

ABSTRACT

The main purpose of polyhouse is to protect cultivation from biotic and abiotic stresses (i.e., adverse climate changes) which cause a major decrease in production. Though semi-automatic and normal polyhouses have some facility like sprinklers, dripping system, fans etc, the overall operation is done manually by assuming the climatic parameters which require good sense and knowledge of climatic parameters as well as a physical presence.

To overcome this scenario this project has made a cost-effective and user-friendly solution with the help of **IoT** and **Machine learning**. This solution encompasses data acquisition of climate parameters (i.e., **temperature**, **humidity**, **moisture**) through sensor network, and online transmission of data to users logged on their respective web-browsers. This system allows transmitting emergency alarm signals via email. Thus, it is easily monitored by the farmer through the mobile from anywhere around the globe.

TABLE OF CONTENTS

Acknowledgement	i
Abstract	ii
List of Figures	iii
List of Tables	iv
CHAPTER 1: Introduction	
1.1 Purpose	1
1.2 Scope	2
1.3 Existing System	2
1.4 Proposed System	2
1.5 Requirements Specification	3
1.5.1 Software Requirements	3
1.5.2 Hardware Requirements	7
CHAPTER 2: Literature Survey	12
CHAPTER 3: Methodology	15
3.1 System Architecture	15
3.1.1 Flask framework	15
3.1.2 Firebase Cloud storage	15
3.1.3 Decision Tree Algorithm	16
3.2 Software Architecture	18
3.3 Flowchart showing the working	19
3.4 Diagrammatic Representations	20
3.5 Features	24
3.5.1 The sensor value is uploaded every 3 seconds	24
3.5.2 Mail alert will be sent if the values exceed optimal level	25
CHAPTER 4:	26
4.1 Testing and Results	26
4.1.1 Unit Testing	27
4.1.2 System Testing	27
4.1.3 Test strategy and approach	27
4.1.4 Test Objectives	27
4.1.5 Features to be tested	28
4.2 Output Results	28
4.2.1 Landing page	28

4.2.2 Login page	29
4.2.3 Main page	30
4.2.4 Mailing	31
Chapter 5 : Conclusion and Future scope	34
Bibliography	35
Chapter 6 : Appendix	36

LIST OF FIGURES

Figure. No	Name of the figure	Pg. No
1.1	Proposed system	2
1.2	PyCharm Community Edition	3
1.3	Arduino IDE	5
1.4	DHT11 Sensor	7
1.5	Soil Moisture Sensor	8
1.6	Arduino UNO – Wi-Fi D1 Module	9
1.7	Jumper wires	11
1.8	USB cable	11
3.1	Decision Tree Algorithm	17
3.2	Software Architecture of system	18
3.3	Flowchart Showing the working	19
3.4	Representation of classes along with attributes and classes	20
3.5	Various use cases in the proposed system	21
3.6	Flow of operation	22
3.7	Illustration of graphical workflow	23
3.8	Storing of data with 3 seconds difference	24
3.9	Mail Alert	25
4.1	Landing page	28
4.2	Login page	29
4.3	Get values page	30
4.4	Mail values crosses the limit	31
4.5	Prediction page	32
4.6	About us page	32
4.7	Contact us page	33

LIST OF TABLES

S.NO	Name of the table	Pg. No.
1.1.	Arduino Pin Configuration	10
2.1.	Paper Articles	14
4.1.	Test Cases for application	26

CHAPTER 1

INTRODUCTION

PROBLEM

Greenhouse plantation is adopted as an alternative to catch up with the supply and demand in agricultural industry because it offers plantation in any season at any location. But it alone is not sufficient to meet up with the demand.

REQUIREMENT & ANALYSIS

It is required to deploy modern technology in the agricultural industry to increase the supply. Although few greenhouses could be found offshore in which they have adopted modern technology to monitor the plantation but they happen to be working manually. In order to achieve maximum output, IOT could help us to monitor and automate certain vital process. By this project a way of predicting the plant growth as well as around the clock monitoring is achieved.

1.1 PURPOSE

Agriculture is the backbone of India where more than 50% of our population is dependent on it. Still with the classical farming methods we are unable to catch up with the demand. So, in order to increase the productivity we must adopt advanced farming method. A few agriculture sensors could be attached to sense the climatic changes in order to monitor the facility around the clock. The data captured from the sensors will be stored online hence it can be provided for different research purpose to develop hybrid seeds to decrease the time period in which the crops/plants could grow in less time. Furthermore the microcontroller attached with sensors in the field will trigger real time alerts to report any fatal change in the climatic or soil condition. From the generated alert the concerned user could take designs to minimize any effect caused which will ultimately increase the productivity. Attaching the proposed idea in greenhouse plantation is even more effective as the planted crops in the greenhouse need closer monitoring. Data captured from the sensors can be directly forwarded to field experts to take their advice on use of fertilizers and pesticides. Hence this project will ensure minimum and effective use of fertilizers and pesticides to minimize any health effects. Best part is all of it could be done sitting remotely which enhances the user to survey even on a bigger farming land or greenhouse facility

1.2 SCOPE

In order to automate the critical farming process using “**SMART AGRICULTURE MONITORING AND ALERTING SYSTEM USING IOT**” along with software product an embedded sensor node is also offered to be deployed within the Greenhouse. This project aims to revolutionaries the farming industry by enabling us to monitor the climatic variables and different soil components, automating the irrigation system and controlling the supply of fertilizers and pesticides. To achieve all this, a web platform to monitor the farming and a wireless sensor node to be deployed within the field is being offered.

1.3 EXISITING SYSTEM

A large portion of farming and agricultural activities are based on the assumptions, which at times fail. Farmers have to bear huge losses and at times they end up committing suicide. Since we know the benefits of proper soil moisture and its quality, air quality and irrigation, in the growth of crops, such parameters cannot be ignored.

1.4 PROPOSED SYSTEM

The aim of this project is to introduce the latest technology into the agriculture business and better crop production by collecting real-time status of crop and informing the farmers about it.

- SMS notification
- Valuable information collection
- Detail Data Analysis
- Cost Effective
- Easy to implement

