

Node RED BRZHK

[Cancel](#) [Create](#)

Deploy your app

Select your deployment target and deployment process is started

Deployment target

IBM Cloud Foundry
Deploy and run your applications without managing servers or clusters. A Lite plan is available for quick and easy deployment.

IBM Cloud API key

IBM Cloud API key

The value is required.

Number of instances

1

Memory allocation per instance

64 MB 2000 MB 256

Create a new API key with full access

Warning: This will create a new API key that allows anyone who has it the ability to do anything you could do. You can improve your security posture by using the [IAM UI to create a service ID API key](#) that limits access to only what your pipeline requires, and then pasting that into the template UI instead.

For more information on API keys and access see the [IAM documentation](#).

Key will be called: API Key for NodeREDBRZHK

☐ Save this key in a secrets store for reuse

[Cancel](#) [OK](#)

Step 1. Select the deployment target

Select your deployment target, and then provide the configuration information.

IBM Cloud Foundry

Cloud Foundry is the premier industry standard Platform-as-a-Service (PaaS) that ensures fast, easy, and reliable deployment of cloud-native apps. Cloud Foundry ensures that the build and deploy aspects of coding remain carefully coordinated with any attached services — resulting in quick, consistent and reliable iterating of applications. Cloud Foundry has a Lite plan that allows quick deployments for testing purposes.

Before you begin

- If your account doesn't have a Cloud Foundry org, you must create one. [Create org.](#)

Steps

1. Select the number of instances, memory allocation, region, org, and space.
2. Select the domain and provide a host name.

Increase your memory allocation per instance to 256

Resource list / App details

Node RED BRZHK

Deploy your app

Select your deployment target and configure your DevOps toolchain. After you click **Create**, the toolchain is created, and the deployment process is started automatically.

Deployment target

IBM Cloud Foundry
Deploy and run your applications without managing servers or clusters. A Lite plan is available for quick and easy deployment.

IBM Cloud API key

Number of instances

Memory allocation per instance

64 MB ————— 2000 MB 256

Select region to deploy in: London

Select an organization: divya@thesmartbridge.com

Select a space: dev

Cancel Create

Step 1. Select the deployment target

Select your deployment target, and then provide the configuration information.

IBM Cloud Foundry

Cloud Foundry is the premier industry standard Platform-as-a-Service (PaaS) that ensures fast, easy, and reliable deployment of cloud-native apps. Cloud Foundry ensures that the build and deploy aspects of coding remain carefully coordinated with any attached services — resulting in quick, consistent and reliable iterating of applications. Cloud Foundry has a Lite plan that allows quick deployments for testing purposes.

Before you begin

- If your account doesn't have a Cloud Foundry org, you must create one. [Create org.](#)

Steps

- Select the number of instances, memory allocation, region, org, and space.
- Select the domain and provide a host name.

- Upon successful creation of the node-red app, you will be prompted with success message popup.

Resource list / App details

Node RED BRZHK

Add tags

App details Developer resources

Services (1) Connect existing services Create service

Name	Resources	Actions
Cloudant	Documentation	

Continuous Delivery Remove from toolchain

Toolchain

Name: NodeREDBRZHK

Location: Dallas

Tool integrations

Delivery Pipelines

Name: NodeREDBRZHK

Status: No stages detected

Success! Your DevOps toolchain has been created

3/19/2020, 2:21:10 PM

Knowledge Guide

Next steps

To enable Continuous Delivery and deploy to IBM Cloud:

- Click **Deploy your app**.
- Select a deployment target, select the toolchain settings, and click **Create**. The deployment starts automatically.
- Check the deployment status in the **Delivery Pipeline** section.

Download your app for local development:

- Install the IBM Cloud CLI. [Learn more](#)
- Run the `ibmcloud dev code <APPNAME>` command to download your app. If you are unsure of the app name or would like to download another app, you can list all the IBM Cloud apps that are in a space by running the `ibmcloud dev list` command.

[Learn more about deploying your app](#)

- **Task-2: Launch the node-red application and customization**

- To launch the node-red application click on cloud foundry apps in the dash board
- If it is stopped click on start to start the application. Click on the service to launch

Resource list

Create resource

Collapse all | Expand all

Name	Group	Location	Status	Tags
Filter by name or IP address...	Filter by group or org...	Filter...	Filter...	Filter...
Devices (0)				
VPC infrastructure (0)				
Clusters (0)				
Cloud Foundry apps (1)				
Node RED BRZHK	divya@thesmartbridge.com / dev	London	Stopped	
Cloud Foundry services (1)				
node-red-brzhk-cloudant-15846075...	divya@thesmartbridge.com / dev	London	Provisioned	
Services (3)				
Storage (0)				
Network (0)				
Cloud Foundry enterprise environments (0)				
Functions namespaces (0)				
Apps (1)				
Developer tools (1)				
VMware (0)				
Schematics workspaces (0)				

Start

Edit name

Add tags

Delete

FEEDBACK

- Click on visit app URL to launch the app

Resource list /

Node RED BRZHK This app is awake. Visit App URL

Org: divya@thesmartbridge.com Location: London Space: dev Add tags

Runtime

BUILDPACK SDK for Node.js™

INSTANCES All instances are running Health is 100%

MB MEMORY PER INSTANCE 256

TOTAL MB ALLOCATION 256 0 MB still available

Connections (1)

node-red-brzhk-cloudant-1584607563375-71694

Create connection

Runtime cost

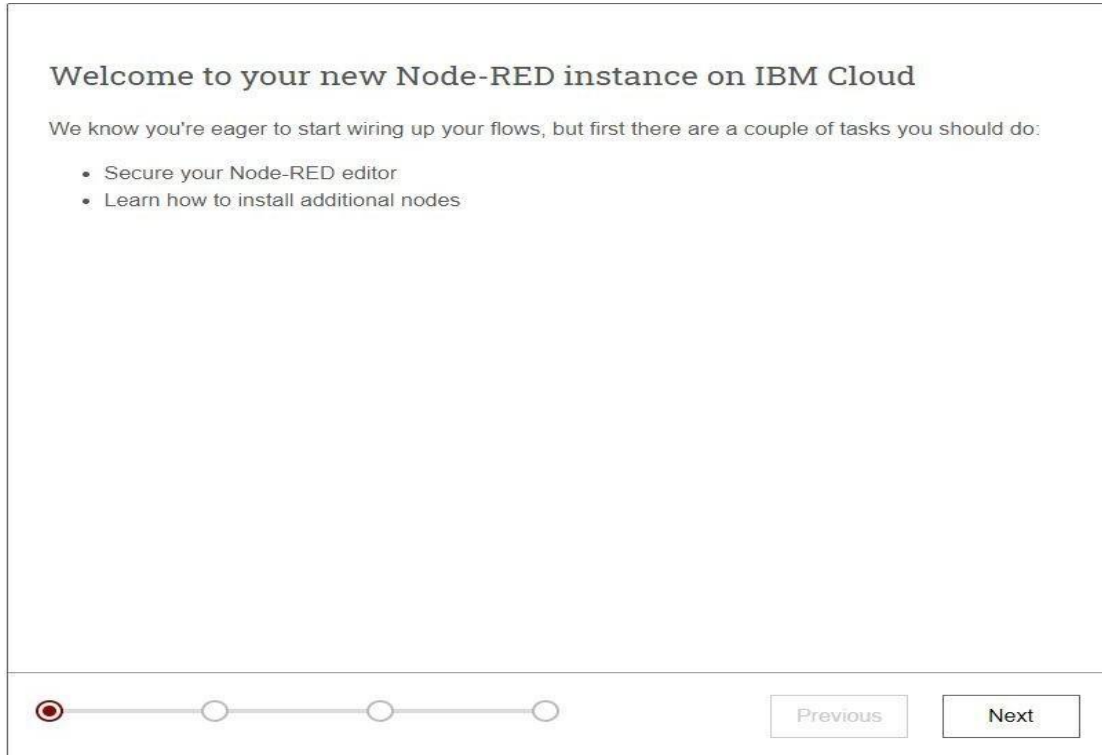
\$0.00 Current charges for billing period

\$0.00 Estimated total for billing period (Mar 1, 2020 - Mar 31, 2020)

Current and estimated cost excludes connected services.

FEEDBACK

- Upon launching the app you are redirected to the following page. Click on next.



- In the next step, you are asked for securing the node-red app, if you want you can provide the credentials as you wish but remember forgetting those credentials lead to deleting the app and have to be started from the beginning. So better avoid giving credentials.
- Click the two checkboxes as shown in the below image. This step is for not securing the node-red editor.

Secure your Node-RED editor

☒ Secure your editor so only authorised users can access it.

Username

Password Must be at least 8 characters

☐ Allow anyone to view the editor, but not make any changes.

☐ *Not recommended:* Allow anyone to access the editor and make changes

- Then after selecting the not recommended you are asked for confirmation proceed with that and click on next.

Secure your Node-RED editor

☐ Secure your editor so only authorised users can access it

☒ *Not recommended:* Allow anyone to access the editor and make changes
 Your editor will not be secured. Anyone with the URL will be able to access your flows, data and bound services.

¹ ☐ Tick this box to confirm you want your editor to be insecure

²

- This procedure is done only for the first time launch.

- For a returning user, you can directly browse from the cloud foundry apps in the dashboard. And click on the visit app URL after getting the app is awake/running.

Learn how to install additional nodes

Node-RED provides a **huge catalog of extra nodes** you can install into the editor.

Many of these nodes can be installed directly from the editor's palette manager feature. However that can cause issues due to the limited memory of the default Node-RED starter application.

The *recommended approach* is to edit your application's `package.json` file to include the additional node modules and then redeploy the application. This can be done using the Continuous Delivery feature on the application's IBM Cloud dashboard.

For more information, follow [this tutorial on IBM Developer](#).

Previous

Next

Finish the install

You have made the following selections:

- *Not recommended:* Allow anyone to access the editor and make changes

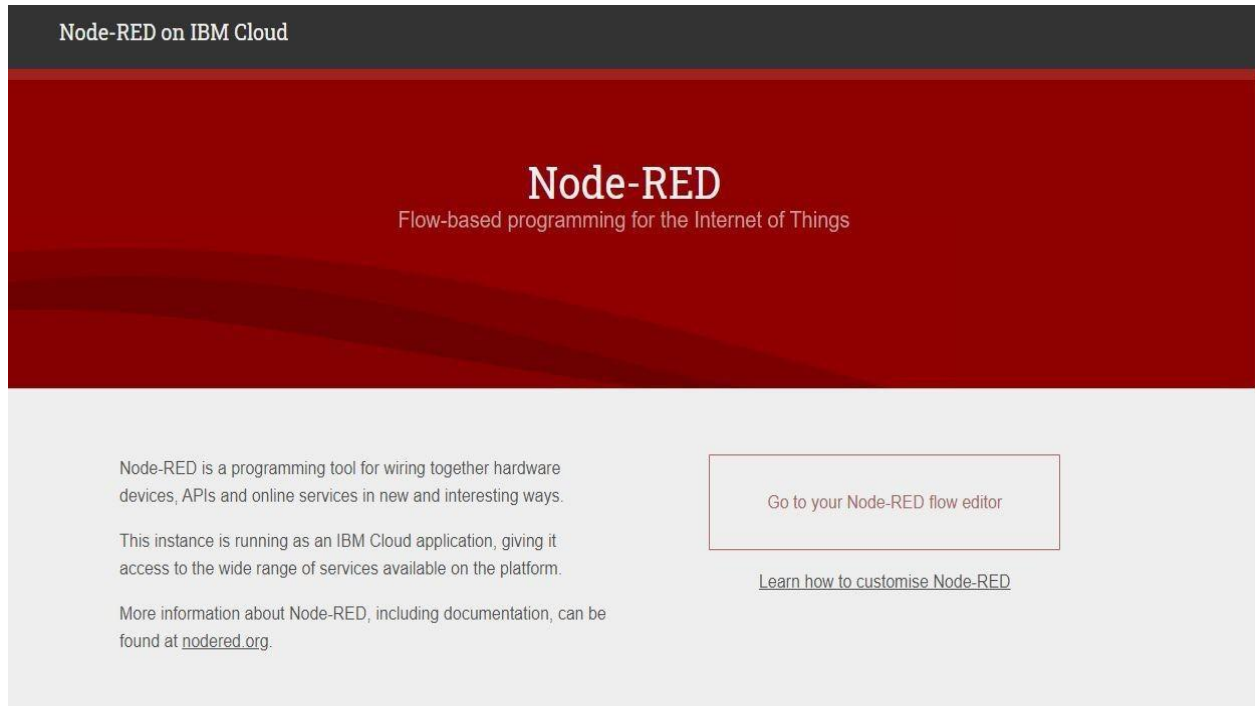
You can change these settings at any time by setting the following environment variables via the IBM Cloud console:

- `NODE_RED_USERNAME` - the username
- `NODE_RED_PASSWORD` - the password
- `NODE_RED_GUEST_ACCESS` - if set to `"true"`, allows anyone read-only access to the editor

