“Weather Monitoring Station”

A REPORT

On

Inter Department Project of III B. Tech. I Semester

By

VED PRAKASH SONI
(201FA07050, IT)

VIKASH KUMAR
(201FA07078, IT)

RANJEET KUMAR DAS
(201FA07082, IT)

VIGNAN’S
Foundation for Science, Technology & Research
(Deemed to be University)
-Estd. u/s 3 of UGC Act 1956

DEPARTMENT OF INFORMATION TECHNOLOGY
2021-2022

VIGNAN'S
Foundation for Science, Technology & Research
(Deemed to be University)
-Estd. u/s 3 of UGC Act 1956

DECLARATION

We hereby declare that the inter department project report entitled "**Weather Monitoring Station**" submitted to the Department of Information Technology, Vignan's Foundation for Science, Technology and Research, deemed to be University. This report is the work done by us in the **Department of Information Technology** by collaborating with the departments "**Electronic and Communication Engineering**".

Place:
Date:

Signature of Students

VIGNAN'S

Foundation for Science, Technology & Research

(Deemed to be University)

-Estd. u/s 3 of UGC Act 1956

CERTIFICATE

This is to certify that the Inter Department Project Report entitled "**Weather Monitoring Staton**" is being submitted by **Ved Prakash Soni (201FA07050), Vikash Kumar (201FA07078) & Ranjeet Kumar Das (201FA07082)** in partial fulfilment for the award of B. Tech Degree in Information Technology at Vignan's Foundation for Science, Technology and Research, deemed to be University It is a record of bonafide work carried out by them in Department of Information Technology, Vignan's Foundation for Science Technology and Research under the supervision of "**Dr. Ramakrishnan R**".

Signature of Project Guide **Signature of HOD**

VIGNAN'S
Foundation for Science, Technology & Research
(Deemed to be University)
-Estd. u/s 3 of UGC Act 1956

ACKNOWLEDGEMENTS

Inter department project is an opportunity to work collaboratively with other departments such as ECE, EEE, Automobile, Bio Technology, Food Technology etc., students and helps in gaining knowledge in other department domains.

We express our gratitude towards the Management for providing opportunity to work and implement inter department projects.

We feel it our responsibility to thank "**Dr. Ramakrishnan R**" under whose valuable guidance that the project came out successfully after each stage. It is a great pleasure for us to express our sincere thanks to **Prof. Dr. N. Veeranjaneyulu**, **HOD, IT** of **VFSTR** Deemed to be University, for providing me an opportunity to do Inter department projects.

We extend our wholehearted gratitude to all our faculty members of Department of Information Technology who helped us in our academics throughout course.

Finally, we wish to express thanks to our family members for the love and affection overseas and forbearance and cheerful depositions, which are vital for sustaining effort, required for completing this work.

With Sincere regards:
Ved Prakash Soni (201FA07050)
Vikash Kumar (201FA07078)
Ranjeet Kumar Das (201FA07082)

CONTENTS

S.NO.	DESCRIPTION	PAGE NO.
1.	ABSTRACT	06
2.	PROBLEM STATEMENT	07
3.	INTRODUCTION	08
4.	SYSTEM ENVIRONMENT	09
5.	ANALYSIS	10
6.	DESIGN	11
7.	IMPLEMENTATION	12
8.	RESULT	14
9.	TESTING	25
10.	CONCLUSION & FUTURE SCOPE	27

ABSTRACT

Internet of Things (IoT) has provided a promising opportunity to build powerful industrial systems and applications by leveraging the growing ubiquity of RFID, wireless, mobile and sensor devices. A wide range of industrial IoT applications have been developed and deployed in recent years. In an effort to understand the development of IoT in industries, this paper reviews the current research of IoT, key enabling technologies, major IoT applications in industries and identifies research trends and challenges. A main contribution of this review paper is that it summarizes the current state-of-the-art of IoT in industries systematically. The advancement of Automation technology, life is getting simpler and easier in all aspects. In today’s world Automatic systems are being preferred over manual system. With the rapid increase in the number of users of internet over the past decade has made Internet a part and parcel of life, and IoT is the latest and emerging internet technology. Internet of things is a growing network of everyday object-from industrial machine to consumer goods that can share information and complete tasks while you are busy with other activities. This paper proposes that the industrial monitoring by using Gas sensor, Temperature sensor, Mems, Piezoelectric Sensor values to read the value and monitoring using Thingspeak system via Raspberry pi.

ThingSpeak is an application platform for the Internet of Things. ThingSpeak allows you to build an application around data collected by sensors. At the heart of ThingSpeak is a ThingSpeak Channel. A channel is where you send your data to be stored. Each channel includes 8 fields for any type of data, location fields, and 1 status field. Once you have a ThingSpeak Channel you can publish data to the channel, have ThingSpeak process the data, and then have your application retrieve the data.