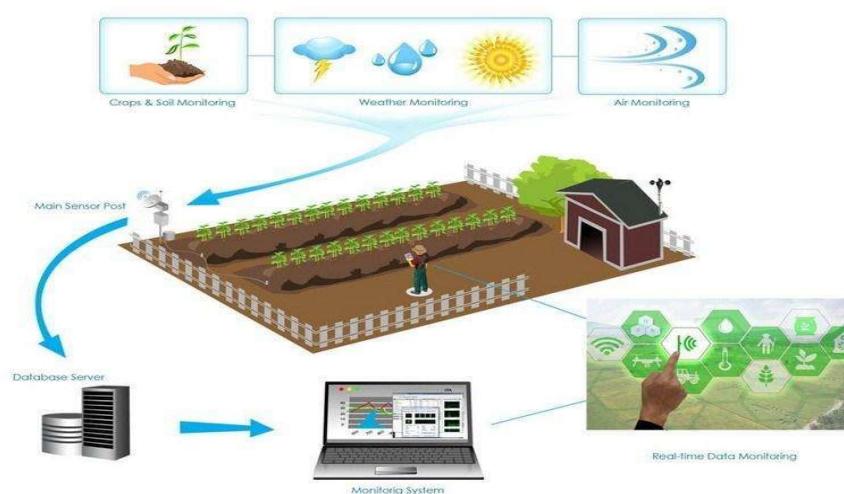


Figure 1.1 Proposed system

1.5 REQUIREMENTS SPECIFICATION

1.5.1 SOFTWARE REQUIREMENTS

Operating System : Windows 7 or above IDE

- PyCharm Community Edition

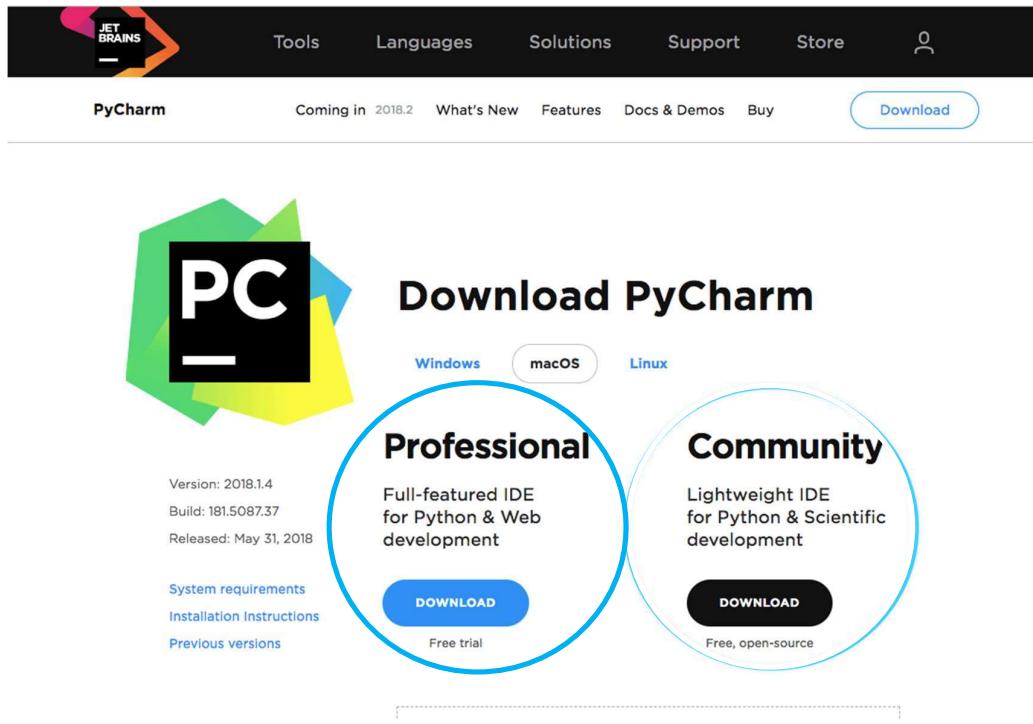


Figure.1.2 PyCharm Community Edition

PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains (formerly known as **IntelliJ**). **PyCharm** is cross-platform, with Windows, macOS and Linux versions.

The **community edition** of **PyCharm** is Apache 2 licensed: meaning it is **free** and open source and you can go to GitHub, and look at the source code. You're **free** to use it whenever, and wherever you like, including at work. Additionally, you can fork and modify it.

STEPS OF INSTALLATION:

You will have to follow the steps given below to install PyCharm on your system. These steps show the installation procedure starting from downloading the PyCharm package from its official website to creating a new project.

Step 1

Download the required package or executable from the official website of PyCharm <https://www.jetbrains.com/pycharm/download/#section=windows>. Here you will observe two versions of package for Windows.

Note that the professional package involves all the advanced features and comes with free trial for few days and the user has to buy a licensed key for activation beyond the trial period. Community package is for free and can be downloaded and installed as and when required. It includes all the basic features needed for installation.

Step 2

Download the community package (executable file) onto your system and mention a destination folder.

Step 3

Now, begin the installation procedure similar to any other software package.

Step 4

Once the installation is successful, PyCharm asks you to import settings of the existing package if any.

- **Arduino IDE**



Figure.1.3 Arduino IDE

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

STEPS OF INSTALLATION:

STEP 1 – First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer.

STEP 2 – DOWNLOAD ARDUINO IDE SOFTWARE.

You can get different versions of Arduino IDE from the [Download page](#) on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

STEP 3 – POWER UP YOUR BOARD.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labelled PWR) should glow.

STEP 4 – LAUNCH ARDUINO IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

STEP 5 – OPEN YOUR FIRST PROJECT.

Once the software starts, you have two options –

- Create a new project.
- Open an existing project example.

STEP 6 – SELECT YOUR ARDUINO BOARD.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools → Board and select your board.

STEP 7 – SELECT YOUR SERIAL PORT.

Select the serial device of the Arduino board. Go to **Tools** → **Serial Port** menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.

STEP 8 – UPLOAD THE PROGRAM TO YOUR BOARD.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Technologies used

- | | |
|------------|------------------------|
| • Python | Back-end Technology |
| • HTML | Front-end Technology |
| • CSS | Front-end Technology |
| • Flask | Frame Work |
| • Firebase | Database to store data |

1.5.2 HARDWARE REQUIREMENTS

Processor : Intel® CORETM i3

RAM : 4GB and above

Hard Disk : 8GB and above

Sensors :

- **DHT11 SENSOR**

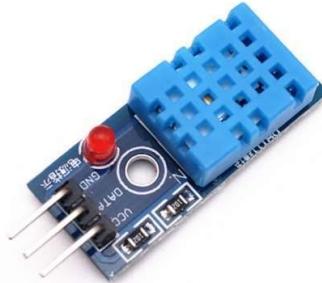


Figure.1.4 DHT11 Sensor

ABOUT DHT11 SENSOR :

The DHT11 sensor comes in a single row 4-pin package and operates from 3.5 to 5.5V power supply. It can measure temperature from 0-50 °C with an accuracy of $\pm 2^{\circ}\text{C}$ and relative humidity ranging from 20-95% with an accuracy of $\pm 5\%$. The sensor provides fully calibrated digital outputs for the two measurements. It has got its own proprietary 1-wire protocol, and therefore, the communication between the sensor and a microcontroller is not possible through a direct interface with any of its peripherals. The protocol must be implemented in the firmware of the MCU with precise timing required by the sensor.

The **DHT11** is a basic, ultra-low-cost digital temperature and humidity **sensor**. It uses a capacitive humidity **sensor** and a thermistor to measure the surrounding air, and spits out a digital signal on the data pin (no analog input pins needed).

This DHT11 Temperature and Humidity Sensor features a calibrated digital signal output with the temperature and humidity sensor complex. Its technology ensures the high reliability and excellent long-term stability. A high-performance 8-bit microcontroller is connected. This sensor includes a resistive element and a sense of wet NTC temperature measuring devices. It has excellent quality, fast response, anti-interference ability and high cost performance advantages.

Each DHT11 sensors features extremely accurate calibration of humidity calibration chamber. The calibration coefficients stored in the OTP program memory, internal sensors detect signals in the process, we should call these calibration coefficients. The single-wire serial interface system is integrated to become quick and easy. Small size, low power, signal transmission distance up to 20 meters, making it a variety of applications and even the most demanding applications. The product is 4-pin single row pin package. Convenient connection, special packages can be provided according to users need.

- **SOIL MOISTURE SENSOR**

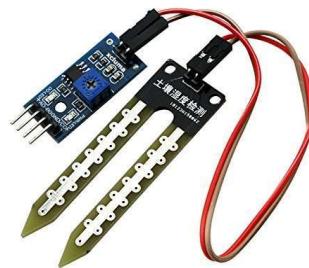


Figure.1.5 Soil moisture Sensor

The Soil Moisture Sensor is used to measure the volumetric **water** content of soil. This makes it ideal for performing **experiments** in courses such as soil science, agricultural science, environmental science, horticulture, botany, and biology.

The Soil Moisture Sensor uses capacitance to measure the water content of soil (by measuring the dielectric permittivity of the soil, which is a function of the water content). Simply insert this rugged sensor into the soil to be tested, and the volumetric water content of the soil is reported in percent.

Specifications

- Range: 0 to 45% volumetric water content in soil (capable of 0 to 100% VWC with alternate calibration)
- Accuracy: $\pm 4\%$ typical
- Typical Resolution: 0.1%
- Power: 3 mA @ 5VDC
- Operating temperature: -40°C to $+60^{\circ}\text{C}$
- Dimensions: 8.9 cm \times 1.8 cm \times 0.7 cm (active sensor length 5 cm)

Board :

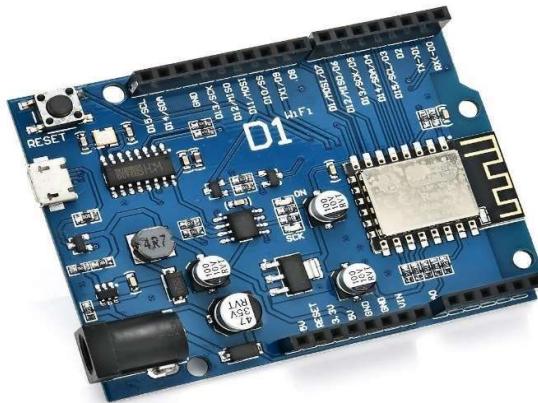


Figure. 1.6 Arduino UNO – Wi-Fi Module

The **Arduino Uno Wi-Fi** is an **Arduino Uno** with an integrated **Wi-Fi module**. The **ESP8266 WiFi Module** is a self-contained SoC with integrated TCP/IP protocol stack that can give access to your **Wi-Fi** network (or the device can act as an access point).

Wi-Fi ESP8266 Development Board WEMOS D1. WEMOS D1 is a WIFI development board based on ESP8266 12E. The functioning is similar to that of NODEMCU, except that the hardware is built resembling Arduino UNO. The D1 board can be configured to work on Arduino environment using BOARDS MANAGER.

Specification:

- Microcontroller: ESP-8266EX
- Operating Voltage: 3.3V
- Digital I/O Pins: 11
- Analog Input Pins: 1
- Clock Speed: 80MHz/160MHz
- Flash: 4M bytes

The table below shows the pins of the wemos d1, the main difference between this and the Arduino UNO is that there is only one Analog input. All I/O pins are 3.3v, so some Arduino shields will not work and obviously be careful what you connect to these pins.