Previous

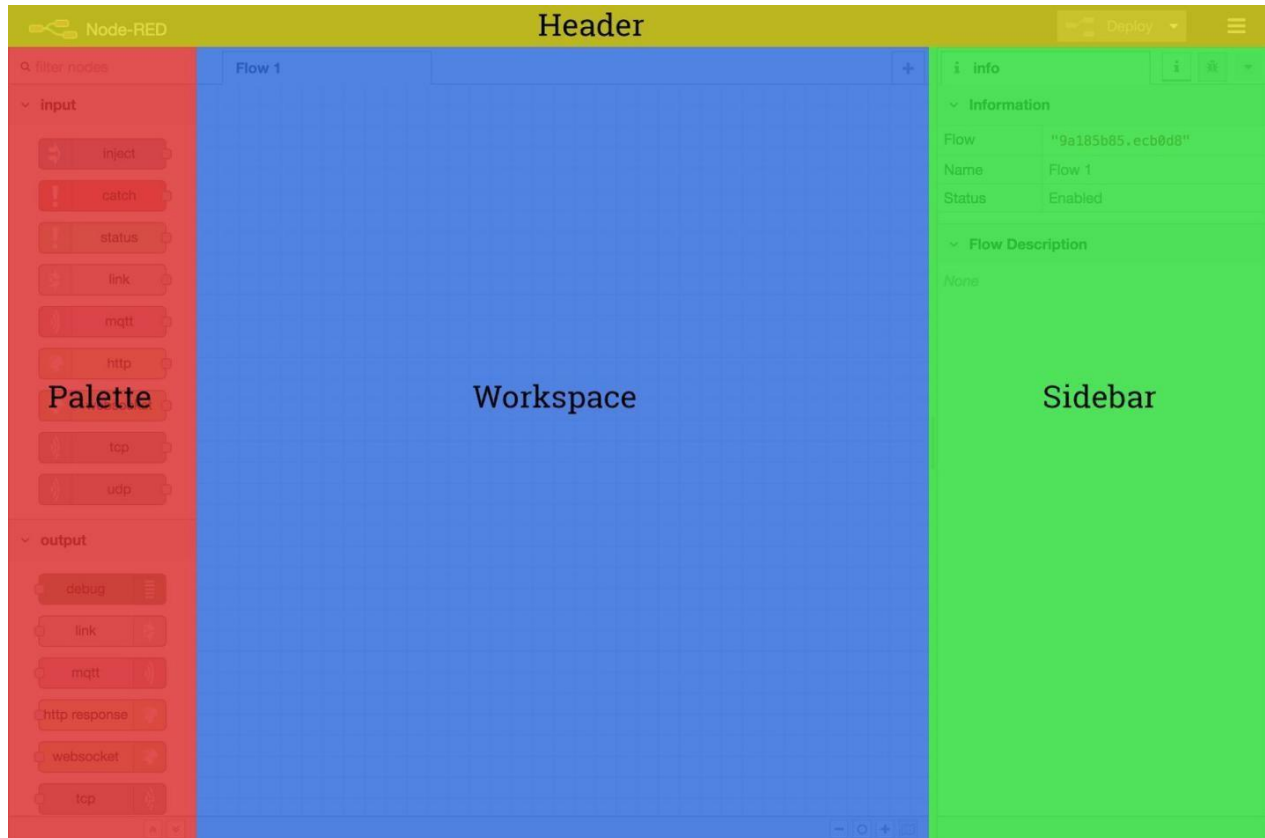
Finish

- After completing the procedure of launching node-red you are asked to go to the editor. Click on go to the editor button to launch the application.

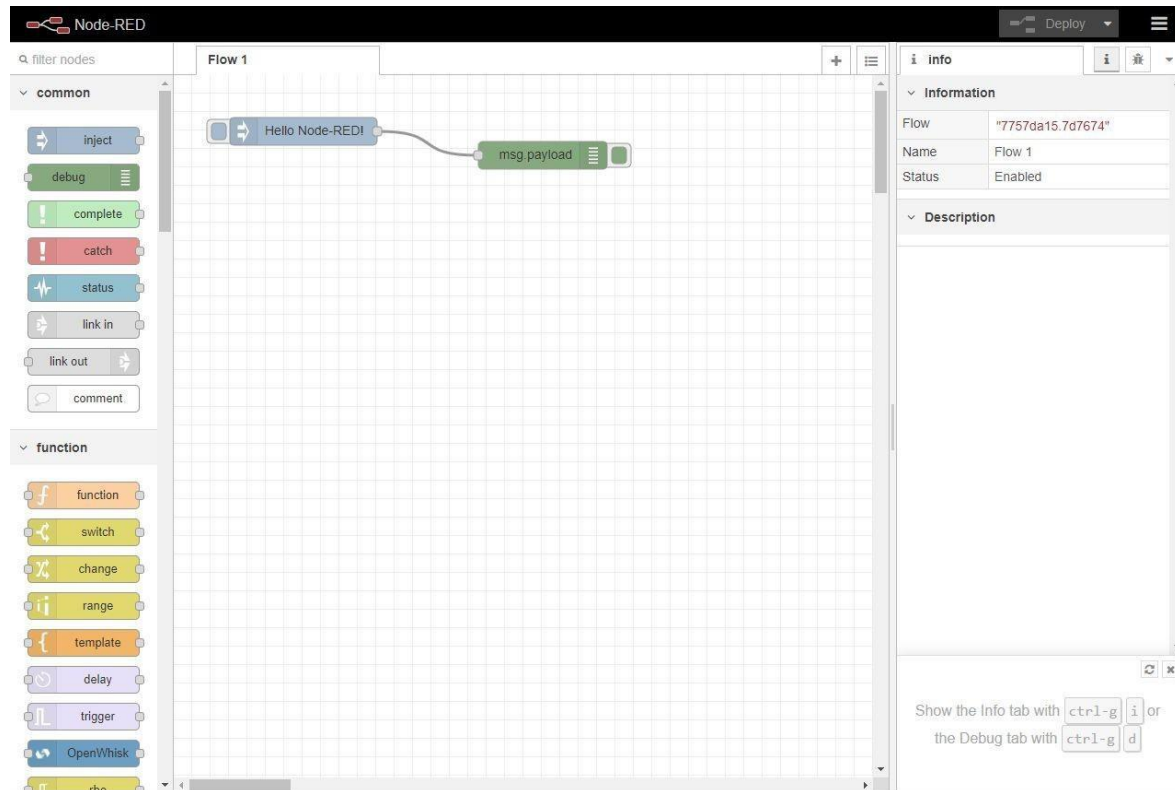


- **Task-3: Basic Node-red flow**

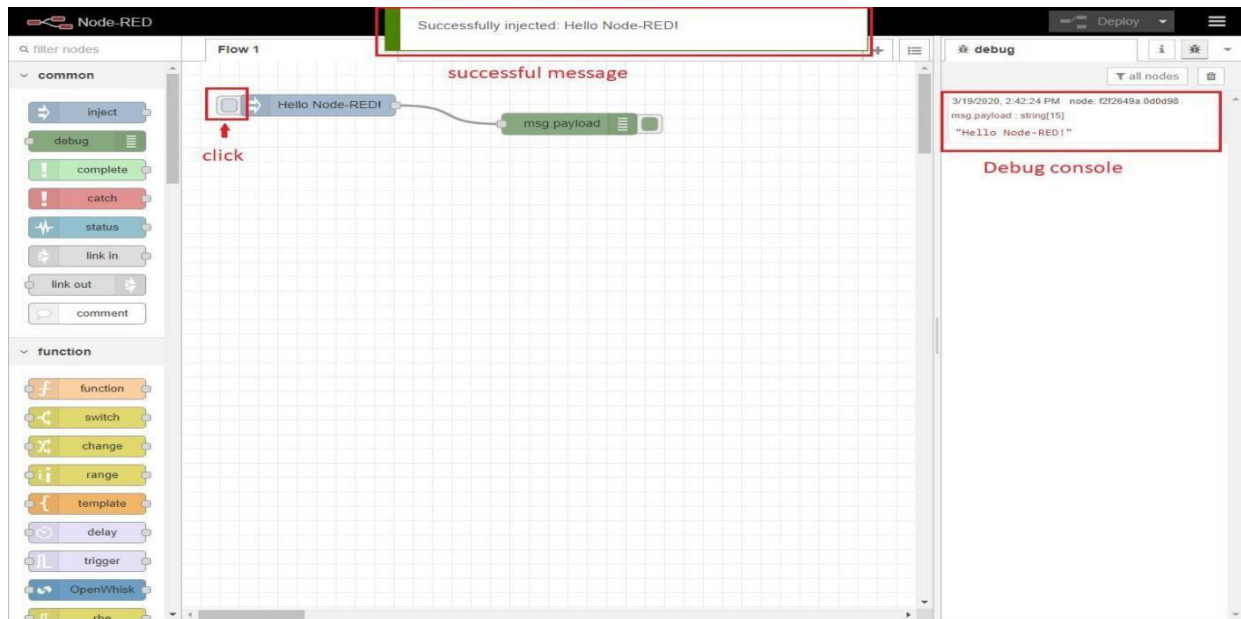
- On launching your node-red application you will be on the editor pane where one can drag and drop the nodes.



- From the palette, one can drag the nodes and drop them on the editor.
- The sidebar will be useful to know the info about nodes by selecting them and the debug console helps us to debug the errors or the output is displayed on the console.
- The default node-red after launching the node-red application for a new user will contain two nodes connected to each other.
- The first node is the input used to inject the input for output.
- The debug node to display the output

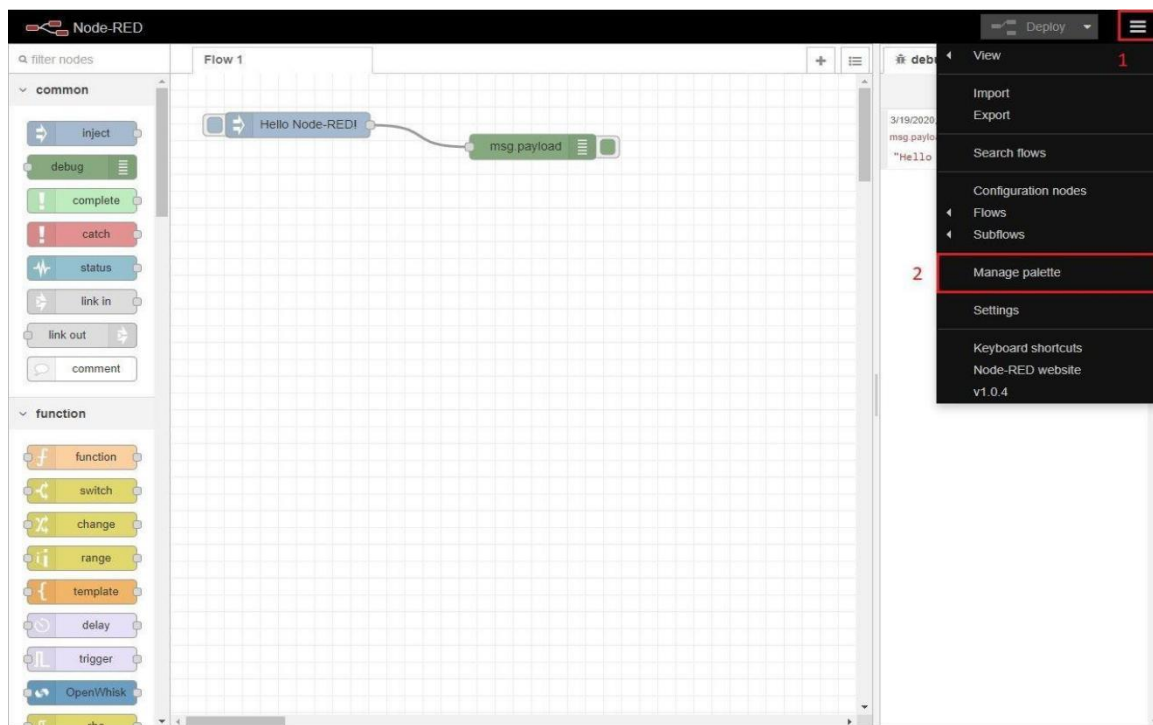


- Click on the inject node to inject the input to display on output.

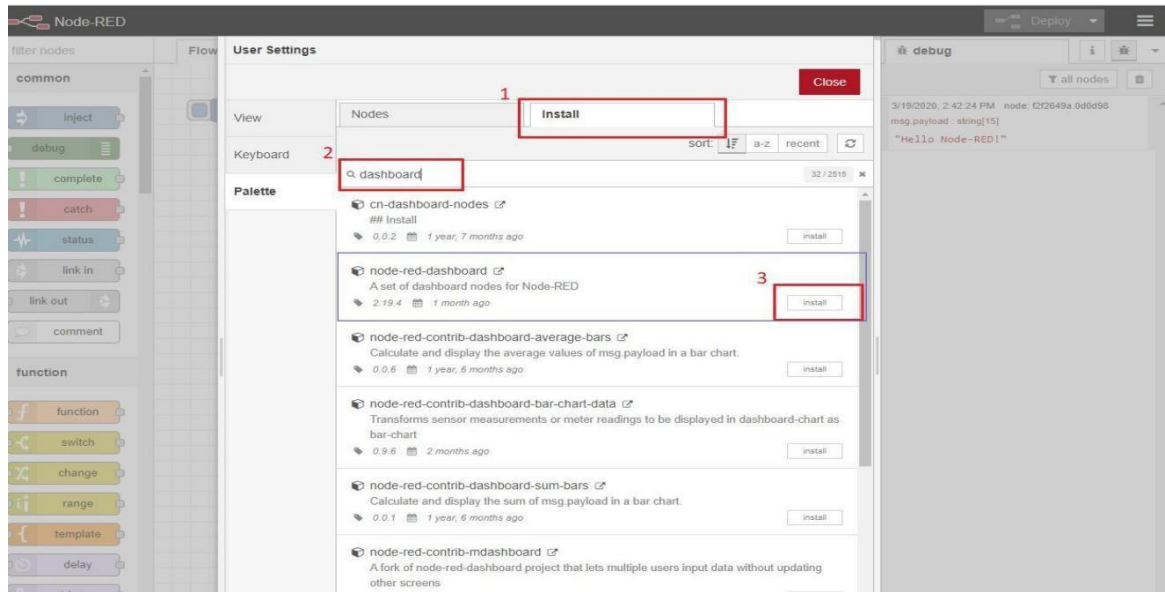


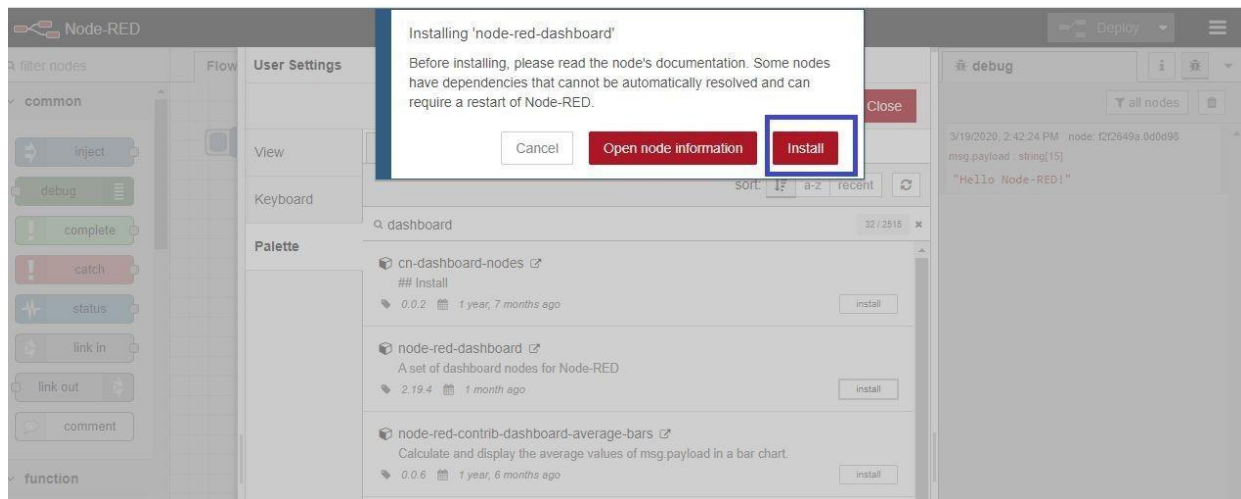
• Task-4:Dashboardnodesinstallation

- Navigatetomanagepaletteforinstallingthenodes required.