PROBLEM STATEMENT

Weather monitoring system being very hand for better performance of the solar plants has the issue of higher cost. The hard drive based data logging facility requires a separate computer setup for its operation and many a times, the data stored cannot be manipulated in a useful mean. These two problems are the primary concerns when you consider a weather monitoring system and we have come up with cost effective innovative solution to provide the layman’s weather monitoring system.

- Limited way for user to know about weather such as temperature, humidity and pressure
- User can't be alerted of the strong winds , heat waves or any other weather related emergency .
- Difficulty in making weather forecasts without data .

INTRODUCTION

Environment is a significant piece of human existence. Sensors are significant segments that are utilized in measure control ventures as well as in regular day to day existence to guarantee the security of building and estimating traffic stream and ecological limitations. IoT represents Internet of Things. It gives web network to cell phones, structures, vehicles and different gadgets like sensors and actuators. By giving correspondence of board frameworks through hardware, programming, sensors and actuators; These things can be gathered and traded with information. By utilizing IoT objects that can be recognized or controlled distantly from a current organization. It gives the capacity to associate the actual world to PC frameworks. IoT Improves Efficiency, Speed. The default climate station is an apparatus for estimating and recording climate boundaries utilizing sensors without human intercession. Assessed boundaries can be put away in the incorporated information lumberjack or can be moved distantly by means of an association interface. In the event that the information is put away in an information lumberjack, the recorded information should be actually followed on a PC a short time later to be prepared. Hence, the correspondence framework is a fundamental piece of the computerized climate station. Today, computerized climate channels are accessible as business items with an assortment of settings and alternatives. Albeit programmed climate stations can be fabricated and utilized in distant pieces of Sri Lanka to decrease the expense of support of climate stations, up to this point the consideration was paid to the development of climate stations. what's more, utilize those devices locally. Programmed climate channels are intended for colleges by interfacing sensors that screen environment cutoff points to PCs/information coders accessible on interchanges hardware or in continuous and inconsistent ports of climate information printed copy. As of late, the University of Colombo has set up a computerized climate station with worked in USB correspondence capacities and information passage capacities. The current assignment is to develop past advancement. The primary motivation behind this venture was to set up a free climate station with distant correspondence frameworks for the discovery and transmission of environment limits. Distant checking of ecological limits is vital for different modern plans and cycles. Before, standard environment checking frameworks dependent on mechanical and electrical gadgets had restrictions like helpless solidness, the requirement for human intercession, parallax-related blunders, and sturdiness. Climate estimates should be solid and exact, regardless of how successful. Likewise, it ought to give simple admittance to every one of the deliberate boundaries. Affectability and precision of estimations might change, and the area of a climate figure can decide the exactness and unwavering quality of a climate information assortment. Normally the client is restricted to choices given by the producer. Indeed, even with minor changes in boundary checking or information preparing, business gadgets at this point don't work. For certain applications it is important to have adaptable and customizable arrangements. Also that business gadgets can be over the top expensive for certain applications purposes. This paper is about a climate conjecture framework intended for explicit purposes.

SYSTEM ENVIRONMENT

Existing System

- Manually Monitoring the Industrial application
- By using the GSM technology, it will take more time to get the exact situation
- CCTV camera monitoring is possible but cant able to sense the gas, temperature, and position of the valves.

Proposed System

- The Internet of Things is regarded as the third wave of information technology after Internet and mobile communication network, which is characterized by more thorough sense and measure, more comprehensive interoperability and intelligence.
- IOT Consumes the time and monitoring the exact situation.

REQUIREMENTS

Hardware

- Raspberry pi 4
- Temperature Sensor
- Rain Sensor
- DTH 11 Sensor
- Jumper wires
- Bread Board
- Data conveter
- Ethernet Cable
- 5 or 3 volts Battery

Software

- OS: Raspbian OS
- Python Language
- ThingSpeak

PROJECT DESCRIPTION

Have you ever wondered about the rapid transition in the weather when you are travelling on a sunny day, and after a mile, it turns into a storm? Or the fluctuations of the weather that we see in the mountain areas?

If yes, then you are in the right place!

The introduction of IoT (Internet of Things) establishes the platform for the world to showcase hi-tech machine-to-machine interaction ranging from connected cars to smart cities to weather monitoring systems and smart homes. The IoT revolution is transforming the way humans interact with machines.