Pin	Function	ESP-8266 Pin
D0	RX	GPIO3
D1	TX	GPIO1
D2	IO	GPIO16
D3(D15)	IO,SCL	GPIO5
D4(D14)	IO,SDA	GPIO4
D5(D13)	IO,SCK	GPIO14
D6(D12)	IO,MISO	GPIO12
D7(D11)	IO,MOSI	GPIO13
D8	IO, Pull-up	GPIO0
D9	IO, pull-up, BUILTIN_LED	GPIO2
D10	IO,pull-down,SS	GPIO15
A0	Analog Input	A0

Table 1.1 : Pin Configuration

Devices :



Figure. 1.7 Jumper Wires

A **jump wire** (also known as jumper, jumper wire, jumper cable, DuPont wire or cable) is an , or group electrical wire of them in a cable, with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.

Individual jump wires are fitted by inserting their "end connectors" into the slots provided in a breadboard, the header connector of a circuit board, or a piece of test equipment.



Figure. 1.8 USB cable

The USB cable has one end as USB type A connector and another end has a USB type B connector. The USB type-A end is connected to the power source or the programming device (e.g., your computer/laptop) and the USB B end is connected to Arduino UNO or MEGA 2560 board.

- Use it to connect Arduino Uno, Arduino Mega 2560, or any board with the USB female A port of your computer.
- Cable color and shape may vary slightly from an image as our stock rotates.
- USB A to B (Male to Male)

CHAPTER 2

LITERATURE SURVEY

[1] 2019 Global IoT Summit (GloTS) by- Noriyoshi Anbai “Secure smart agriculture monitoring technique through isolation”

A gate management system for controlling an operation of agate of a vault including a biometrics information management device having a first memory, The current work provides empirical measurement results from an Internet of Things (IoT) platform based on remote telemetry applications for agriculture. This paper aims to demonstrate how the processing of Big Data and the concept of decentralized cloud operation can answer the demands of IoT applications in agriculture and how smart farming will help farmers operate more efficiently and more secured. The architecture is based on ADCON Remote Telemetry Units (RTUs), Software as a Service (SaaS), a Platform as a Service (PaaS) represented by Grafana and influxDB database.

[2] 2016 (ICGTSPICC) by- K. A. Patil “A model for smart agriculture using IoT”
smart agriculture is an automated and directed information technology implemented with the IOT (Internet of Things). IOT is developing rapidly and widely applied in all wireless environments. In this paper, sensor technology and wireless networks integration of IOT technology has been studied and reviewed based on the actual situation of agricultural system. A combined approach with internet and wireless communications, Remote Monitoring System (RMS) is proposed. Major objective is to collect real time data of agriculture production environment that provides easy access for agricultural facilities such as alerts through Short Messaging Service (SMS) and advices on weather pattern, crops etc.

[3] 2018 (ICIRCA) by Rahul Dagar ; Subhranil Som ; Sunil Kumar Khatri “Smart Farming – IoT in Agriculture” This paper is about the implementation of IoT in Agriculture. IoT helps in better crop management, better resource management, cost efficient agriculture, improved quality and quantity, crop monitoring and field monitoring etc. can be done. The IoT sensors used in proposed model are air temperature sensor, soil pH sensor, soil moisture sensor, humidity sensor, water volume sensor etc. In this paper I surveyed typical agriculture methods used by farmers these days and what are the problems they face, I visited poly houses for further more information about new technologies in farming. The proposed model is a simple architecture of IoT sensors that collect information and send it over the Wi-Fi network to the server, there server can take actions depending on the information.

[4] 2018 (WiSPNET) G. Sushanth ; S. Sujatha “IOT Based Smart Agriculture System”

The paper aims at making use of evolving technology i.e. IOT and smart agriculture using automation. Monitoring environmental conditions is the major factor to improve yield of the efficient crops. The feature of this paper includes development of a system which can monitor temperature, humidity, moisture and even the movement of animals which may destroy the crops in agricultural field through sensors using Arduino board and in case of any discrepancy send a SMS notification as well as a notification on the application developed for the same to the farmer's smartphone using Wi-Fi/3G/4G. The system has a duplex communication link based on a cellular-Internet interface that allows for data inspection and irrigation scheduling to be programmed through an android application.

S.No	Name of Author	Title of Paper	Year	Description
[1]	George Suciu	Secure smart agriculture monitoring technique through isolation	June 2019	The architecture is based on ADCON Remote Telemetry Units (RTUs), Software as a Service (SaaS), a Platform as a Service (PaaS) represented by Grafana and influxDB database.
[2]	K. A. Patil ; N. R. Kale	A model for smart agriculture using IoT	June 2017	In This project, sensor technology and wireless networks integration of IOT technology has been studied and reviewed based on the actual situation of agricultural system. A combined approach with internet and wireless communications, Remote Monitoring System (RMS) is proposed.

An IoT and Machine Learning based model for smart agriculture monitoring and alerting

[3]	Rahul Dagar	Smart Farming – IoT in Agriculture	Jan 2019	The proposed model is a simple architecture of IoT sensors that collect information and send it over the Wi-Fi network to the server, there server can take actions depending on the information.
[4]	G. Sushanth	IOT Based Smart Agriculture System	Nov 2018	The feature of this paper includes development of a system which can monitor temperature, humidity, moisture and even the movement of animals which may destroy the crops in agricultural field through sensors using Arduino board and in case of any discrepancy send a SMS notification.

Table 2.1. Paper Articles

CHAPTER 3

METHODOLOGY

3.1 SYSTEM ARCHITECTURE

3.1.1 FLASK FRAMEWORK

Flask is a web framework. This means flask provides tools, libraries, and technologies that allows to build a web application. This web application can be some web pages, a blog, a wiki or go as big as a web-based calendar application or a commercial website.

Flask is part of the categories of the micro-framework. Micro-framework is normally a framework with little to no dependencies to external libraries. This has pros and cons. Pros would be that the framework is light, there is little dependency to update and watch for security bugs, cons is that sometime you will have to do more work by yourself or increase yourself the list of dependencies by adding plugins.

In the case of Flask, its dependencies are:

- Werkzeug a WSGI utility library
- jinja2 which is its template engine

3.1.2 FIREBASE CLOUD STORAGE

Cloud Storage for Firebase is a powerful, simple, and cost-effective object storage service built for Google scale. The Firebase SDKs for Cloud Storage add Google security to file uploads and downloads for Firebase apps, regardless of network quality. We can use our SDKs to store images, audio, video, or other user-generated content. On the server, we can use [Google Cloud Storage](#), to access the same files.

Firebase SDKs for Cloud Storage perform uploads and downloads regardless of network quality. Uploads and downloads are robust, meaning they restart where they stopped, saving users time and bandwidth.

Firebase SDKs for Cloud Storage integrate with Firebase Authentication to provide simple and intuitive authentication for developers. One can use its declarative security model to allow access based on filename, size, content type, and other metadata.

Key Features

- Cloud Fire store**: It is a flexible, scalable database for mobile, web, and server development from Firebase and Google Cloud Platform.
- The Firebase Realtime Database**: It stores JSON application data, like game state or chat messages, and synchronizes changes instantly across all connected devices.
- Firebase Remote Config**: It stores developer-specified key-value pairs to change the behavior and appearance of app without requiring users to download an update.

3.1.3 DECISION TREE ALGORITHM:

INTRODUCTION

Classification is a two-step process, learning step and prediction step, in machine learning. In the learning step, the model is developed based on given training data. In the prediction step, the model is used to predict the response for given data. Decision Tree is one of the easiest and popular classification algorithms to understand and interpret.

DECISION TREE ALGORITHM

Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, the decision tree algorithm can be used for solving **regression and classification problems** too.

The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by **learning simple decision rules** inferred from prior data(training data).

In Decision Trees, for predicting a class label for a record we start from the **root** of the tree. We compare the values of the root attribute with the record's attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node.

Important Terminology related to Decision Trees:

- Root Node**: It represents the entire population or sample and this further gets divided into two or more homogeneous sets.
- Splitting**: It is a process of dividing a node into two or more sub-nodes.
- Decision Node**: When a sub-node splits into further sub-nodes, then it is called the decision node.
- Leaf / Terminal Node**: Nodes do not split is called Leaf or Terminal node.

5. **Pruning:** When we remove sub-nodes of a decision node, this process is called pruning.
You can say the opposite process of splitting.
6. **Branch / Sub-Tree:** A subsection of the entire tree is called branch or sub-tree.
7. **Parent and Child Node:** A node, which is divided into sub-nodes is called a parent node of sub-nodes whereas sub-nodes are the child of a parent node.