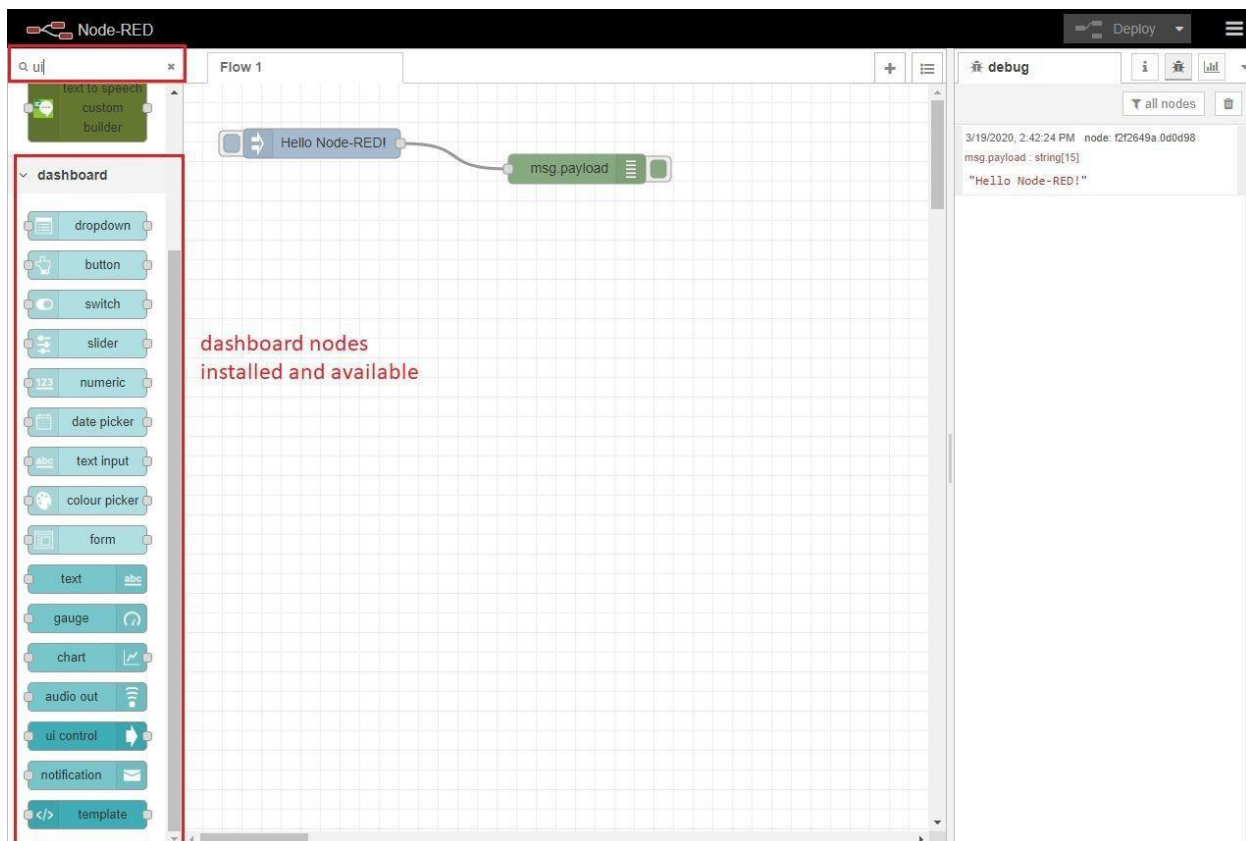


- Searchfor node-red-dashboard for installing the UI nodes used to create webApp





- Click on **Install** in the confirmation popup to install the UI nodes
- Upon successful installation you are prompted with the list of nodes installed. To view the nodes installed search for **UI** in the palette



- **Text to Speech:**

The Text to Speech service converts written text to natural-sounding speech. The service streams the synthesized audio back with minimal delay. The audio uses appropriate cadence and intonation for its language and dialect to provide voices that are smooth and natural..

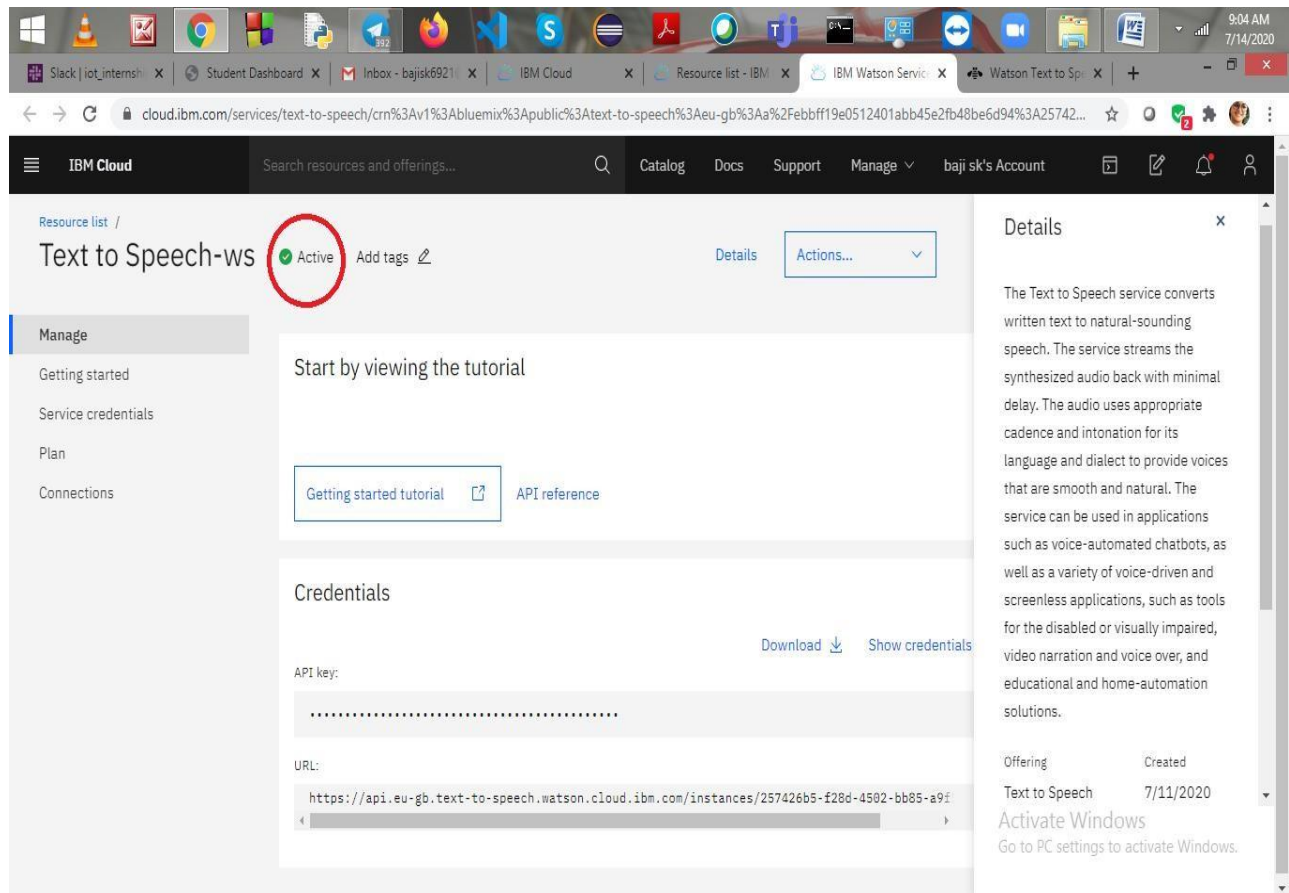


Fig. Text to speech

Node-Red Cloudant

IBM Cloudant is a fully managed JSON document database that offers independent serverless scaling of provisioned throughput capacity and storage. Cloudant is compatible with Apache CouchDB and accessible through a simple to use HTTPS API for web, mobile, and IoT applications. Cloudant is SOC2 and ISO27001 compliant with HIPAA readiness optional for Dedicated Hardware environments. Cloudant Standard plan instances come with a 99.99% SLA. All data is encrypted at rest and over the wire. Cloudant JSON documents are stored in triplicate across three separate availability zones for in-region HA/DR in regions that support AZ's. Any Cloudant instance deployed from the Frankfurt region/location will be in an EU-managed environment. See <https://ibm.com/cloud/cloudant> or the documentation link below for more details. Use the new Standard on Transaction Engine plan which offers cheaper storage and query costs, strongly consistent reads, scalable and consistent global secondary indexes, elimination of in-region document conflicts, and additional in-database encryption of data values.

The screenshot displays the IBM Cloudant console interface. The resource name is 'node-red-yiefz-cloudant-1594426328184', which is in an 'Active' state. The 'Capacity' tab is selected, showing the following details:

Capacity details	
You are currently subscribed to the Lite plan.	
Reads	20 per second
Writes	10 per second
Global queries	5 per second
Storage	0.00 GB of 1GB

An 'Upgrade' button is visible next to the plan information. The left sidebar shows options for 'Manage', 'Service credentials', 'Plan', and 'Connections'. The top navigation bar includes 'Catalog', 'Docs', 'Support', and 'Manage'. The bottom of the screen shows an 'Announcements' section and a Windows activation notice.

Fig The current plan that we are using in cloud.

Resource list / node-red-yiefz-cloudant-1594426328184 Active [Add tags](#)

Manage Overview Dashboard **Capacity** Docs [Launch Dashboard](#) [Upgrade Plan](#)

You are currently subscribed to the Lite plan.

Current Capacity		Target Capacity	
20	Reads per second	100	Reads per second
10	Writes per second	50	Writes per second
5	Global queries per second	5	Global queries per second
\$0.000 /hour \$0.00 /month		\$0.105 /hour \$76.65 /month	

[Upgrade to Change Capacity](#)

Fig .Capacity of cloud

Databases [Database name](#) [Create Database](#) [JSON](#) [Bell](#)

Your Databases

Name	Size	# of Docs	Partitioned	Actions
doorbel	12.0 KB	7	No	Refresh Lock Delete
noderedyiefz	70.3 KB	4	No	Refresh Lock Delete