The Weather Monitoring System using the IoT abstract is one such application of IoT that has paved the way for organizations to create new and efficient solutions. Businesses are rapidly adopting smart management systems that improve the accuracy of weather forecasts and transform IoT to 'Weather of Things' that collect weather data from drones, connected vehicles, wireless signals, and other IoT devices.

So let's dive into one of the most interesting topics of Weather Monitoring Systems using IoT.

Because of the rapidly changing climate, the weather forecast is uncertain and inaccurate these days. As a result, the Weather Reporting System is primarily utilized to monitor the constantly changing climatic and weather conditions over-regulated areas like homes, industry, agriculture, and so on.

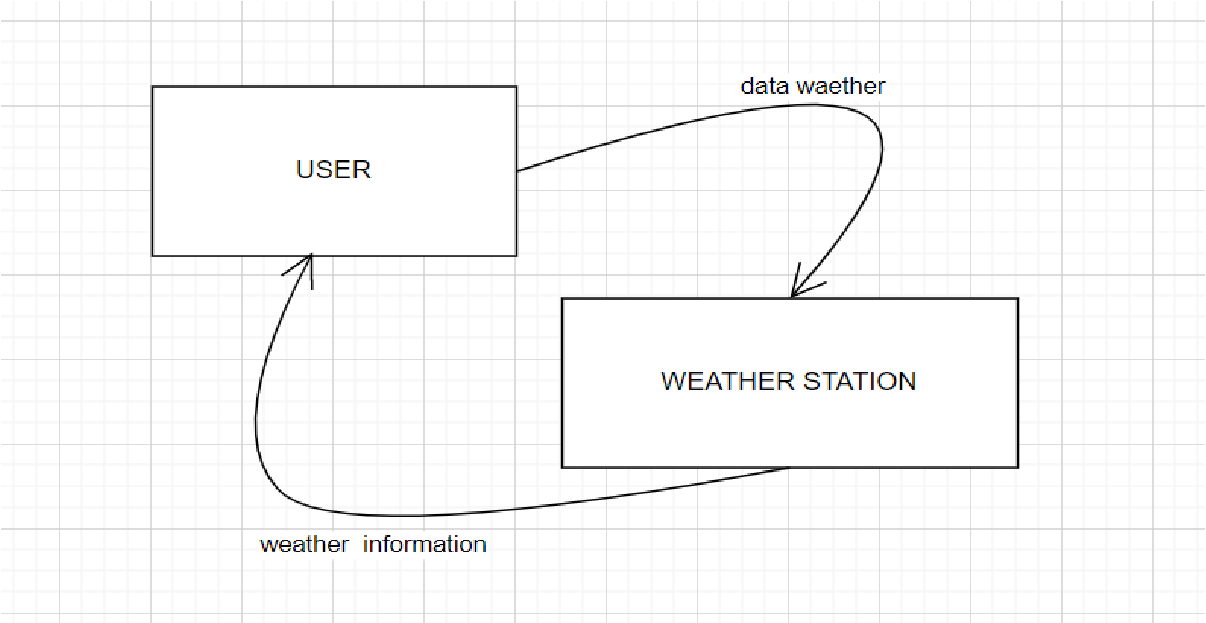
When objects like an environment furnished with sensor devices, microcontrollers, and different software applications become a self-monitoring and self-protecting environment, it is called a smart environment.

Similarly, here, the system uses sensors to monitor and adjust environmental parameters such as temperature, CO levels, and relative humidity. Then, it sends the data to a web page to plot the sensor data, shown as graphical statistics. The data updated from this system can be accessed on the internet from anywhere in the world. The embedded system enables the user to access the various criteria and store the data in the cloud.