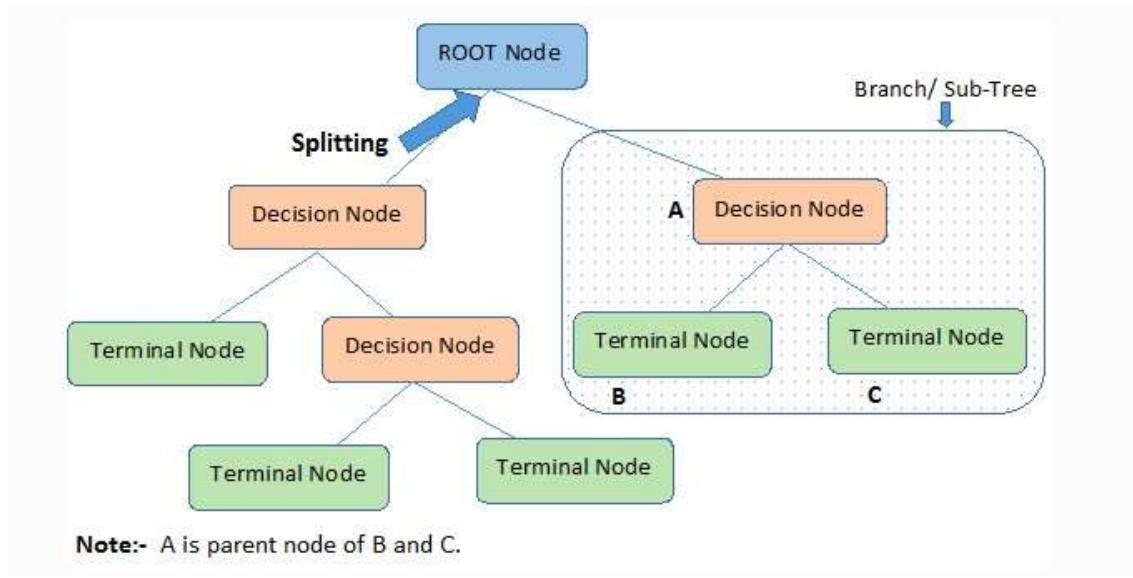


Figure 3.1 : Decision Tree Algorithm

How do Decision Trees work?

The decision of making strategic splits heavily affects a tree's accuracy. The decision criteria are different for classification and regression trees.

Decision trees use multiple algorithms to decide to split a node into two or more sub-nodes. The creation of sub-nodes increases the homogeneity of resultant sub-nodes. In other words, we can say that the purity of the node increases with respect to the target variable. The decision tree splits the nodes on all available variables and then selects the split which results in most homogeneous sub-nodes.

The algorithm selection is also based on the type of target variables. Let us look at some algorithms used in Decision Trees:

ID3 → (extension of D3)

C4.5 → (successor of ID3)

CART → (Classification And Regression Tree)

CHAID → (Chi-square automatic interaction detection Performs multi-level splits when computing classification trees)

MARS → (multivariate adaptive regression splines)

The ID3 algorithm builds decision trees using a top-down greedy search approach through the space of possible branches with no backtracking. A greedy algorithm, as the name suggests, always makes the choice that seems to be the best at that moment.

3.2 SOFTWARE ARCHITECTURE

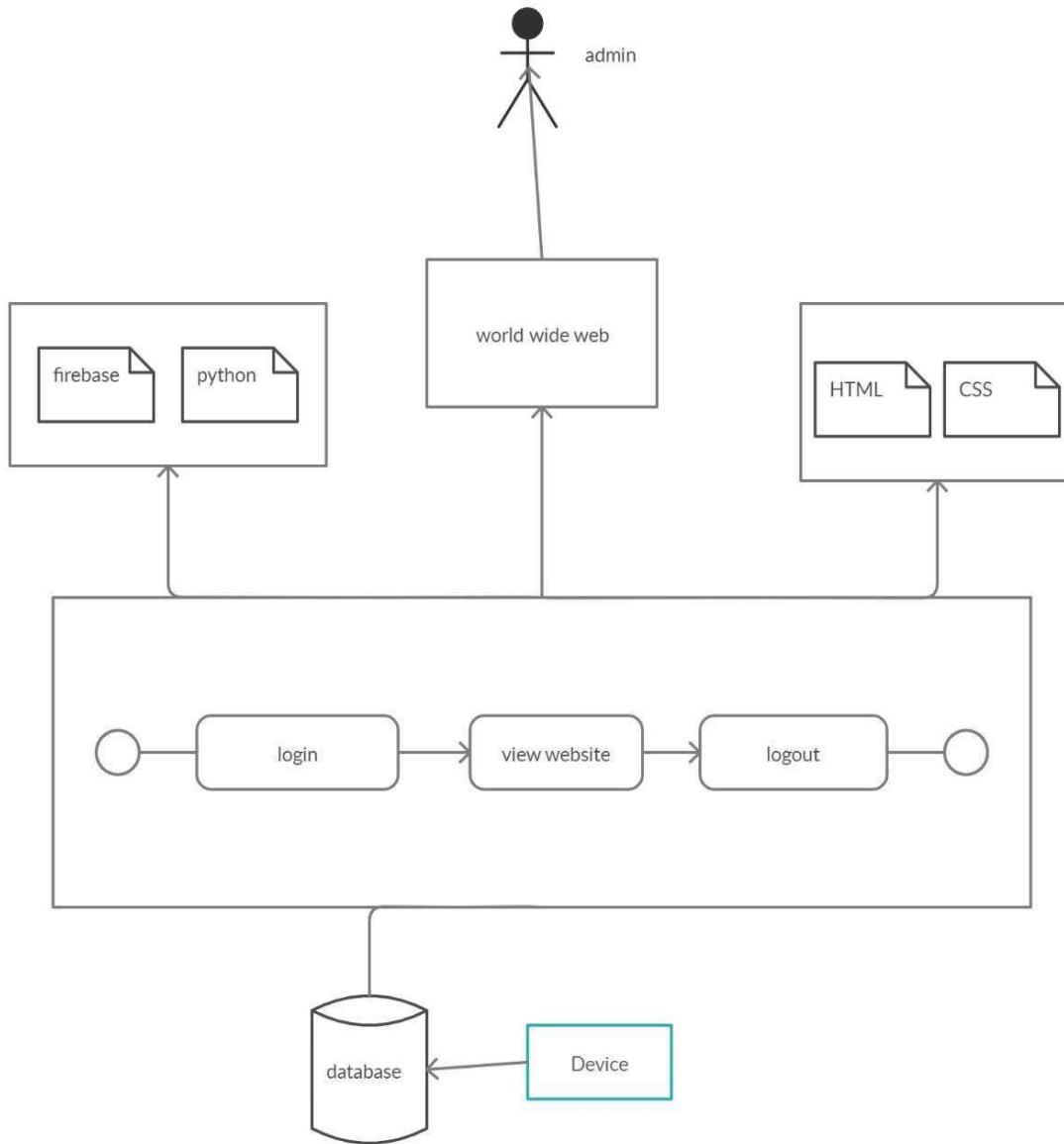


Figure.3.2 Software Architecture of system

In the above figure 3.2, the administrator communicates with the help of World Wide Web(WWW). He does the entire activity of managing users and device information by using Firebase cloud storage. The backend is done by using Flask. The front end is developed by using HTML, and CSS.