Showing 1–2 of 2 databases. Databases per page 20 1

Fig. By clicking that button we can create database

We are using doorbell database

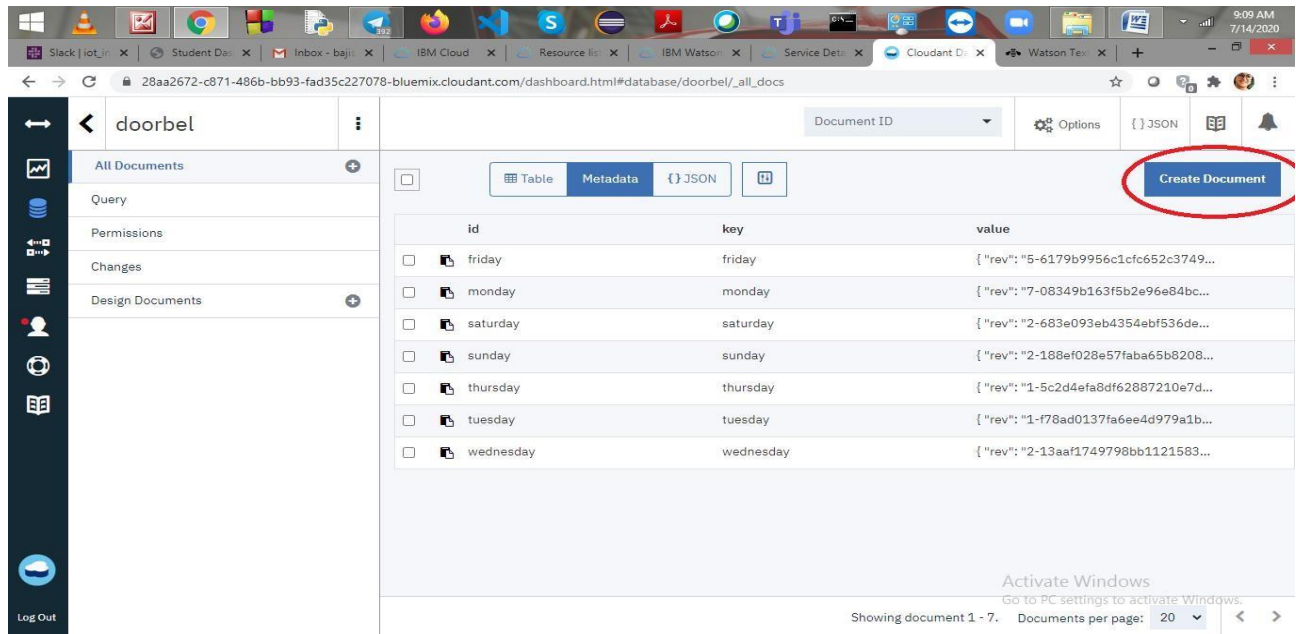


Fig. These are documents in ourdoorbelldatabase

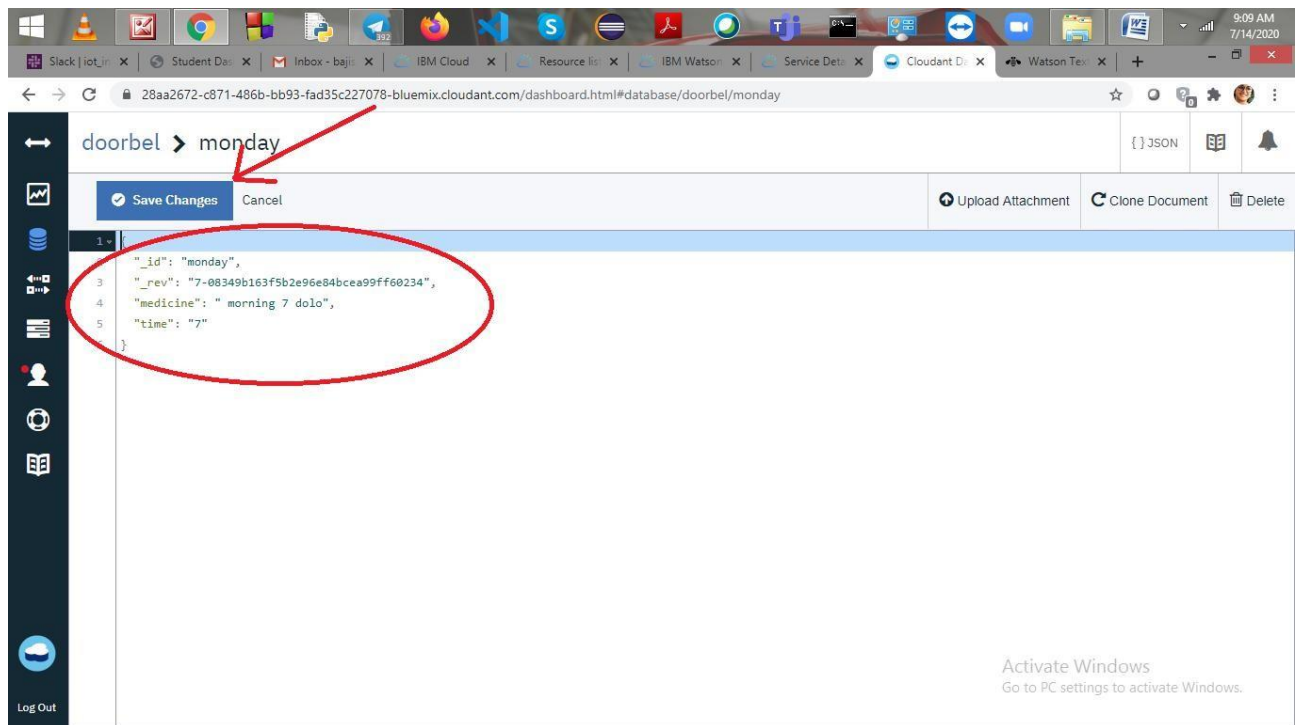


Fig. Document sample code

4.10

MIT INVENTOR

4.10.1

Introduction:

MIT means Massachusetts Institute of Technology(MIT) App Inventor is a cloud-based tool, by which you can build android applications using web browser. This website offers all the support you'll need to learn how to build your own apps.

4.10.2 Designer Panel:

Now, you are in the designer panel of your project, where you can layout the “user interfaces” of your app. You choose things for the user interface things like Buttons, Images, Textboxes to the palette.

Click and hold on the word “Button” in the palette, drag your mouse over to the Viewer. Drop the button and a new button will appear on the Viewer. You can change the properties of the components like height, width, text, Color to the properties panel.

Block Editor:

The Blocks Editor is where you program the behaviour of your app. There are Built-in blocks that handle things like math, logic, and text with each component have added.

Click on the Button1 drawer. Click and hold the “when Button1.Click do” block. Drag it over to the workspace and drop it there. This is the block that will handle what happens when the button on your application is clicked. It is called an “Event Handler”.

4.10.3

Designer Panel:

4.2

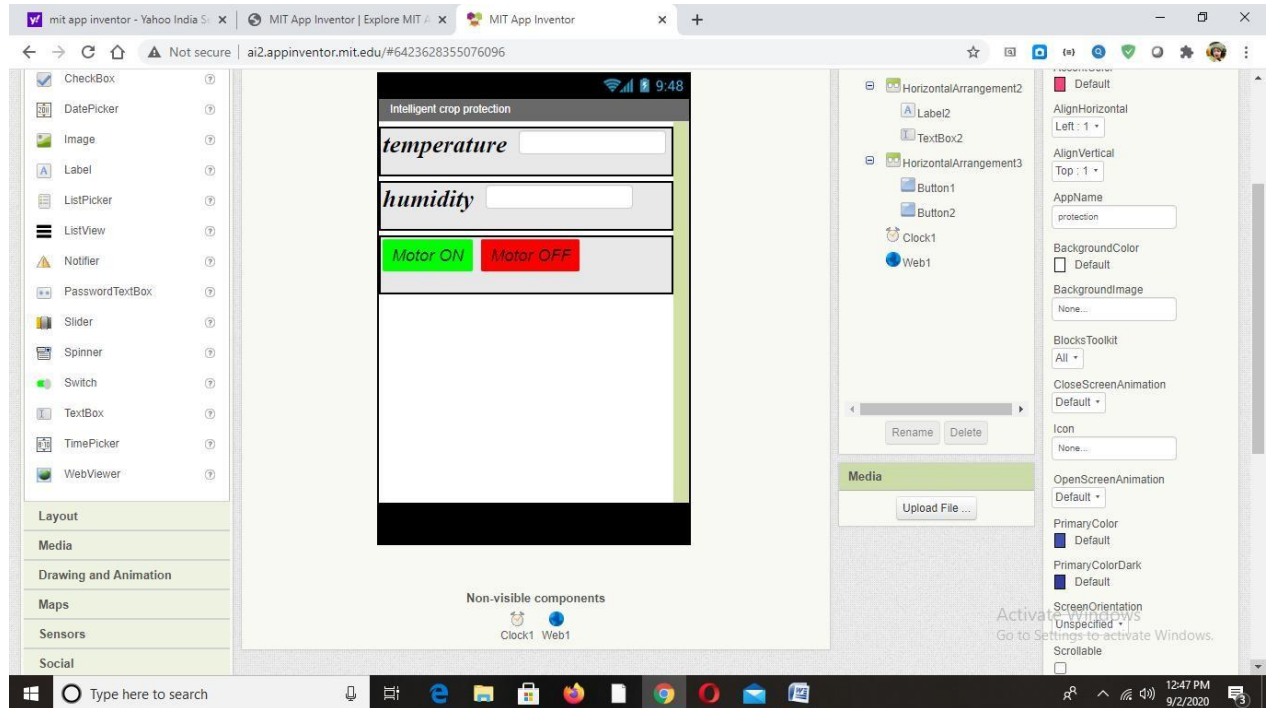


Fig:MITAppInventor

4.10.4

Live Testing:

- You can test your app while you are building it. For this, you need MIT AI2 Companion to connect your Android phone or tablet.
- The method for getting the AI2 Companion App is to download the app from the Play Store by searching for “MIT AI2 Companion” and installing it on your phone.

4.3

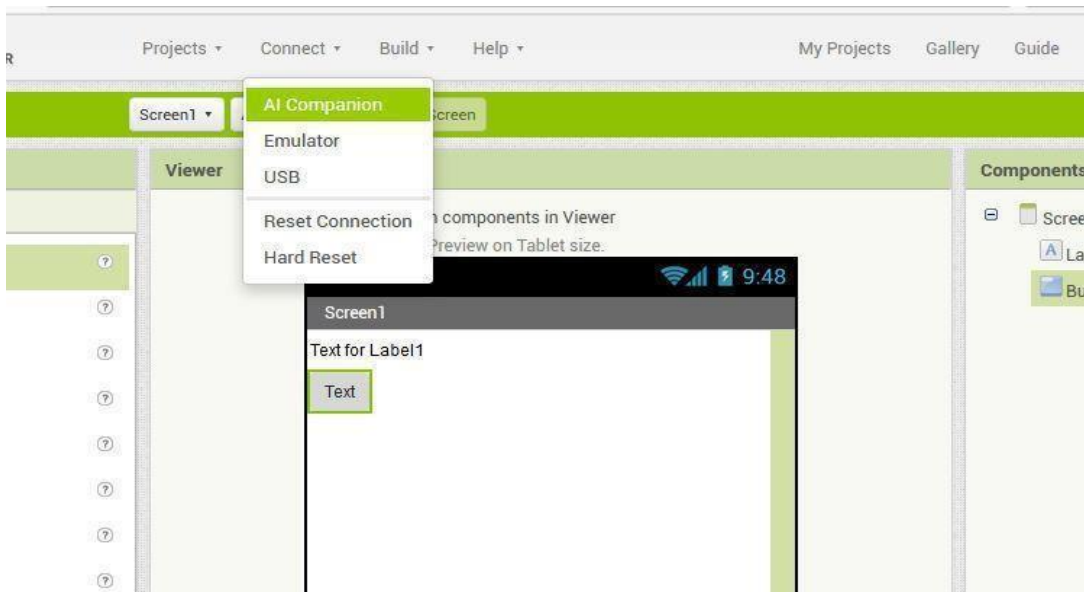


Fig: Live testing

After installation open the app and scan QR code by clicking “Scan QR code” or you can type the code after that click on “connect with code” button.

- After Connecting, you can see the app on your phone.

4.4

Mobile application:

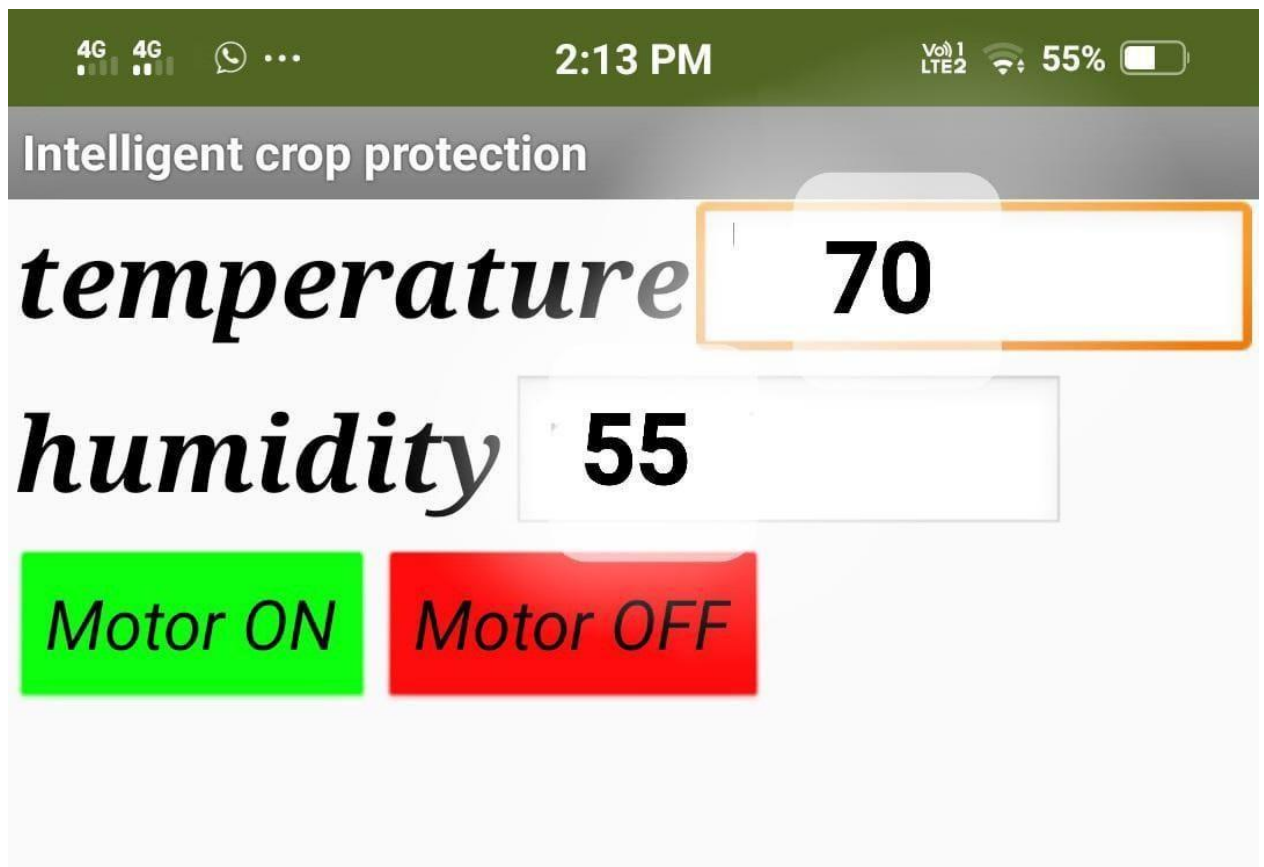


Fig: Mobile application

CHAPTER 5

RESULT AND FUTURE WORKS

5.1 RESULTS

Python is a general-purpose of interpreted, interactive, object-oriented, and high-level programming language. As an interpreted language, Python has a design philosophy that emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly brackets or keywords), and a syntax that allows programmers to express concepts in fewer lines of code than might be used in languages such as C++ or Java. It provides constructs that enable clear programming on both small and large scales. Python interpreters are available for many operating systems. Python, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. CPython is managed by the non-profit Python Software Foundation. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional, and procedural, and has a large and comprehensive standard library.

5.2 FUTURE WORKS

- The smart agricultural needs availability of internet continuously.
- Rural part of most of the developing countries do not fulfill this requirement. Moreover, internet connection is slower.
- The smart farming-based equipments require farming to understand and learn the use of technology.