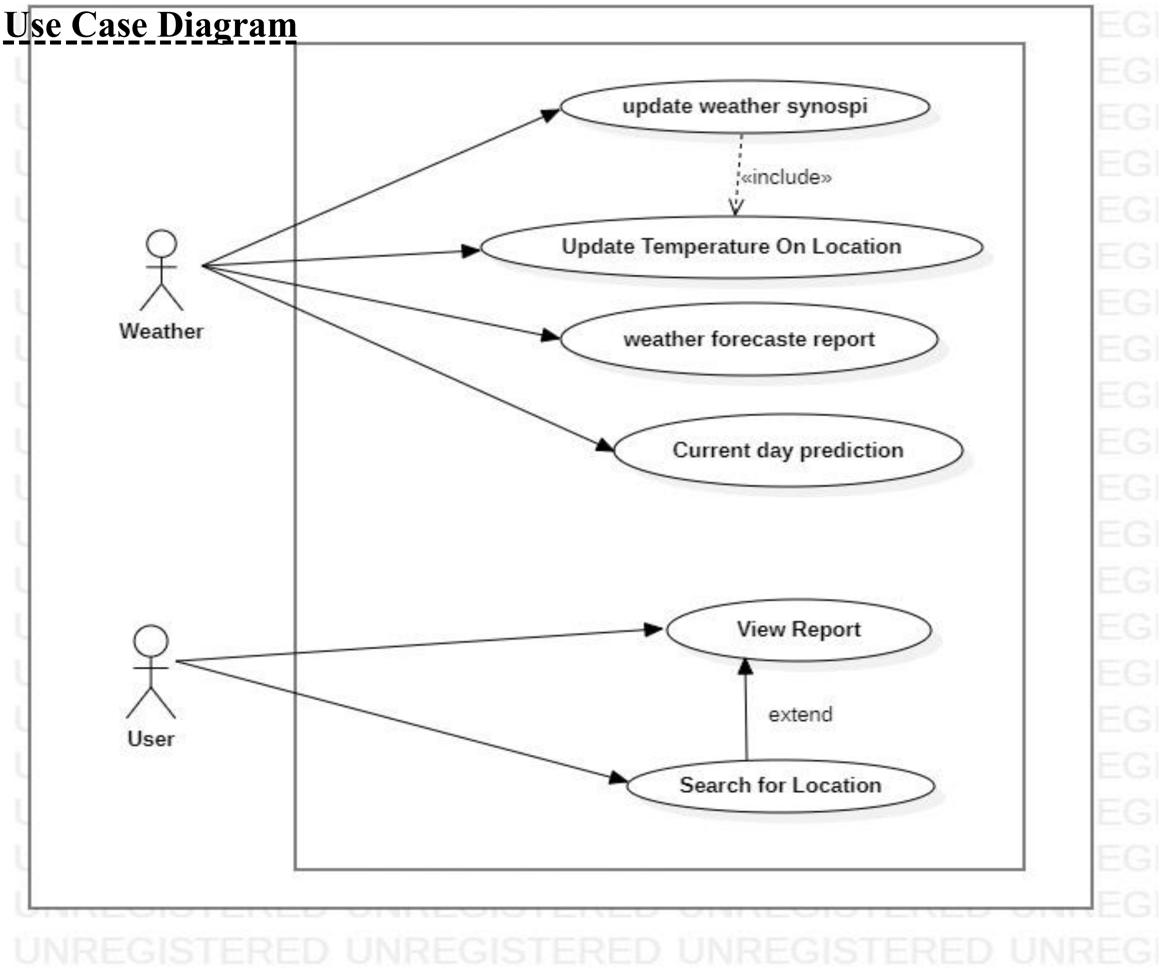
Hence, the Internet of Things (IoT) is the core root of linking all the sensors to the internet and monitoring the weather in real-time.