3.3 FLOWCHART SHOWING THE WORKING

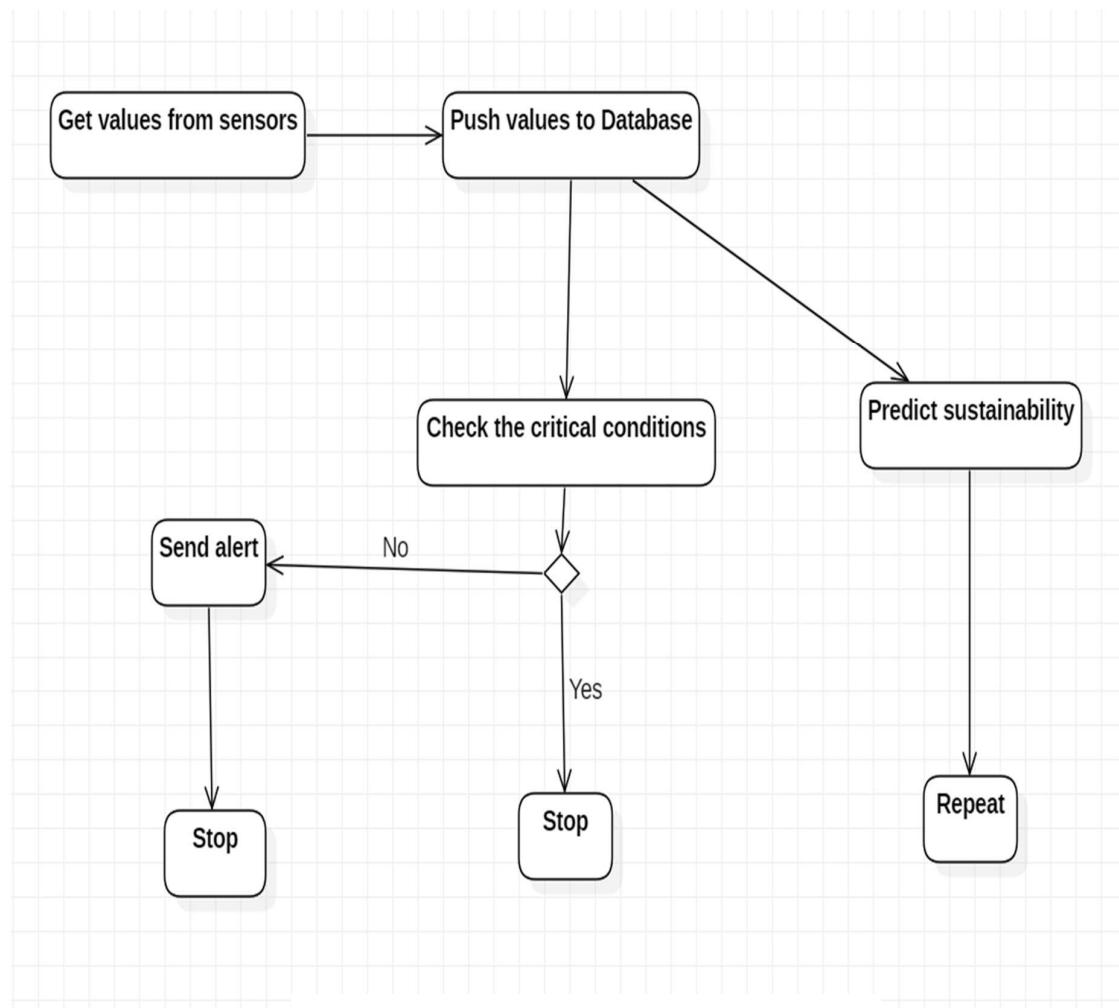


Figure.3.3 Flowchart Showing the working

In the above figure 3.3 , initially the device collects all the data from the sensors and push it to the database. Controller checks for the critical conditions and if the conditions are not suitable the controller sends an email alert .The collected data is also used for predicting crop sustainability

3.4 DIAGRAMMATIC REPRESENTATIONS

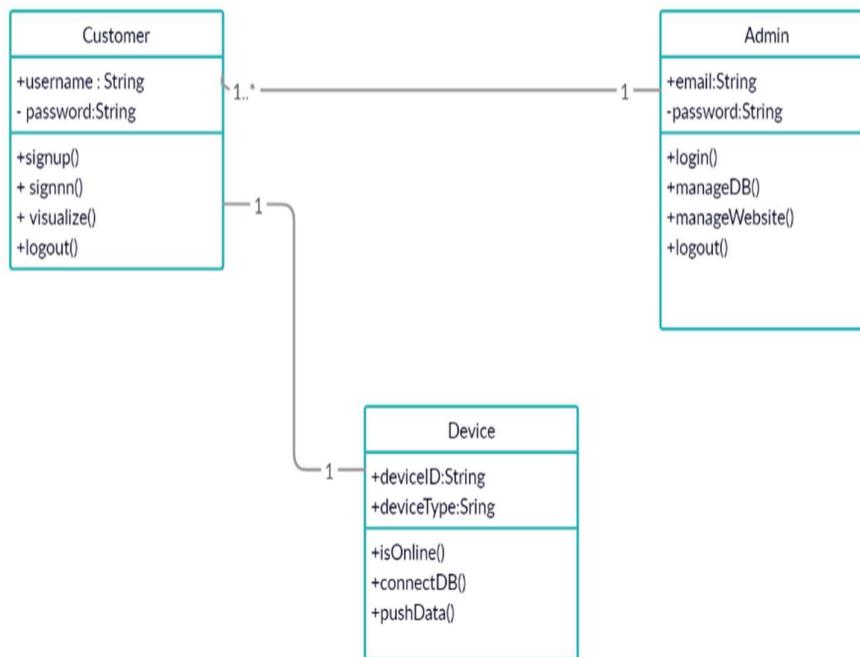


Figure.3.4 Representation of classes along with attributes and classes

In the above figure 3.4, the class diagram consists of three classes namely Admin, Customer and Device

The attributes and operations of each class are as follows:

Admin:-

Attributes: email, password

Operations: login(), manageDB(), manageWebsite()

Customer:-

Attributes: username, password

Operations: signin(), signup(), visualize(), logout()

Device:-

Attributes: SensorId, Timestamp, DataValue

Operations: pushdata(DataValue), isonline()

There is a single Admin for many Users and a single User for many Sensors.

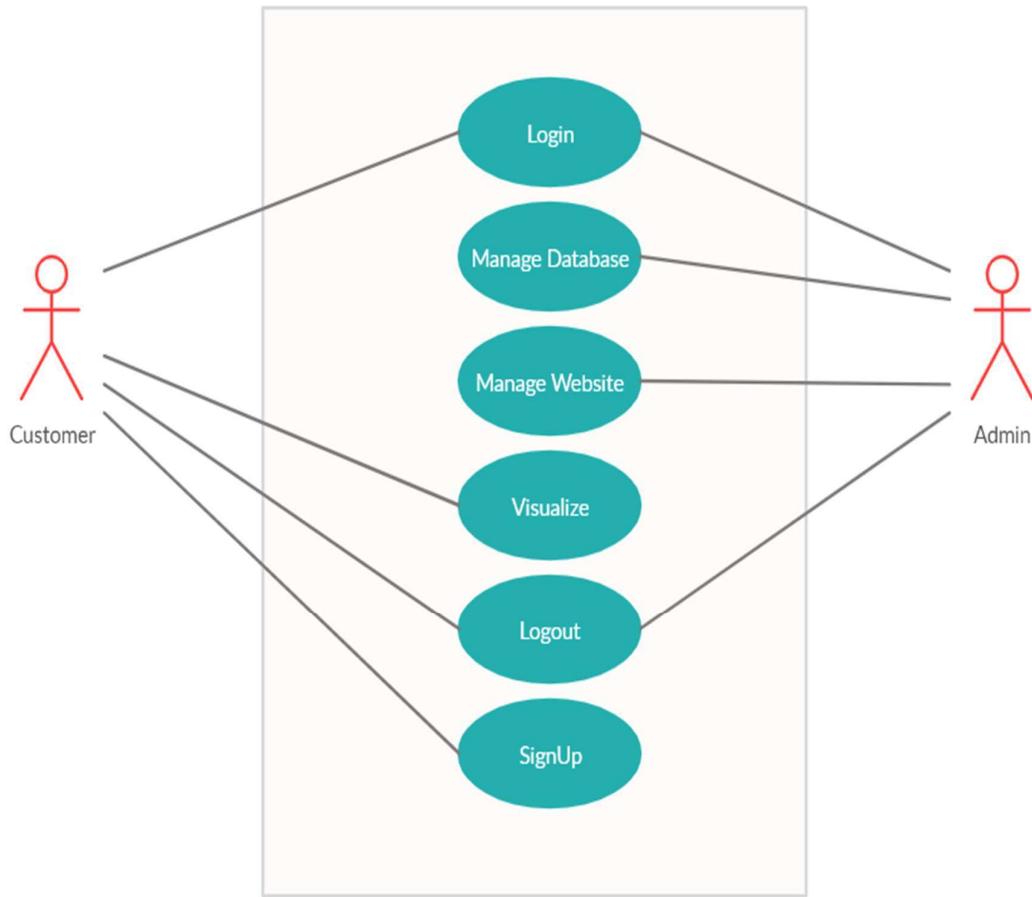


Figure.3.5 Various use cases in the proposed system

In the above figure 3.5, Use case diagram consists of two actors. They are Customer and Admin . Customer will login or Signup if he doesn't have an account. Customer can visualize all the graphs and finally he will Logout. Admin will login and he manages the database and also the website. The various use cases are as follows:-

- Login-** admin and customer should login to view or mange the website
- Manage Database-** the database has maintained by admin.
- Manage website-** admin have to look up for website
- Visualize-** Customer can have the data in the form of graphs.
- Logout-** finally admin or user should logout.
- Sign-up-** if user do not have an account he can create using signup