This is a major challenge in adopting smart agricultural farming at large scale across the countries.

5.1

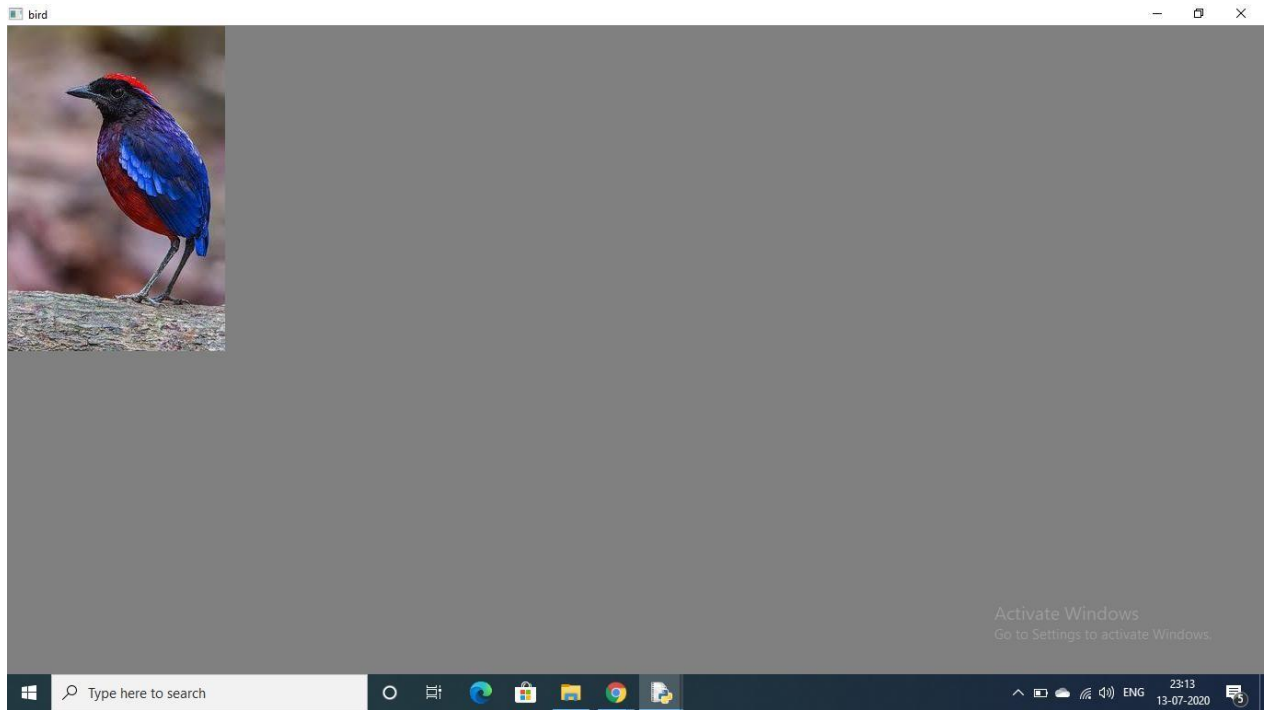


Fig: Image Capture

```
Python 3.7.7 Shell
File Edit Shell Debug Options Window Help
Python 3.7.7 (tags/v3.7.7:d7c567b08f, Mar 10 2020, 10:41:24) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\91812\Desktop\createbucket.py =====
Creating new bucket: prasuna9
CLIENT ERROR: An error occurred (BucketAlreadyExists) when calling the CreateBucket operation: The requested bucket name is not available. The bucket namespace is shared by all users of the system. Please select a different name and try again.
>>> |
```

Activate Windows
Go to Settings to activate Windows.

Ln: 8 Col: 4

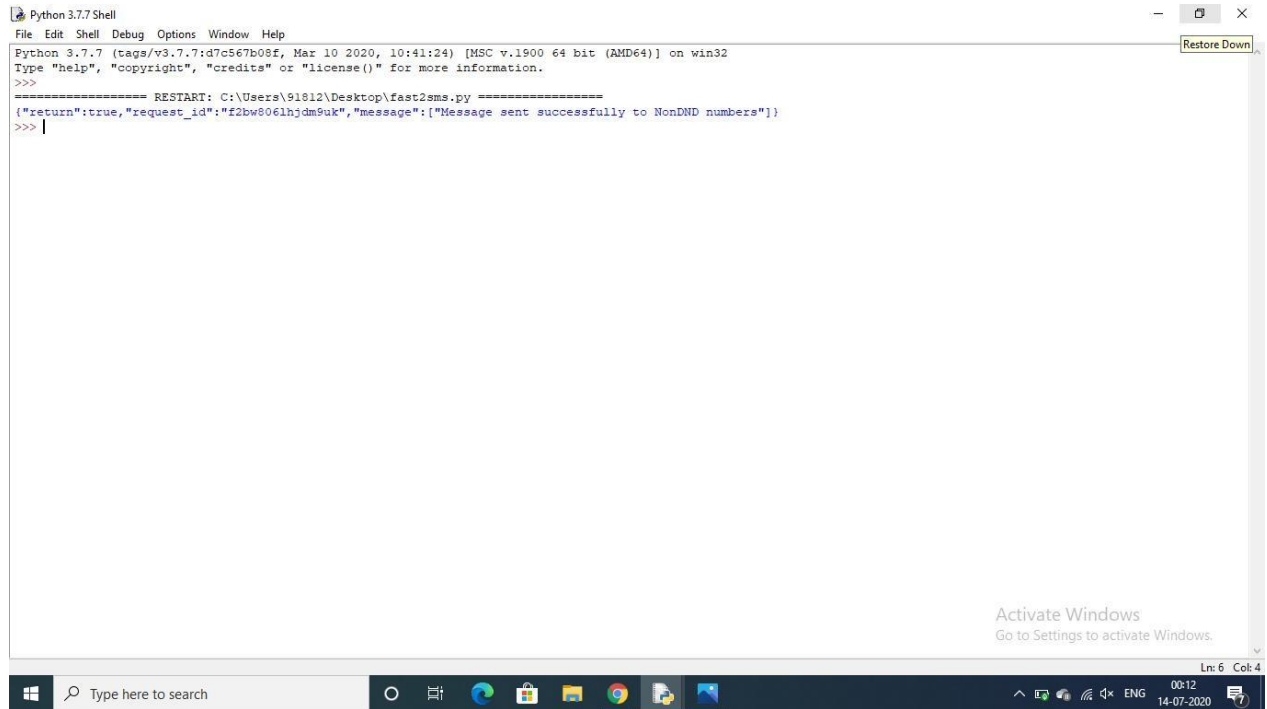
Fig: Create Bucket

```
Python 3.7.7 Shell
File Edit Shell Debug Options Window Help
Python 3.7.7 (tags/v3.7.7:d7c567b08f, Mar 10 2020, 10:41:24) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\91812\Desktop\multipartupload.py =====
Starting file transfer for crop.jpg to bucket: prasuna9
Transfer for crop.jpg Complete!
>>> |
```

Activate Windows
Go to Settings to activate Windows.

Ln: 9 Col: 4

Fig: Image Transfer To Bucket



```
Python 3.7.7 Shell
File Edit Shell Debug Options Window Help
Python 3.7.7 (tags/v3.7.7:d7c567b08f, Mar 10 2020, 10:41:24) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\91812\Desktop\fast2sms.py =====
{"return":true,"request_id":"f2bw806lhjdm9uk","message":["Message sent successfully to NonDND numbers"]}
>>>
```

Activate Windows
Go to Settings to activate Windows.

Ln: 6 Col: 4

Fig:Fast2SMS



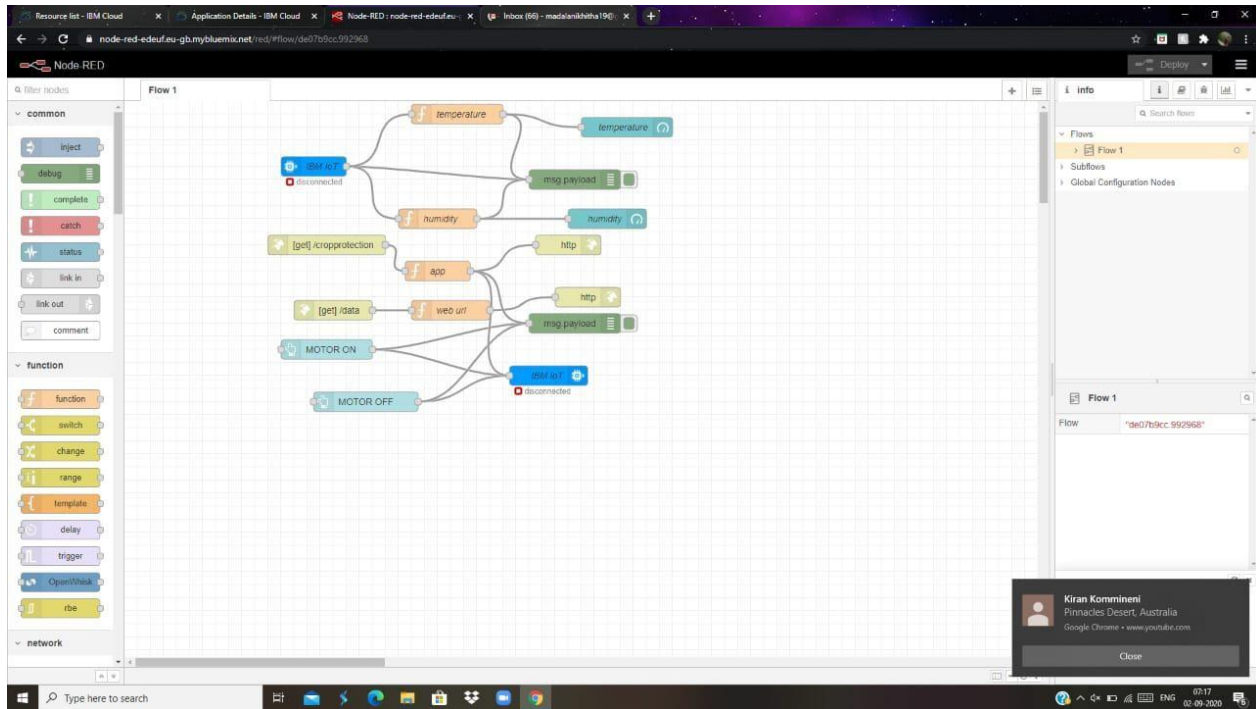
Fig:Fast2 SMS

```
*Python 3.7.7 Shell
File Edit Shell Debug Options Window Help
Published Temperature = 57 C Humidity = 21 % to IBM Watson
Published Temperature = 50 C Humidity = 21 % to IBM Watson
Published Temperature = 62 C Humidity = 27 % to IBM Watson
Published Temperature = 66 C Humidity = 23 % to IBM Watson
Published Temperature = 41 C Humidity = 29 % to IBM Watson
Published Temperature = 78 C Humidity = 30 % to IBM Watson
Published Temperature = 59 C Humidity = 24 % to IBM Watson
Published Temperature = 38 C Humidity = 21 % to IBM Watson
Published Temperature = 67 C Humidity = 26 % to IBM Watson
Published Temperature = 44 C Humidity = 21 % to IBM Watson
Published Temperature = 75 C Humidity = 12 % to IBM Watson
Published Temperature = 33 C Humidity = 21 % to IBM Watson
Published Temperature = 52 C Humidity = 37 % to IBM Watson
Published Temperature = 31 C Humidity = 39 % to IBM Watson
Published Temperature = 47 C Humidity = 13 % to IBM Watson
Published Temperature = 78 C Humidity = 23 % to IBM Watson
Published Temperature = 59 C Humidity = 30 % to IBM Watson
Published Temperature = 72 C Humidity = 36 % to IBM Watson
Published Temperature = 34 C Humidity = 14 % to IBM Watson
Published Temperature = 33 C Humidity = 19 % to IBM Watson
Published Temperature = 34 C Humidity = 27 % to IBM Watson
Published Temperature = 73 C Humidity = 16 % to IBM Watson
Published Temperature = 46 C Humidity = 15 % to IBM Watson
Published Temperature = 74 C Humidity = 20 % to IBM Watson
Published Temperature = 52 C Humidity = 33 % to IBM Watson
Published Temperature = 80 C Humidity = 34 % to IBM Watson
Published Temperature = 56 C Humidity = 25 % to IBM Watson
Published Temperature = 56 C Humidity = 34 % to IBM Watson
Published Temperature = 37 C Humidity = 37 % to IBM Watson
Published Temperature = 32 C Humidity = 34 % to IBM Watson
Published Temperature = 37 C Humidity = 12 % to IBM Watson
Published Temperature = 65 C Humidity = 40 % to IBM Watson
Published Temperature = 30 C Humidity = 21 % to IBM Watson
Published Temperature = 58 C Humidity = 30 % to IBM Watson
Published Temperature = 41 C Humidity = 14 % to IBM Watson
Published Temperature = 76 C Humidity = 12 % to IBM Watson
Published Temperature = 69 C Humidity = 10 % to IBM Watson
Published Temperature = 76 C Humidity = 14 % to IBM Watson
Published Temperature = 71 C Humidity = 12 % to IBM Watson
Published Temperature = 50 C Humidity = 15 % to IBM Watson
Published Temperature = 66 C Humidity = 29 % to IBM Watson

Activate Windows
Go to Settings to activate Windows.

Ln: 5 Col: 0
```