DESIGN

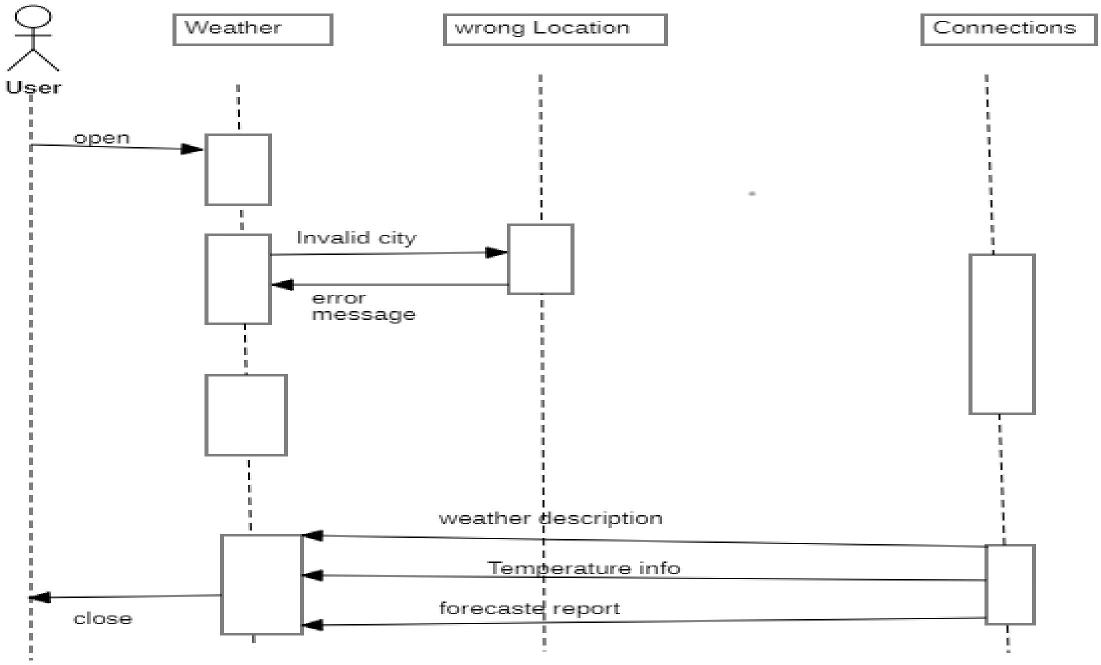
Framework:-



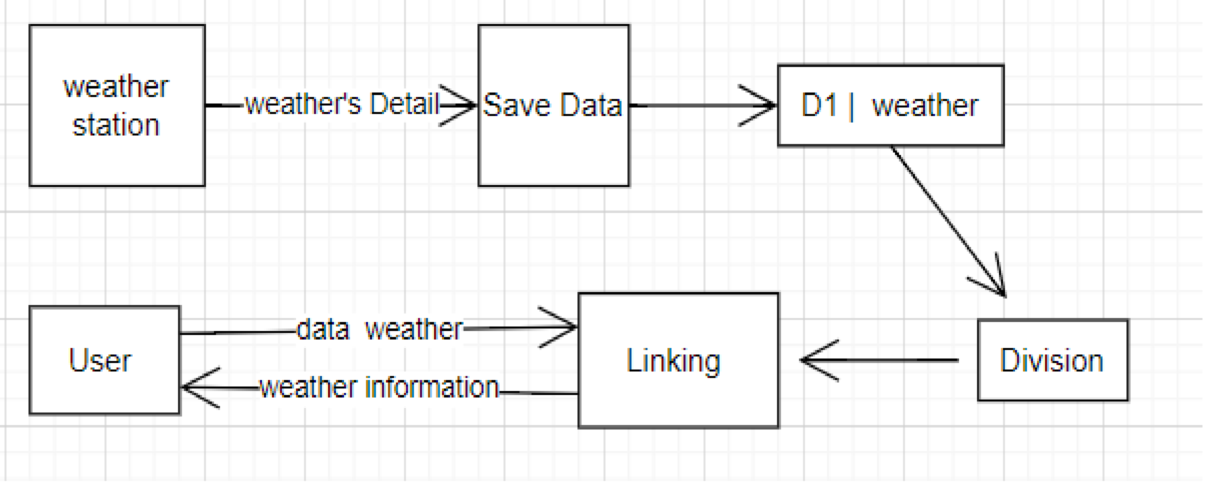
Use Case Diagram



Sequence Diagram



Data Flow Diagram



IMPLEMENTATION

Web.html:-

```
<!DOCTYPE html>
<html lang="en">
 <head>
   <meta charset="UTF-8" />
   <meta http-equiv="X-UA-Compatible" content="IE=edge" />
   <meta name="viewport" content="width=device-width, initial-scale=1.0 maximum-scale=1, user-
scalable=no"/>
   <title>Document</title>
   <script src="https://code.jquery.com/jquery-3.6.1.min.js"></script>
   <style>
     body {
   background: linear-gradient(rgb(37, 37, 112), rgb(0, 2, 10));
   height: 100vh;
   padding: 0;
   margin: 0;
   user-select: none;
}

#bapofall {
   background-color: rgba(250, 235, 215, 0);
   height: 100%;
   width: 100%;
   display: flex;
   flex-direction: column;
   justify-content: space-between;
   align-items: center;
}
```

```
#upperbap {
    background: linear-gradient(rgb(76, 204, 255), rgb(54, 134, 238));
    width: 100%;
    height: 73%;
    border-bottom-left-radius: 20px;
    border-bottom-right-radius: 20px;
}

#lowerbap {
    background-color: rgba(20, 115, 199, 0);
    width: 100%;
    height: 27%;
    display: flex;
    justify-content: center;
    align-items: flex-end;
}

#lowerbetas {
    width: 95%;
    height: 98%;
    display: flex;
    flex-direction: column;
    justify-content: space-between;
    align-items: center;
}

#lowerbetakapehla {
    width: 100%;
    height: 20%;
    background-color: rgba(52, 52, 138, 0);
    border-radius: 10px;
    display: flex;
    justify-content: space-between;
    align-items: center;
}

#whichday {
    width: 45%;
    padding-left: 5%;
    height: 100%;
    display: flex;
    justify-self: center;
    align-items: center;
    color: rgb(219, 219, 219);
    font-family: Arial, Helvetica, sans-serif;
    font-weight: bold;
}

#showallforecast {
    width: 45%;
    padding-right: 5%;
    height: 100%;
    display: flex;
    justify-content: flex-end;
    align-items: center;
    color: rgb(194, 194, 194);
    font-weight: 300;
    font-family: Arial, Helvetica, sans-serif;
}
```

```
#lowerbetakadusra {
    width: 100%;
    height: 80%;
    display: flex;
    justify-content: space-evenly;
    align-items: center;
    background-color: rgba(111, 111, 255, 0);
}

#minione {
    width: 23%;
    height: 80%;
    /* background: linear-gradient(rgb(76, 204, 255), rgb(54, 134, 238)); */
    background-color: rgba(101, 101, 201, 0.072);
    box-shadow: 0px 0px 3px rgba(87, 155, 232, 0.547);
    border-radius: 10px;
    display: flex;
    flex-direction: column;
    justify-content: space-evenly;
    align-items: center;
}

#minitwo {
    width: 23%;
    height: 80%;
    background-color: rgba(101, 101, 201, 0.072);
    box-shadow: 0px 0px 3px rgba(87, 155, 232, 0.547);