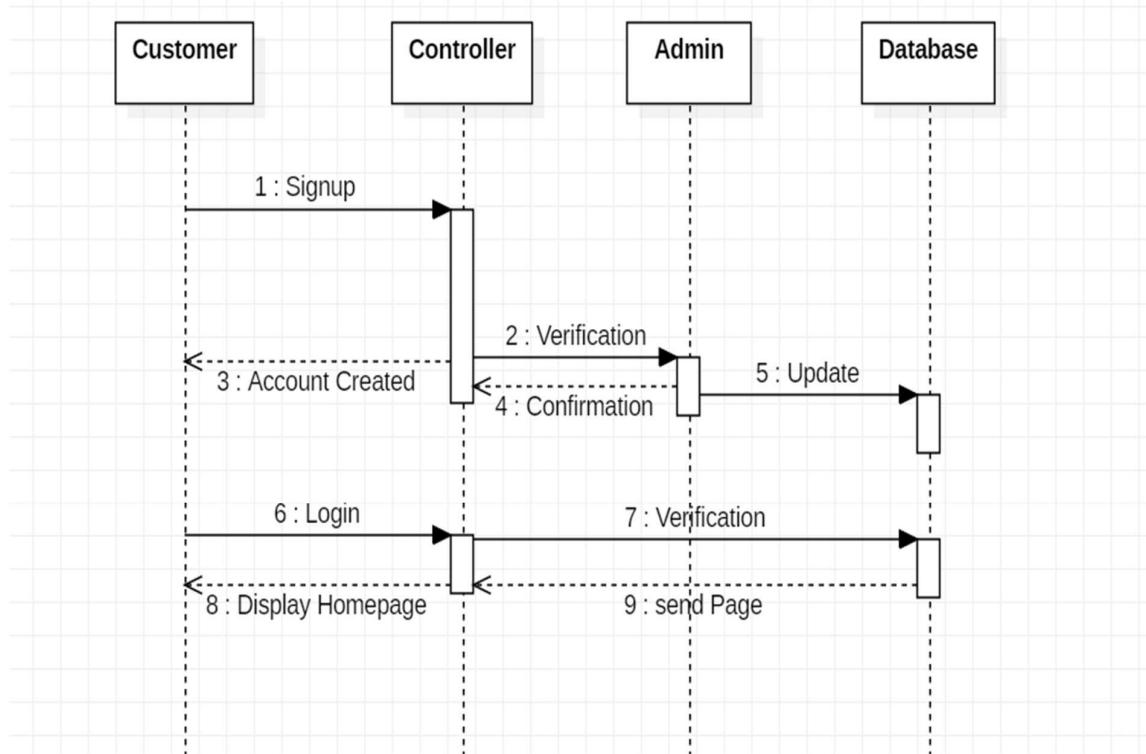


Figure.3.6 Flow of Operations

In the above figure 3.6, the sequence diagram defines the flow of operations. The behavioral execution states are:

1. Signup
2. Verification
3. Account created
4. Confirmation
5. Update
6. Login
7. Verification
8. Display Homepage

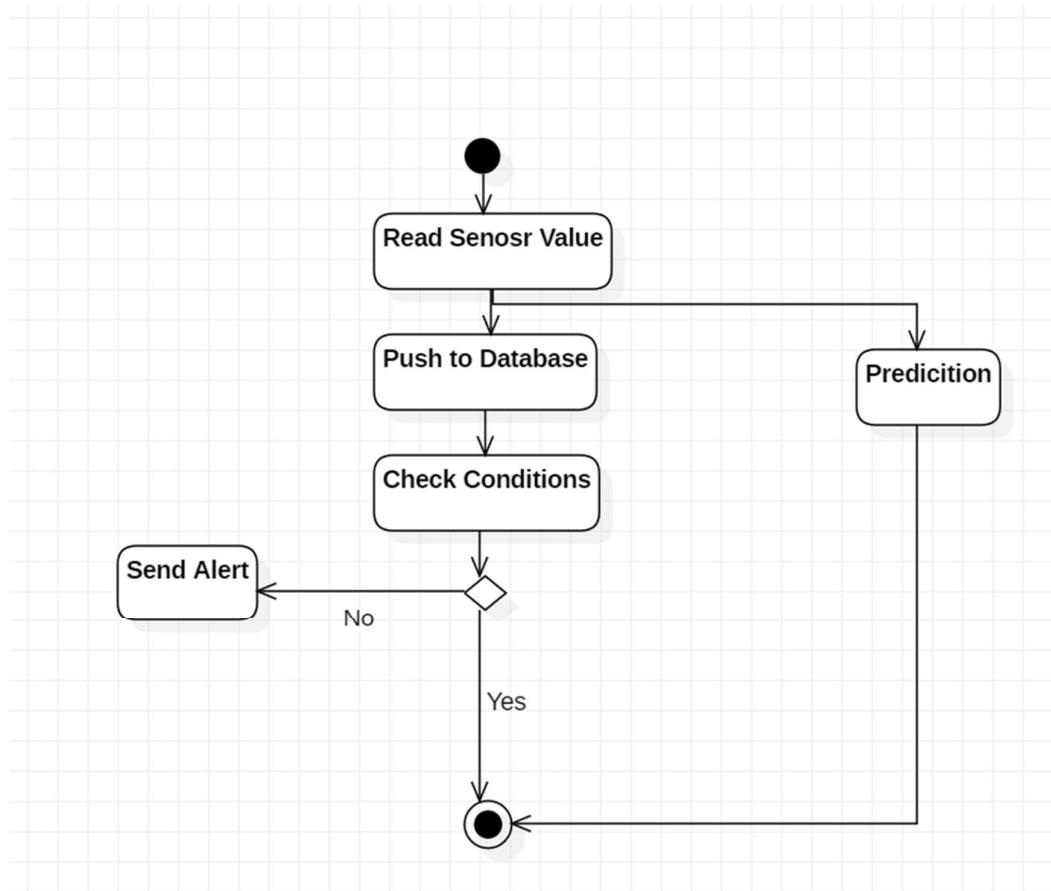


Figure.3.7 Illustration of graphical workflow

In the above figure 3.7, Activity Diagram represents workflows in a graphical way. This diagram below describes the business overflow on the operational workflow of any component in a system.

The activity states in this diagram are:-

1. Read Sensor Value
2. Push to Database
3. Send the alert
4. Prediction

3.5 FEATURES

There are various features in this project such as:

- 3.5.1 The sensor value is uploaded every 3 seconds

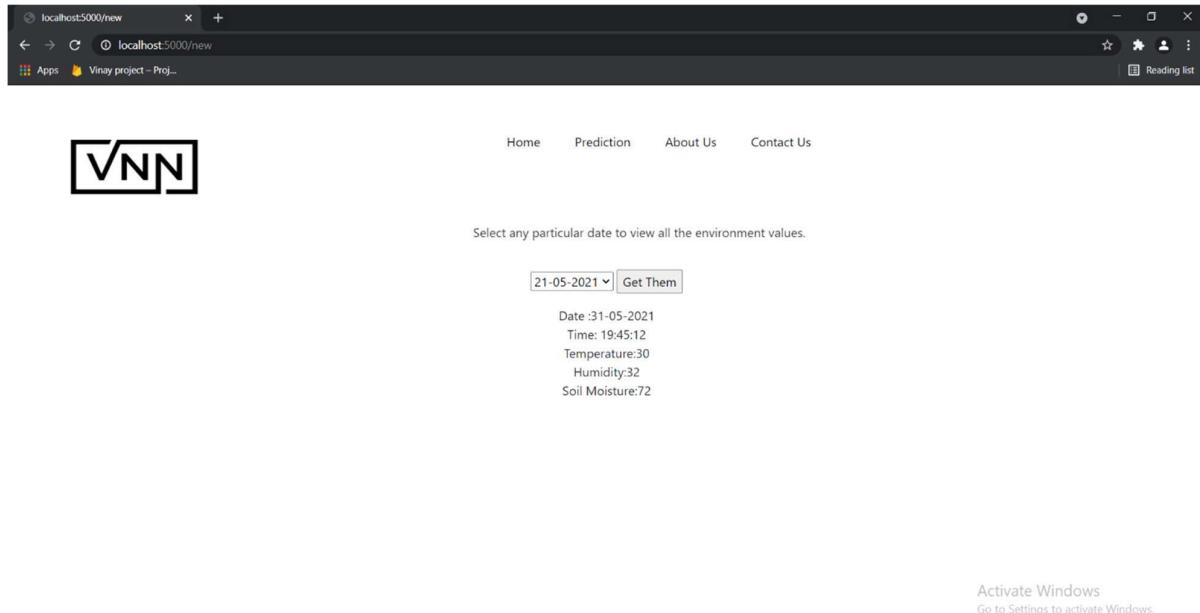


Figure.3.8 Storing of data with 3 seconds difference

The sensor will be requested for sensor value for every 3 seconds and that sensor value will be uploaded to the database for every 3 seconds for precise monitoring.