Fig: Temperature and Humidity Values



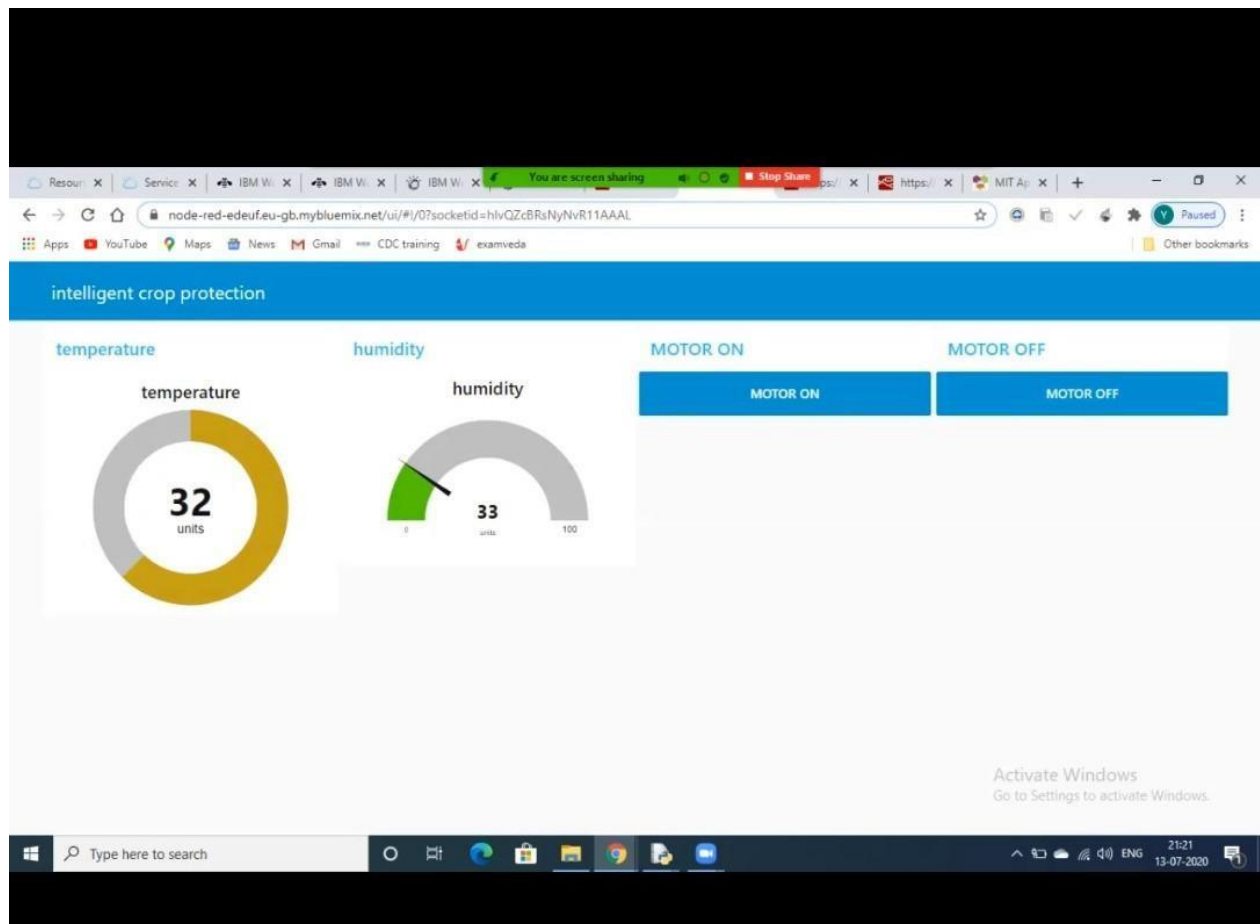


Fig: Temperature/Humidity Level

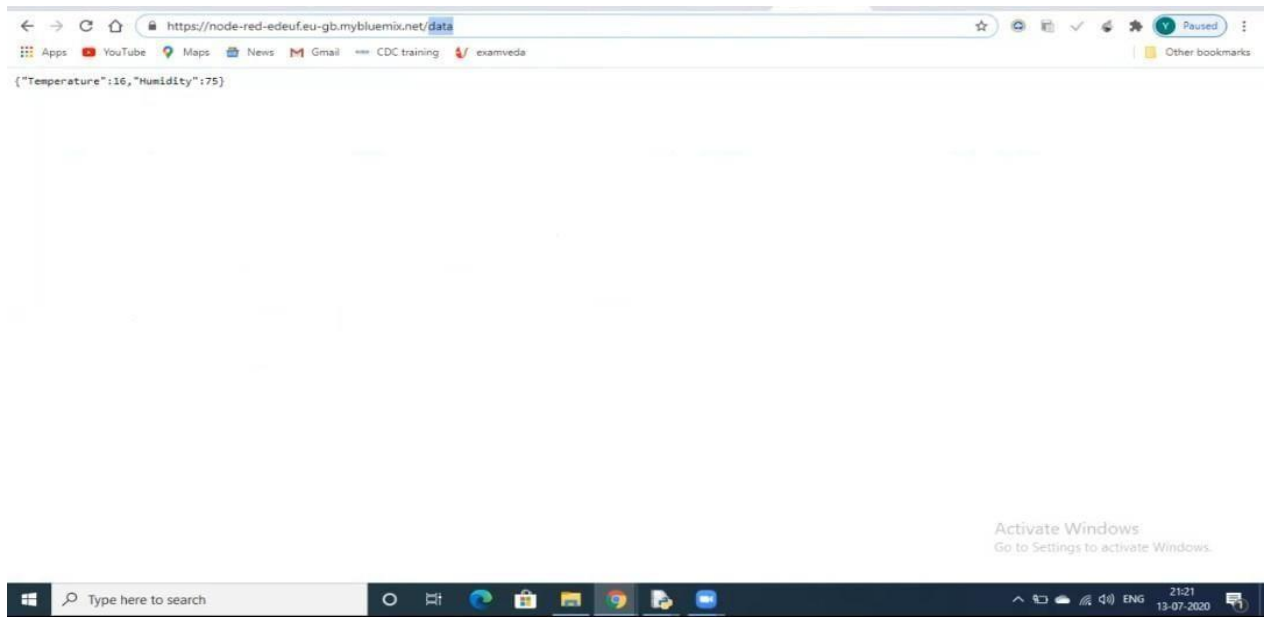


Fig:Temperature/HumidityValues

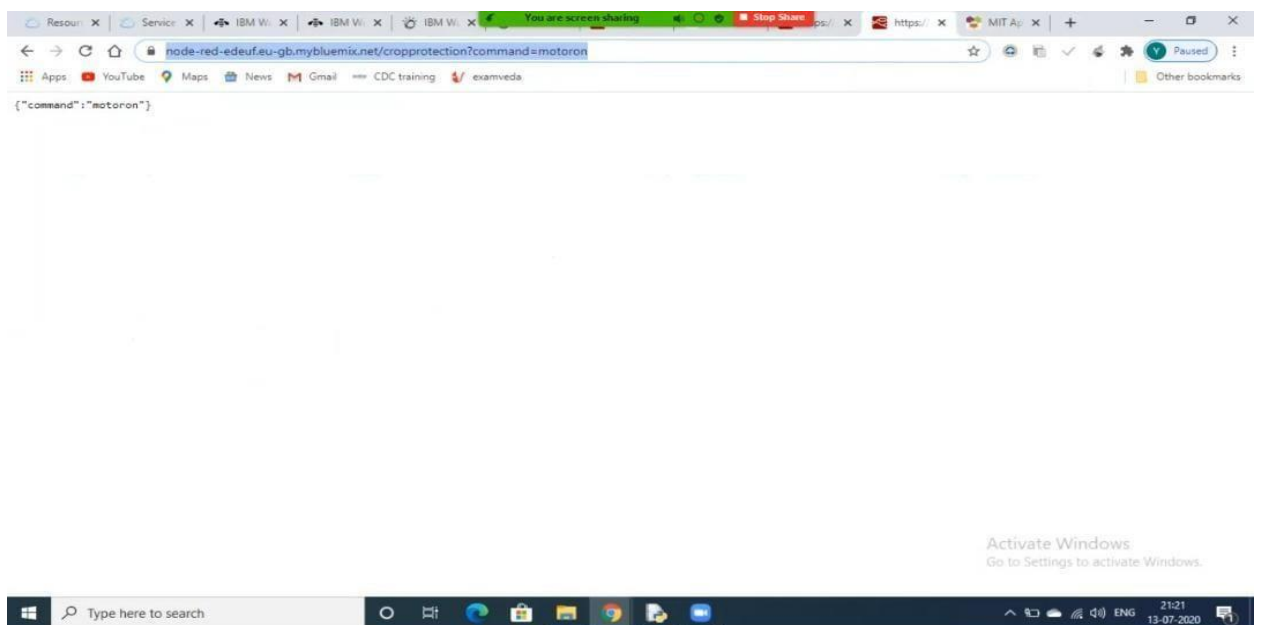


Fig: Motor Detail

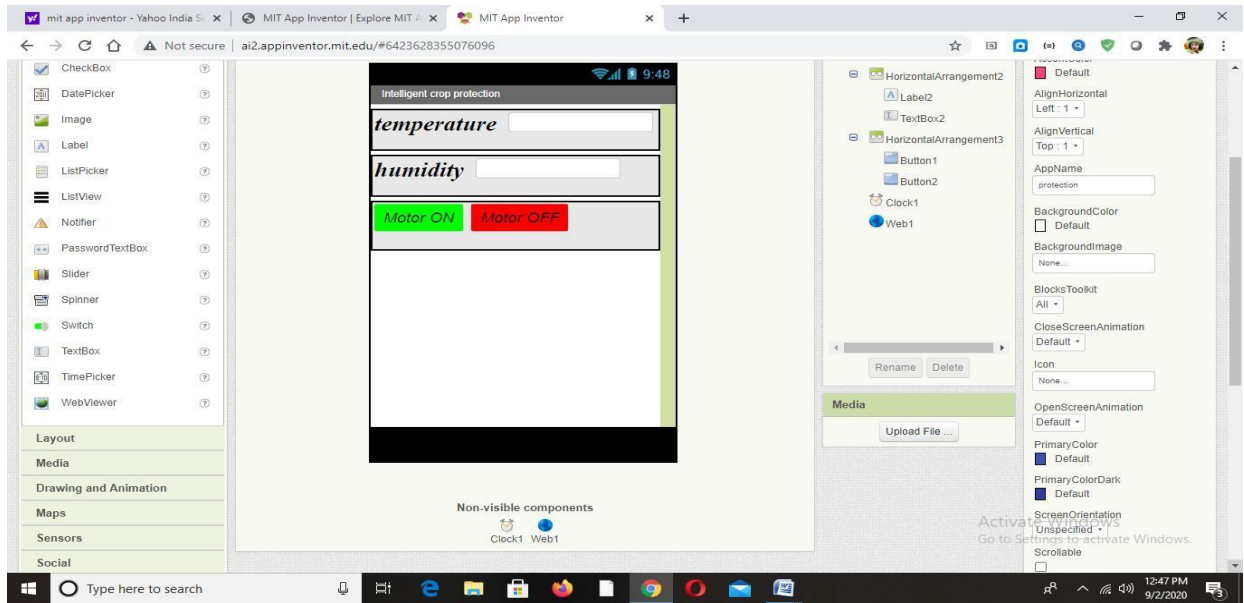


Fig: MIT App Inventor

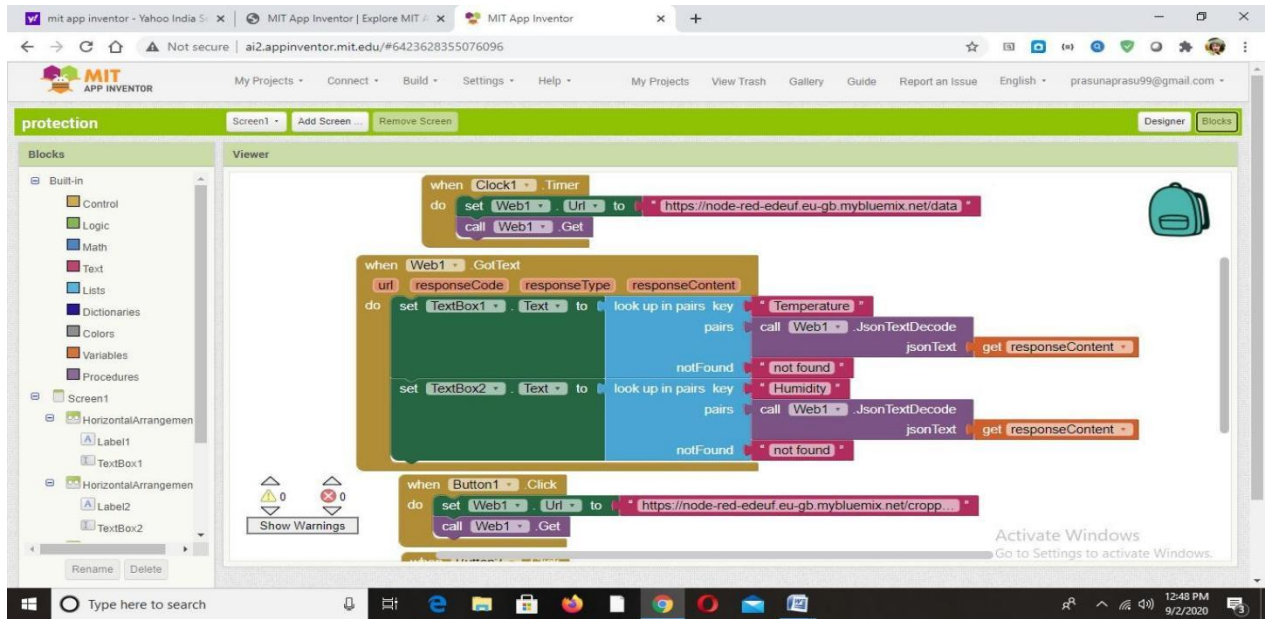


Fig: Block Designing

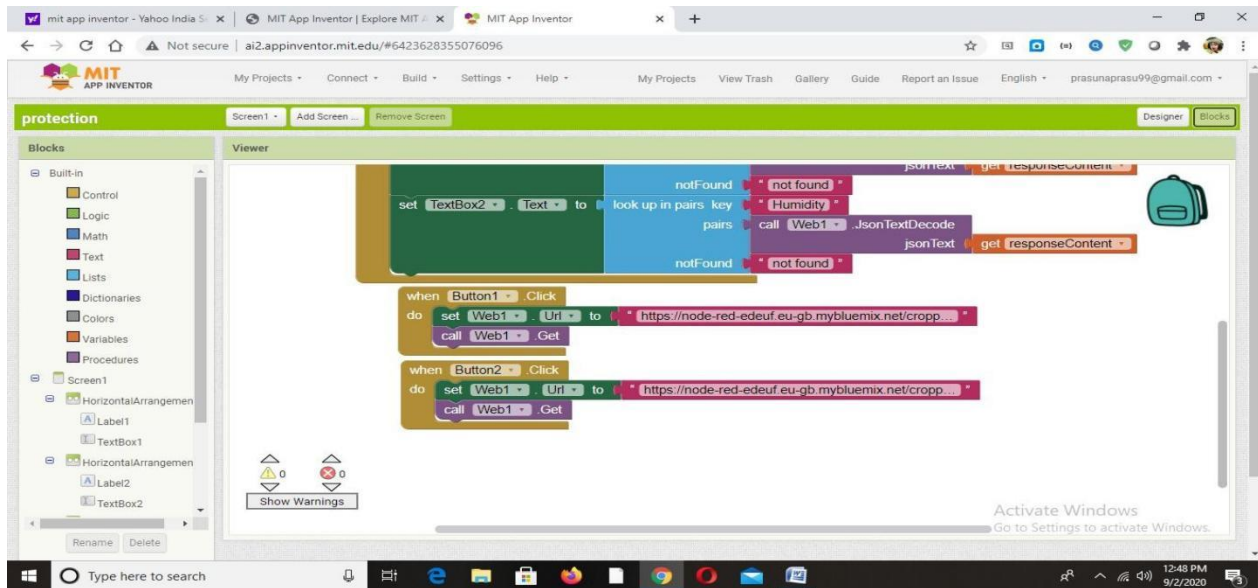


Fig: Block Designing Continue

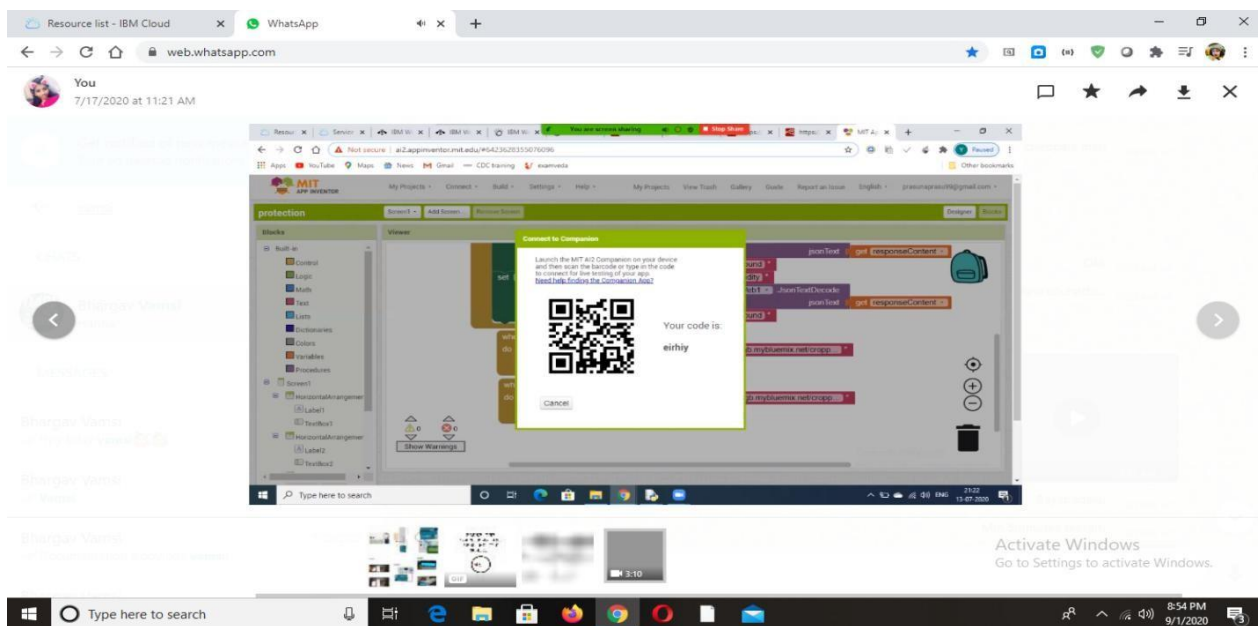


Fig: Scan URL

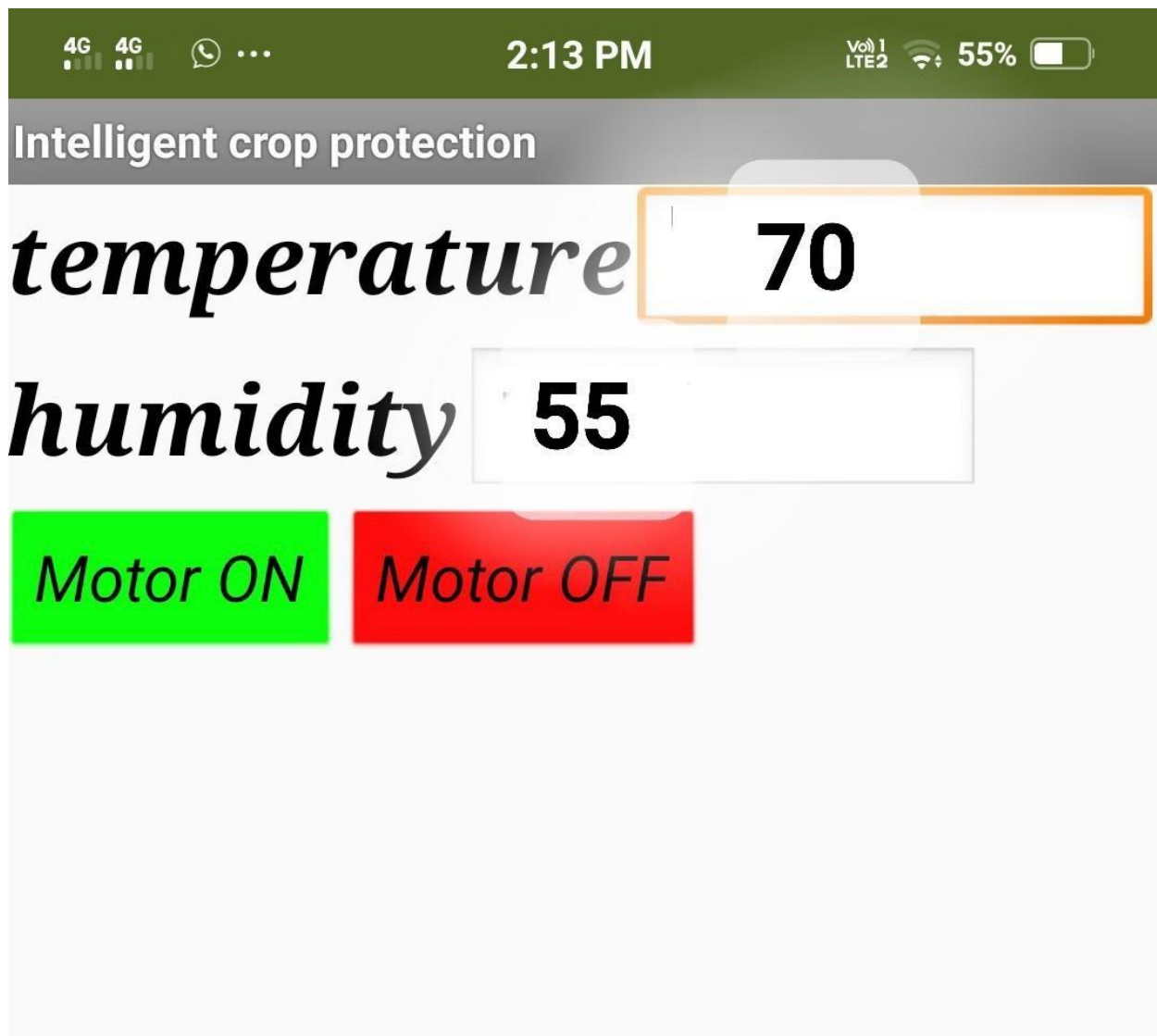


Fig: Mobile App

CHAPTER 6

CONCLUSION

The problem of crop vandalization by wild animals has become a major social problem in the current time. It requires urgent attention and an effective solution. Thus this project carries a great social relevance as it aims to address this problem. Farmland protection and surveillance based system which is low cost, and also consumes less energy. The main aim is to prevent the loss of crops and to protect the area from interest to the agricultural Areas such a system will be helpful to the farmers in protecting their orchards and fields and save them from significant financial losses.

REFERENCES

- [1] <https://www.irjet.net/archives/V5/i3/IRJET-V5I3895.pdf> · PDF file A SMART FARMLAND USING RASPBERRY PI CROP PREVENTION AND ANIMAL INTRUSION DETECTION SYSTEM S. Santhiya¹, Y. Dhamodharan², N E ...
Design and Implementation of an
- [2] <https://smartinternz.com/guided-project/intelligent-crop-protection...> Project Description. An intelligent crop protection system helps the farmers in protecting the crop from the animals and birds which destroys the crop. This system also helps farmers to monitor
- [3] <https://www.collectionbooks.net/pdf/crop-protection-reference> 15/01/2022 · Download or read book entitled Crop Protection Reference written by and published by Unknown online. This book was released on 15 January 2022 with total page null pages
- [4] <https://amrita.edu/project/intelligent-animal-attack-prevention-system...> We propose a solar powered, IoT based intelligent system that can be used to prevent crop damage due to wild animal attacks. The system implements IoT technology along with simple
- [5] <https://www.frontiersin.org/journals/plant-science/sections/...> In the future, plant protection will become sustainable, intelligent, and towards the era of unmanned smart plant protection. This section aims to provide a forum for researchers from diverse interdisciplinary areas to present their latest achievements for pla
- [6] <https://ebookswbiz.com/pdf/crop-protection-reference> If the content Crop Protection Reference not Found or Blank , you must refresh this page manually or visit our sister site Crop Protection Reference DOWNLOAD READ ONLINE.
Download Crop

APPENDEX

Image

```
import cv2
img = cv2.imread("sparrow.jpg", 1)
print(img)
cv2.imshow("bird", img)
k = cv2.waitKey(0)
if k == 27:
    cv2.destroyAllWindows
```

MultiPartUpload

```
import ibm_boto3
    from ibm_botocore.client import Config, ClientError

    # Constants for IBM COS values
    COS_ENDPOINT = "https://s3.jp-tok.cloud-object-