    border-radius: 10px;
    display: flex;
    flex-direction: column;
    justify-content: space-evenly;
    align-items: center;
}

#minithree {
    width: 23%;
    height: 80%;
    background-color: rgba(101, 101, 201, 0.072);
    box-shadow: 0px 0px 3px rgba(87, 155, 232, 0.547);
    border-radius: 10px;
    display: flex;
    flex-direction: column;
    justify-content: space-evenly;
    align-items: center;
}

#minifour {
    justify-content: center;
    align-items: center;
    border-radius: 20px;
    color: white;
    font: 1.5em arial ;
}

#weatherlocation #weatherplace {
    display: flex;
    justify-content: center;
    align-items: center;
    width: 95%;
    height: 35%;

    color: white;

    background-color: rgba(255, 255, 255, 0);

    border-radius: 5px;

    font-family: Arial, Helvetica, sans-serif;

    font-weight: bold;

    font-size: large;

}


#weatherlocation #weathercountry {

    display: flex;

    justify-content: center;

    align-items: center;

    width: 95%;

    height: 35%;

    color: white;

    background-color: rgba(255, 255, 255, 0);

    border-radius: 5px;

    font-family: Arial, Helvetica, sans-serif;

}


#weathermiddlebody {

    width: 100%;

    height: 70%;

    display: flex;

    flex-direction: column;

    justify-content: space-evenly;

    align-items: center;

    border-radius: 10px;

    background-color: rgba(245, 245, 220, 0);
```

```
}

#weatherlogo {
    width: 90%;
    height: 55%;
    border-radius: 10px;
    display: flex;
    justify-content: center;
    align-items: center;
    /* background-image: url("https://cdn-icons-png.flaticon.com/512/1146/1146860.png"); */
    background-size: contain;
    background-repeat: no-repeat;
    background-position: center;
}

#tempvalue {
    display: flex;
    justify-content: center;
    align-items: center;
    width: 80%;
    height: 25%;
    border-radius: 10px;
    background-color: rgba(255, 228, 196, 0);
    color: white;
    font-size: 5em;
    font-family: Arial, Helvetica, sans-serif;
    font-weight: bolder;
}
```

```
#weathercondition {
    display: flex;
    justify-content: center;
    align-items: flex-start;
    width: 80%;
    height: 10%;
    border-radius: 10px;
    background-color: rgba(255, 228, 196, 0);
    color: white;
    font-size: 30px;
    font-family: Arial, Helvetica, sans-serif;
    font-weight: bolder;

#weatherdate {
    display: flex;
    justify-content: center;
    align-items: flex-start;
    color: white;
    font-size: 15px;
    font-family: Arial, Helvetica, sans-serif;
    font-weight: bolder;
}
#weatherlowerbody {
    width: 100%;
    height: 19%;
    display: flex;
    justify-content: space-evenly;
    align-items: center;
    background-color: rgba(188, 249, 249, 0);
    border-radius: 10px;
```