3.5.2 Mail alert will be sent if the values exceed optimal level

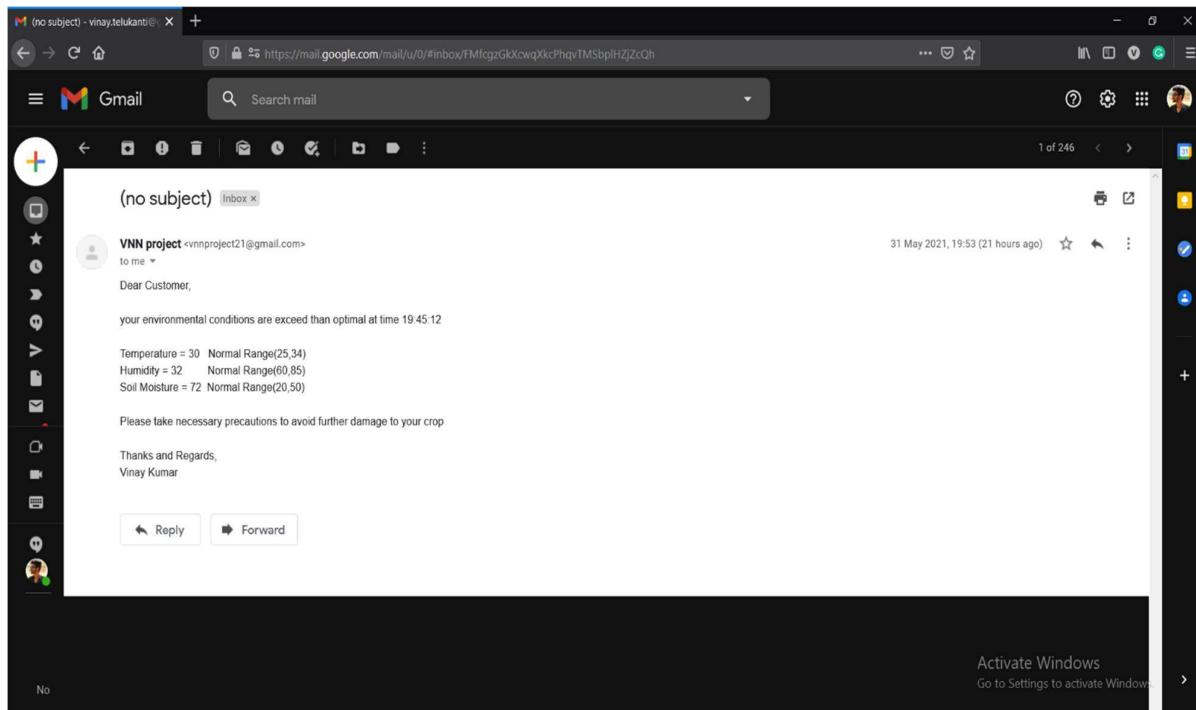


Figure.3.9 Mail Alert

In the above figure 3.9, The sensor values are monitored and if the values which are measured recently, crosses the considerable range then a mail is sent that temperature is crossed the limits.

CHAPTER 4

TESTING AND RESULTS

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. In fact, testing is the one step in the software engineering process that could be viewed as destructive rather than constructive.

A strategy for software testing integrates software test case design methods into a well-planned series of steps that result in the successful construction of software. Testing is the set of activities that can be planned in advance and conducted systematically. The underlying motivation of program testing is to affirm software quality with methods that can economically and effectively apply to both strategic to both large and small-scale systems.

Test Case	Test Cases	Description	Input	Expected Output	Actual Output	Result
Id						
Logi n_1	Empty Field	Empty Field	No Data	Please fill out this detail	Please fill out this detail	Pass
ogin _2	Enteri ng	Userna me, invalid	Invalid User- Id, Passwo rd	Oops! Invalid Credenti als!	Oops! Invalid Crede ntials!	Pass
Logi n_3	Enteri ng	Userna me, Valid	User- Id, Passwo rd	Navigates to Values page	Navigat es to Values page	Pass
		Userna me and Passw ord				

Ad	Entering	E-	Invalid	Invalid	Invalid	Pass
mi	invalid	mail	E-mail	Admin	Admin	
n	credenti	ID	ID	Credentials	Credentia	ls

Table 4.1. Test cases for application

4.1.1 UNIT TESTING

A unit is the smallest testable part of software. It usually has one or a few inputs and usually a single output. In procedural programming a unit may be an individual program, function, procedure, etc. In our project individual modules of our application so that bugs can be removed.

4.1.2 SYSTEM TESTING

System Testing is a level of the software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements. In our application whole system was tested and appropriate testing methods were applied in order to remove bugs.

4.1.3 TEST STRATEGY AND APPROACH

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

4.1.4 TEST OBJECTIVES

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

4.1.5 FEATURES TO BE TESTED

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

4.2 OUTPUT RESULTS

4.2.1 LANDING PAGE

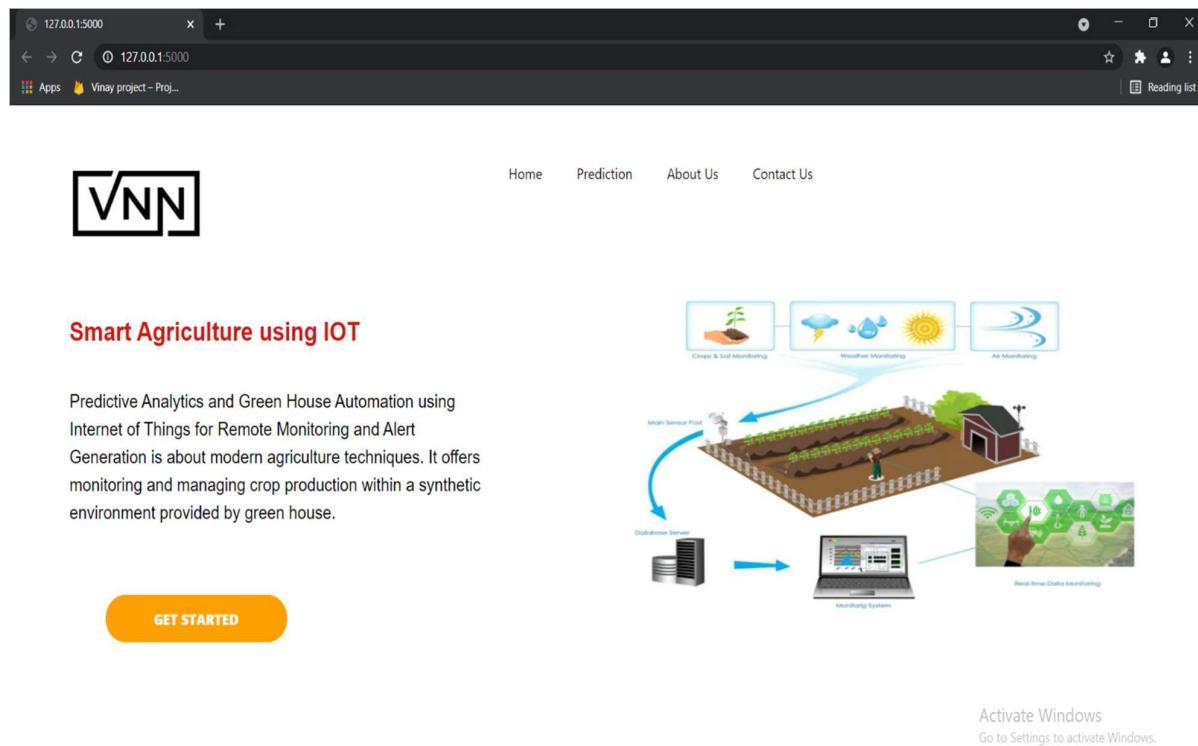


Figure.4.1 Landing Page

In above figure 4.1, When the website is launched landing page will appear.

After clicking Get Started button , login page will be appeared.

4.2.2 LOGIN PAGE

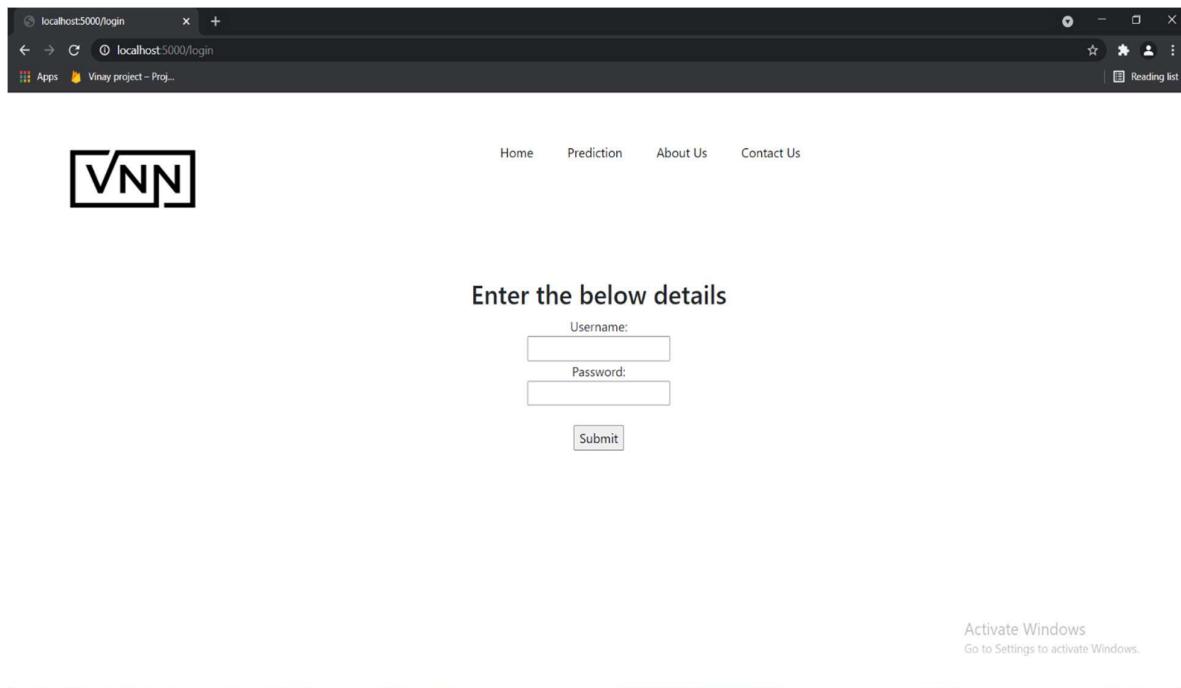


Figure.4.2 Login Page

In above figure.4.2, When the user clicks get started button the login page appears where the User have to login with his credentials to access all other features

If the User username and password are correct and matched with the ones in the database then the web page navigates user to the Main Page.