storage.appdomain.cloud" # Current list available at https://control.cloud-ob
ject-storage.cloud.ibm.com/v2/endpoints
    COS_API_KEY_ID =
"SP5ocpLq81KIfmp5STmK16bZGQ9BfGGG6QzCvgWkFPf
7" # eg "W00YiRnLW4a3fTjMB-odB-2ySfTrFBIQQWanc--
P3byk" COS_AUTH_ENDPOINT = https://iam.cloud.ibm.com/identity/token

    COS_RESOURCE_CRN = "crn:v1:bluemix:public:cloud-object-storage
:global:a/c884a8feeb094e7185cc3ba3f64639ee:67b6223a-cbda-4023-8e71-
721e65adea0a::" # eg "crn:v1:bluemix:public:cloud-object-
storage:global:a/3bf0d9003abfb5d29761c3e97696b71c:d6f04d83-6c4 f-
4a62-a165-696756d63903::"

    # Create resource
    cos = ibm_boto3.resource("s3", ibm_api_key_id=COS_API_KEY_I
D, ibm_service_instance_id=COS_RESOURCE_CRN, ibm_auth_
endpoint=COS_AUTH_ENDPOINT, config=Config(signature_v
ersion="oauth"), endpoint_url=COS_ENDPOINT
)
```

```

def multi_part_upload(bucket_name, item_name, file_path):

try:
    print("Starting file transfer for {0} to bucket:
    {1}\n".format(item_name, bucket_name))

    # set 5MB chunk size
    part_size = 1024*1024*5

    # set threshold to 15
    file_threshold = 1024*1024*15

    # set the transfer threshold and chunk
    transfer_config = ibm_boto3.s3.transfer.TransferConfig(
        multipart_threshold=file_threshold, multipart_chunk
        size=part_size

```

```

    )

    # the upload_fileobj method will automatically execute a
    # multi-part upload in 5 MB chunks for all files over 15 MB
    with open(file_path, "rb") as
        file_data: cos.Object(bucket_name, item_name).upload_fileobj(
            Fileobj=file_data, Config=transfer_config
        )

    print("Transfer for {0} Complete!\n".format
          (item_name)) except ClientError as e:
        print("CLIENT ERROR: {0}\n".format(e)) except Exception
        as e:
    print("Unable to complete multi-part upload: {0}".format(e)) multi_part_upload
    ("prasuna9", "crop.jpg", "sparrow.jpg")

```

Motor

```
import
Timeimport
sys
importibmi
otf.applicat
ionimport
ibmiotf.de
viceimport
random
#ProvideyourIBMWatsonDevic
eCredentialsorganization="7d0e
fs"
deviceType
=
"raspberry"
deviceId="
12345"auth
Method =
"token"aut
hToken="1
23456789"
defmyCommandCallback(cmd):
    print("Commandreceived:%s"%cmd.data)#Commands
try:
    deviceOptions={"org":organization,"type":deviceType,"id":deviceId,"auth-method":
authMethod,"auth-token":authToken}
    deviceCli=ibmiotf.device.Client(d
eviceOptions)#.....
    .....

exceptExceptionas e:
    print("Caughtexceptionconnectingdevice:
    %s"%str(e))sys.exit()
#Connectandsendadatapoint"hello"withvalue"world"intothecloudasanevento
ftype"greeting"10 times
devi
ceCl
i.con
nect(
)whi
leTr
ue:
```

```

hum=random.
randint(10,40
)#print(hum)
temp=random.randint(30,80)
#Send Temperature & Humidity
to IBM Watsondata = {
'Temperature' : temp, 'Humidity':
hum }#print(data)
defmyOnPublishCallback():
    print("PublishedTemperature=      %sC"%temp,"Humidity=%s%%"%
hum,"toIBMWatson")
    success = deviceCli.publishEvent("cropprotection",
"json", data, qos=0,on_publish=myOnPublishCallback)
    ifnotsuccess:
        print("Not
connected to
IoTTF")time.sleep(
2)
deviceCli.commandCallback=myCommandCallback