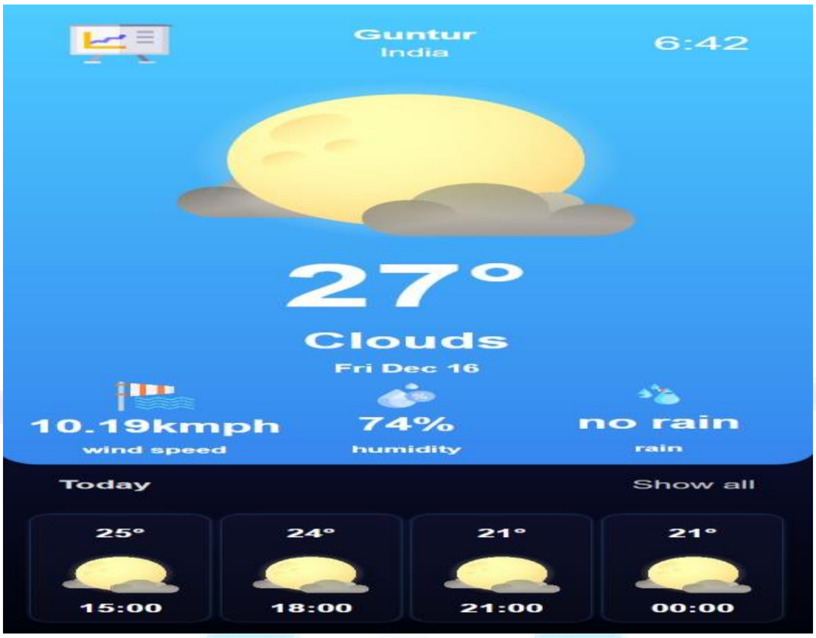
```
}

#weatherwind {
    width: 30%;
    height: 100%;
    display: flex;
    flex-direction: column;
    justify-content: space-evenly;
    align-items: center;
    background-color: rgba(116, 116, 210, 0);
    color: white;
    color: white;
    font-size: 15px;
    font-family: Arial, Helvetica, sans-serif;
    font-weight: bolder;
    border-radius: 10px;
}

#weatherrain {
    width: 30%;
    height: 100%;
    display: flex;
    flex-direction: column;
    justify-content: space-evenly;
    align-items: center;
    background-color: rgba(116, 116, 210, 0);
    color: white;
    color: white;
    font-size: 15px;
    font-family: Arial, Helvetica, sans-serif;
    font-weight: bolder;
    border-radius: 10px;
}
    height: 30%;
    border-radius: 5px;
    background-color: rgba(250, 235, 215, 0);
    display: flex;
    font-size: x-large;
    justify-content: center;
    align-items: center;
    font-family: Arial, Helvetica, sans-serif
#weatherlowerbodydetailweatherhumidity {
    width: 100%;
    height: 20% background-size: contain;
}#weatherlowerbodyvalueweatherrain {
    width: 100%;
    height: 30%;
   background-color: rgba(250, 235, 215, 0);
  border-bottom: 1px solid white;
}forecastheader{
  width: 85%;
  height: 100%;
  display: flex;
  justify-content: center;
  align-items: center;
  font-family: Arial, Helvetica, sans-serif;
  font-size: x-large;
  color: white;
}.
```

RDHT.PY:-

```
import time

import board

import adafruit_dht

from time import sleep

from gpiozero import Buzzer, InputDevice

import urllib.request

import time

import random

dhtDevice = adafruit_dht.DHT11(board.D4,use_pulseio=False)

buzz =Buzzer(13)

no_rain = InputDevice(18)


def buzz_now(iterations):

    for x in range(iterations):

        buzz.on()

        sleep(0.1)

        buzz.off()

        sleep(0.1)



def dht():

    try:

        # Print the values to the serial port

        temperature_c = dhtDevice.temperature

        temperature_f = temperature_c * (9 / 5) + 32

        humidity = dhtDevice.humidity

        return temperature_c,humidity
    except RuntimeError as error:

        # Errors happen fairly often, DHT's are hard to read, just keep going

        print(error.args[0])

        time.sleep(2.0)

        #continue

        return -1,-1

    except Exception as error:

        dhtDevice.exit()

        raise error


def thingspeak(temp,hum,rain):

    BASE_URL ="https://api.thingspeak.com/update?api_key=0X9AP9Q0E6WA2S4S&field1={}&field2={}&field3={}".format(temp,hum,rain)

    print(BASE_URL)

    data = urllib.request.urlopen(BASE_URL)

    print(data)


while True:

    temp,hum,rain=0,0,0

    if no_rain.is_active:

        print("no rain detected")

        rain=0

    if not no_rain.is_active:

        print("its raining - get washing in!")

        rain=1

        buzz_now(5)


    temp,hum=dht()

    time.sleep(1.0)

    thingspeak(temp,hum,rain)
```