```

```
#Disconnect the device and application from the cloud device
Cli.disconnect()
```

Text To Speech

```
from ibm_watson import TextToSpeechV1
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator
from playsound import playsound

authenticator = IAMAuthenticator('fumbqyCTaCbMHNOsJDHaFn2eIkGDO2P9UbEiAIFd1HLE')
text_to_speech = TextToSpeechV1(authenticator=authenticator)

text_to_speech.set_service_url('https://api.eu-gb.text-to-speech.watson.cloud.ibm.com/instances/13787e5d-dfd8-4822-92e2-f657bd5e2ba2')

with open('Hello_world.mp3', 'wb') as audio_file:
    audio_file.write(text_to_speech.synthesize('Hello ho wru', voice='en-US_AllisonV3Voice', accept='audio/mp3').get_result().content)
playsound('Hello_world.mp3')
```

Fast To SMS

```
import requests

url = "https://www.fast2sms.com/dev/bulk"
querystring = {"authorization": "AoG6FDNZMByHXcJri5z792RUx0vOfelaQd4ngk3STWYVElCjwhCYW2sE74wAKg3iGban9ZJ1hdym0HQD", "sender_id": "FSTSMS", "message": "The crop was filled with insects", "language": "english", "route": "p", "numbers": "7993554909,6303"}

response = requests.get(url, params=querystring)
```

```
494845"}}
```

```
headers= {  
  'cache-control': "no-cache"  
}
```

```
response=requests.request("GET",url,headers=headers,params=
querystring)print(response.text)
```

Face Detective

```
importcv2
importnumpy asnp

face_classifier=cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
eye_classifier=cv2.CascadeClassifier("haarcascade_eye.xml")

#It willreadthe
firstframe/imageofthevideovid
eo=cv2.VideoCapture(0)

whileTrue:
    #capture the first
    framecheck,frame=video.read
    ()
    gray=cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)

    #detectthefacesfromthevideousingdetectMultiSca
    lefunctionfaces=face_classifier.detectMultiScale(
    gray,1.3,5)eyes=eye_classifier.detectMultiScale(
    gray,1.3,5)

    #drawingrectangleboundriesforthed
    etectedfacefor(x,y,w,h)in faces:
        cv2.rectangle(frame, (x,y), (x+w,y+h),
        (127,0,255), 2)cv2.imshow('Face
        detection', frame)
    #drawingrectangleboundriesforthed
    etectedeyesfor(ex,ey,ew,eh) ineyes:
        cv2.rectangle(frame,(ex,ey),(ex+ew,ey+eh),(
        127,0,255),2)cv2.imshow('Face detection',
        frame)

    #waitKey(1)-forevery1millisecond
    newframewillbecapturedKey=cv2.waitKey(1)
    if
        Key=
        =ord('
        q'):#r
        77
```

lease

the

camer

avide

o.rele

ase()#

destro

yallwi

ndow

s

cv2.destroyAl

lWindows()br

eak

```
response=requests.request("GET",url,headers=headers,params=
querystring)print(response.text)
```

Face Detective

```
importcv2
importnumpy asnp

face_classifier=cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
eye_classifier=cv2.CascadeClassifier("haarcascade_eye.xml")

#It willreadthe
firstframe/imageofthevideovid
eo=cv2.VideoCapture(0)

whileTrue:
    #capture the first
    framecheck,frame=video.read
    ()
    gray=cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)

    #detectthefacesfromthevideousingdetectMultiSca
    lefunctionfaces=face_classifier.detectMultiScale(
    gray,1.3,5)eyes=eye_classifier.detectMultiScale(
    gray,1.3,5)

    #drawingrectangleboundriesforthed
    etectedfacefor(x,y,w,h)in faces:
        cv2.rectangle(frame, (x,y), (x+w,y+h),
        (127,0,255), 2)cv2.imshow('Face
        detection', frame)
    #drawingrectangleboundriesforthed
    etectedeyesfor(ex,ey,ew,eh) ineyes:
        cv2.rectangle(frame,(ex,ey),(ex+ew,ey+eh),(
        127,0,255),2)cv2.imshow('Face detection',
        frame)

    #waitKey(1)-forevery1millisecond
    newframewillbecapturedKey=cv2.waitKey(1)
    if
```

```
Key==o  
rd('q')  
:#re
```

```
lease the  
camer  
avide  
o.rele  
ase()#  
destro  
yallwi  
ndow  
s  
cv2.destroyAl  
lWindows()br  
eak
```

```

client.connect()

database_name= "sample"
my_database=client.create_database(database_name)
if my_database.exists():
    print(f"'{database_name}' successfully created.")

def multi_part_upload(bucket_name, item_name, file_path):
    try:
        print("Starting file transfer for {0} to bucket: {1}\n".format(item_name, bucket_name))
        # set 5 MB chunks
        part_size=1024 * 1024 * 5

        # set threshold to
        15
        MB
        file_threshold
        =1024*1024*15

        # set the transfer threshold and chunk
        size
        transfer_config=ibm_boto3.s3.transfer.TransferConfig(
            multipart_threshold=file_threshold,
            multipart_chunksize=part_size
        )

        # the upload_fileobj method will automatically execute a
        multi-part upload in 5 MB chunks for all files over 15 MB
        with open(file_path, "rb") as
            file_data:
                cos.Object(bucket_name, item_name).upload_fileobj(
                    Fileobj=file_data,
                    Config=transfer_config
                )

        print("Transfer for {0} Complete!\n".format(item_name))
    except ClientError as be:
        print("CLIENT ERROR: {0}\n".format(be))
    except Exception as e:
        print("Unable to complete multi-part upload: {0}".format(e))

```

```
#It will read the
first frame/image of the video
vid = cv2.VideoCapture(0)

while True:
    #capture the first
    frame, check = video.read()
    if not check:
        break
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

```

#detectthefacesfromthevideousingdetectMultiScale
functionfaces=face_classifier.detectMultiScale(
gray,1.3,5)eyes=eye_classifier.detectMultiScale(
gray,1.3,5)

print(faces)

#drawingrectangleboundariesforthed
etectedfacefor(x,y,w,h)in faces:
    cv2.rectangle(frame,(x,y),(x+w,y+h),
    (127,0,255),2)cv2.imshow('Face detection',
    frame)picname=datetime.datetime.now().strftime("%
    y-%m-%d-%H-%M")
    cv2.imwrite(picname+".jpg",frame)
    multi_part_upload("prasuna9",picname+".jpg",picname+".jpg")json_document
    ={"link":COS_ENDPOINT+"/"+"prasuna9"+"/"+picname+".jpg"}
    new_document=my_database.create_document(json_document)
    #Checkthatthethedocument
    existsinthe da
    print(f"Documentsuccessfullycreated.")

url="https://www.fast2sms.c

om/dev/bulk"querystring=
{"authorization":"AoG6FDNZMByHXcJri5z792RUx0vOfelaQd4ngk3STWYVE
lCjwhCYW2s
E74wAKg3iGban9ZJ1hdym0HQD","sender_id":"FSTSMS","message":"The
crop was
filledwithinsects","language":"english","route":"p","numbers":"7993554909,6303
494845"}

headers= {
    'cache-control':'no-cache'
}

```

```

response=requests.request("GET",u
rl,headers=headers,params=queryst
ring)print(response.text)
#drawingrectangleboundriesforthed
etectedeyesfor(ex,ey,ew,eh) ineyes:
    cv2.rectangle(frame,(ex,ey),(ex+ew,ey+eh),(
    127,0,255),2)cv2.imshow('Face detection',
    frame)
    break

```

```

#waitKey(1)-forevery1millisecond
newframewillbecapturedKey=cv2.waitKey(1)
if
    Key=
    =ord('
    q'):#re
    lea

```

```

video.
releas
e())#de
stroya
llwind
ows
cv2.destroyAllWindows()

```

Bucket Creation

```

import ibm_boto3
from ibm_botocore.client import Config, ClientError

# Constants for IBM COS values
COS_ENDPOINT = "https://s3.jp-tok.cloud-object-
storage.appdomain.cloud" # Current list available at https://control.cloud-
object-
storage.cloud.ibm.com/v2/endpoints
COS_API_KEY_ID = "SP5ocpLq81KIf
mp5STmK16bZGQ9BfGGG6QzCvgWkFPf7"#eg"W00YiRnLW4a3fTjMB-
odB-2ySfTrFBIQQWanc--P3byk"
COS_AUTH_ENDPOINT =
"https://iam.cloud.ibm.com/identity/token"
COS_INSTANCE_CRN = "crn:v1:bluemix:public:cloud-object-
storage:global:a/c884a8feeb094e7185cc3ba3f64639ee:67b6223a-
cbda-4023-8e71-721e65adea0a::" # eg
"crn:v1:bluemix:public:cloud-object-
storage:global:a/3bf0d9003abfb5d29761c3e97696b71c:d6f04d83-
6c4f-4a62-a165-696756d63903::"
# Create resource
cos = ibm_boto3.resource("s3", ibm_api_key_id=COS_
    API_KEY_ID, ibm_service_instance_id=COS_INS
    TANCE_CRN, ibm_auth_endpoint=COS_AUTH_E
    NDPOINT, config=Config(signature_version="oaut
    h"), endpoint_url=COS_ENDPOINT
)

def create_bucket(bucket_name):
    print("Creating new bucket: {}".format(bu
    cket_name))
    try:
        cos.Bucket(bucket_name).create(
            CreateBucketConfiguration={
                "LocationConstraint": "jp-tok-standard"
            }
        )
    except ClientError as err:
        print(err)

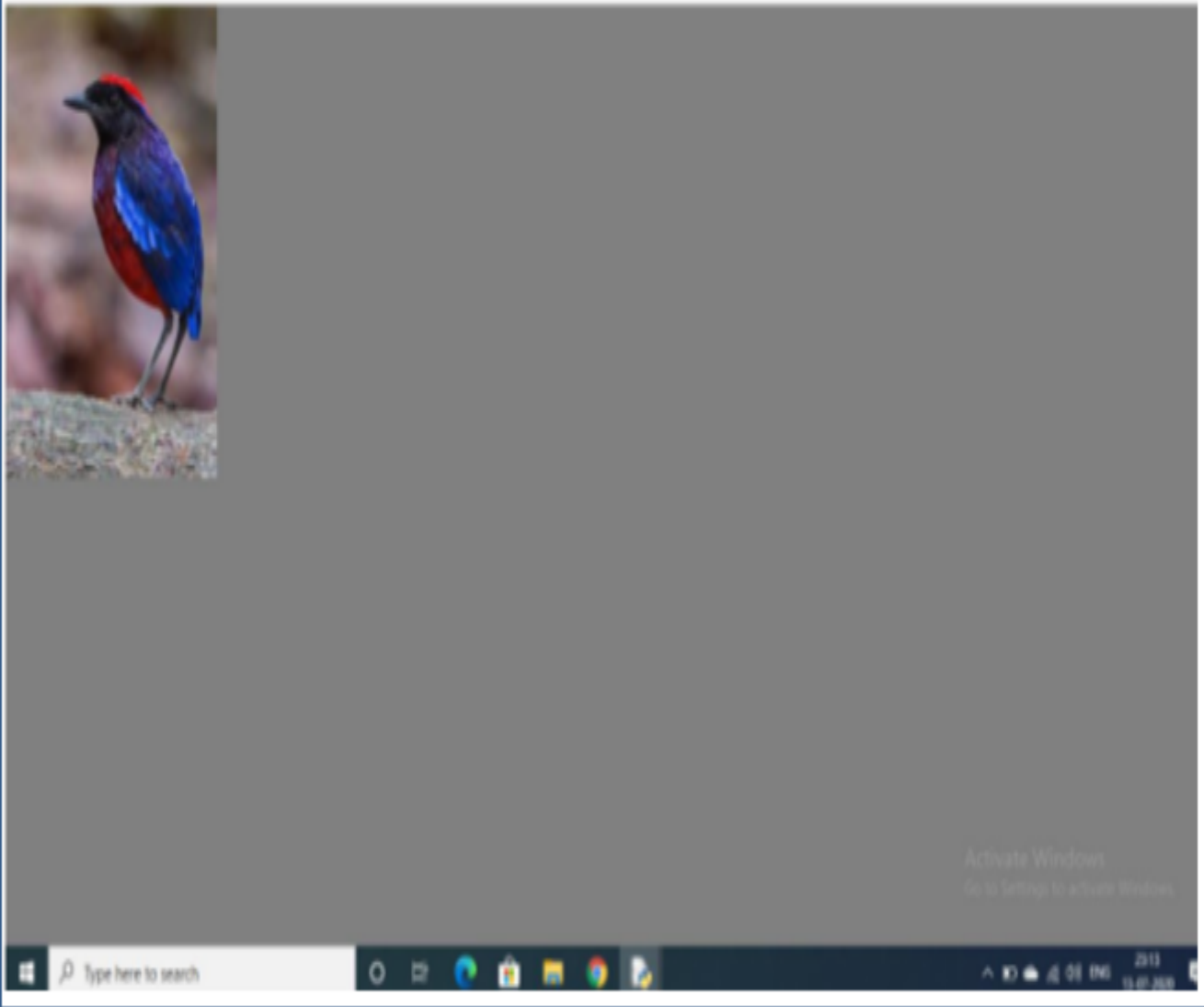
```

```
)
    print("Bucket: {0}
created!".format(bucket_name))except
ClientErrorasbe:
    print("CLIENTERROR:{0}\
n".format(be))exceptException
ase:
    print("Unabletocreatebucket:{0}".format(e))

create_bucket("prasuna9")
```

OUTPUT SCREENSHOTS

Image capture



- Fast2SMS



BIBLIOGRAPHY

- <https://thesmartbridge.com/documents/pdf/Node-red-Creation.pdf>
- <https://flows.nodered.org/node/node-red-dashboard>
- <http://www.iotgyan.com/gateways/raspberry-pi>