RESULT

CURRENT DATA



Guntur
India

6:42

27°

Clouds

Fri Dec 16

10.19kmph
wind speed

74%
humidity

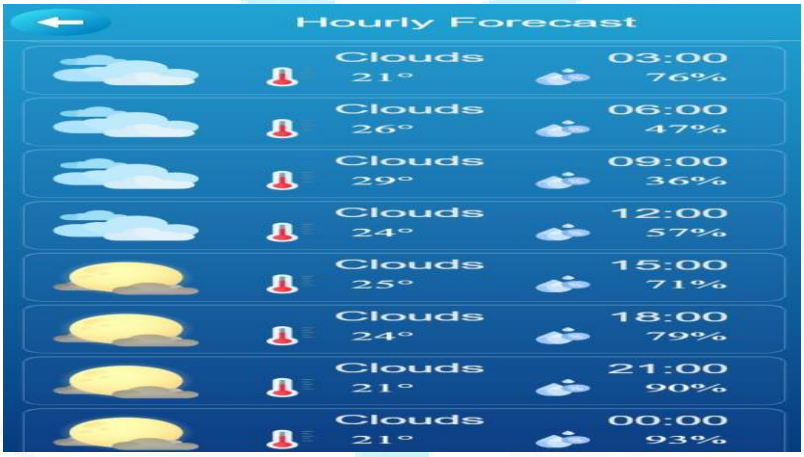
no rain
rain

Today

Show all

25°	24°	21°	21°
15:00	18:00	21:00	00:00

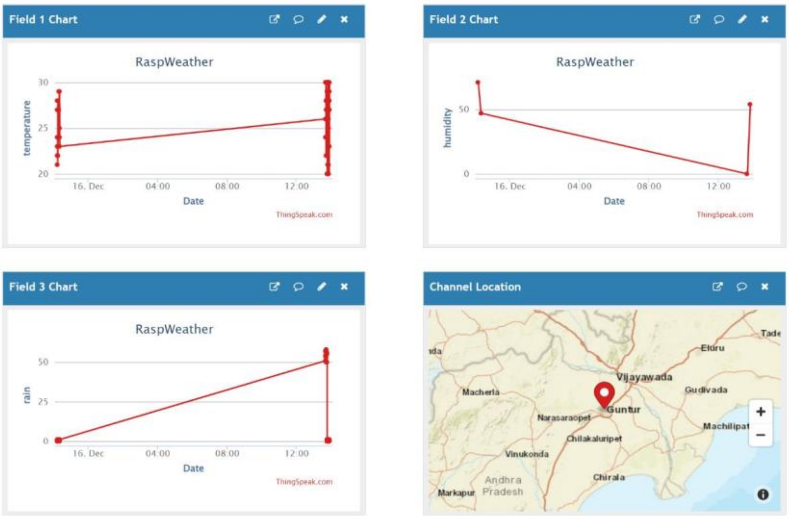
FORECASTING DATA



Hourly Forecast

Clouds 21°	03:00 76%
Clouds 26°	06:00 47%
Clouds 29°	09:00 36%
Clouds 24°	12:00 57%
Clouds 25°	15:00 71%
Clouds 24°	18:00 79%
Clouds 21°	21:00 90%
Clouds 21°	00:00 93%

THINGSPEAK_DATA



TESTING

UNIT TESTING :-

Unit testing is undertaken when a module has been created and succesfully reviewed .In order to test a single module we need to provide a complete environment ie besides the module we would require

- The procedures belonging to other modules that the module under test calls
- Non local data structures that module accesses
- A procedure to call the functions of the module under test with appropriate parameters

INTEGRATION TESTING :-

In this type of testing we test various integration of the project module by providing the input.

The primary objective is to test the module interfaces in order to ensure that no errors are occurring when one module invokes the other module.

CONCLUSION & FUTURE SCOPE

This IoT based framework gives constant checking of ecological boundaries. This framework screens temperature, moistness, pressure, height, light force and downpour water level. Information can be seen from anyplace on the planet. By utilizing this framework, the customer can constantly screen diverse ecological boundaries with no connection with extra worker. Raspberry Pi itself goes about as a worker. This is proficiently done by Raspbian working framework. This climate observing framework is planned utilizing Raspberry pi is having minimal expense, little size, low force utilization, quick information move, great execution and far off checking. This framework has a few limits, it don't have inherent Wi-Fi and implicit Real Time Clock. For systems administration direct web association must be given. Just as all sensors must be associate straightforwardly to the GPIO header. For future advancement further developed rendition of Raspberry Pi board framework can be utilized. More sensors can be added to extend the framework likewise for distant area observing sunlight based board and wind factory can be utilized for providing capacity to the framework. In this paper one potential answer for the climate estimating framework through Wi-Fi network is created. The principle aphorism was to utilize the economical segments and achieve greatest best exact framework which could screen the climate continuously application in rural grounds. Utilizing the sensors for air temperature, air mugginess, light, soil dampness, and downpour recognition in mix with Raspberry PI a model had been created. Information from the sensors is sent to cut off where it tends to be seen universally which will be effectively open to everybody.