If the two fields i.e. username and password are left empty then an error message, “Login Failed! Username or password is not entered” is raised. If the details are incorrect then an error message, “Invalid Admin credentials” is raised.

4.2.3 Main Page

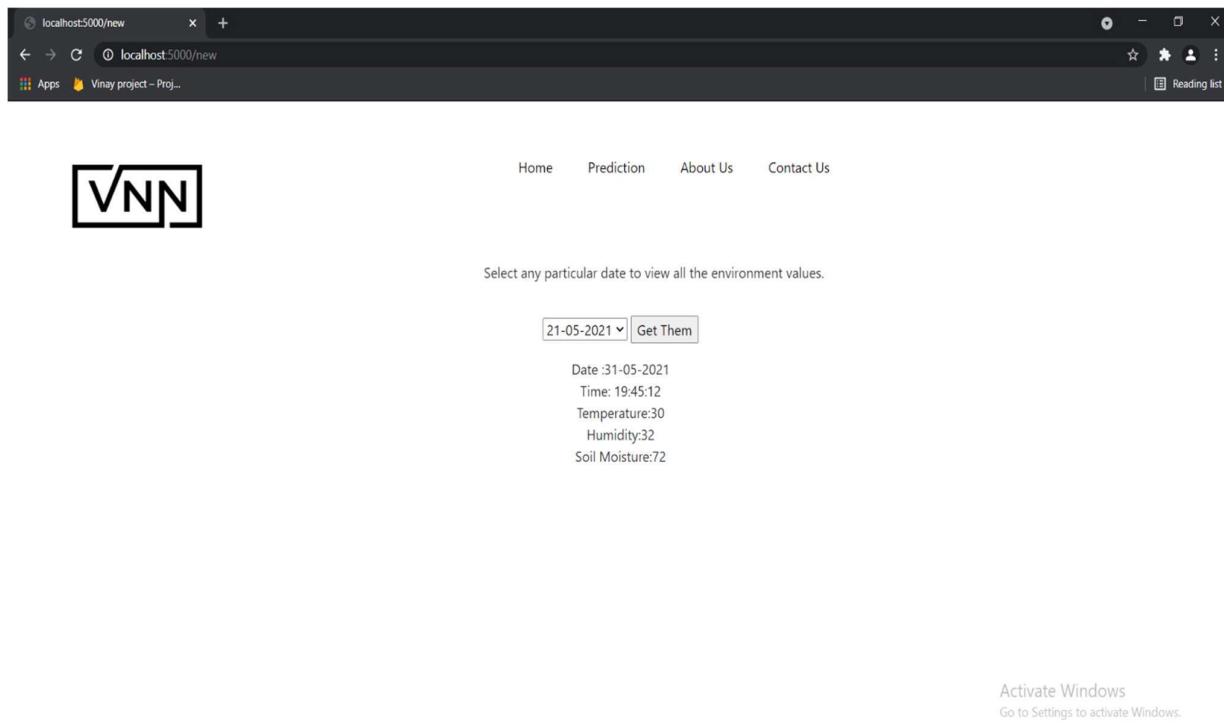


Figure.4.3 Get values page

In the above figure.4.3, if the user credentials are matched, then the web page is navigated to this main page.

This main page is unique to each user where he can find all the values of the sensors which has been setup in his field.

The user has to select a particular date to view the values of the sensors of that particular date.

4.2.4 Mailing

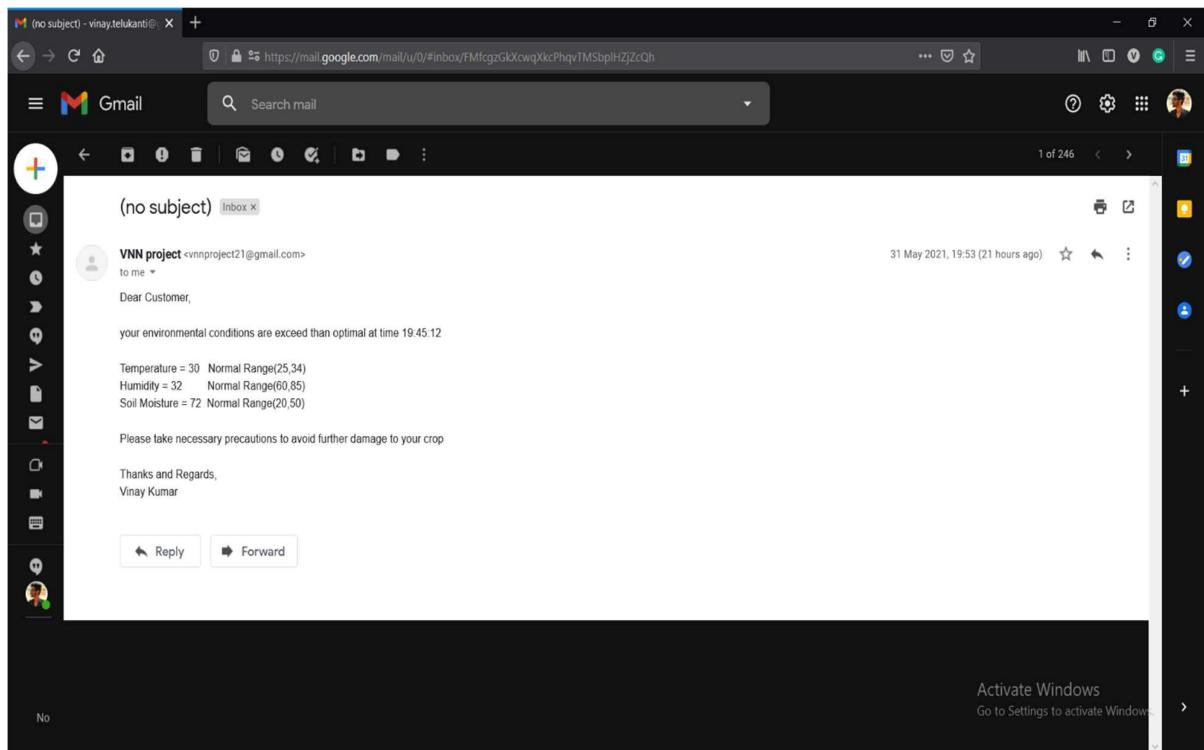


Figure.4.4 Mail values crosses the limit

In the above figure.4.4, The values are monitored and if the values which are measured recently, crosses the considerable range then a mail is sent those values have crossed the limits.

The screenshot shows a web browser window titled "localhost:5000/subscribe". The page has a header with navigation links: Home, Prediction, About Us, and Contact Us. On the left is a logo consisting of a square containing the letters "VNN". The main content area contains the text "Enter the below details" followed by three input fields: "Temperature:" with value "30", "Humidity:" with value "32", and "SoilMoisture:" with value "72". Below these fields is a "Submit" button. At the bottom of the page, the text "Output : Difficult to Sustain" is displayed.

Figure.4.5 Prediction Page

The screenshot shows a web browser window titled "localhost:5000/aboutus". The page has a header with navigation links: Home, Prediction, About Us, and Contact Us. On the left is a logo consisting of a square containing the letters "VNN". The main content area contains several paragraphs of text. One paragraph discusses the history and importance of agriculture. Another paragraph describes the proposed IOT and smart agriculture system, mentioning the use of sensors and wireless protocols. A third paragraph details the system's functionality, including temperature, humidity, and moisture monitoring, and the sending of email alerts. A fourth paragraph notes the system's predictive capabilities and data mining potential. A fifth paragraph discusses the cost-benefit ratio. At the bottom right of the page, there is a watermark: "Activate Windows Go to Settings to activate Windows."

Figure.4.6 About us page

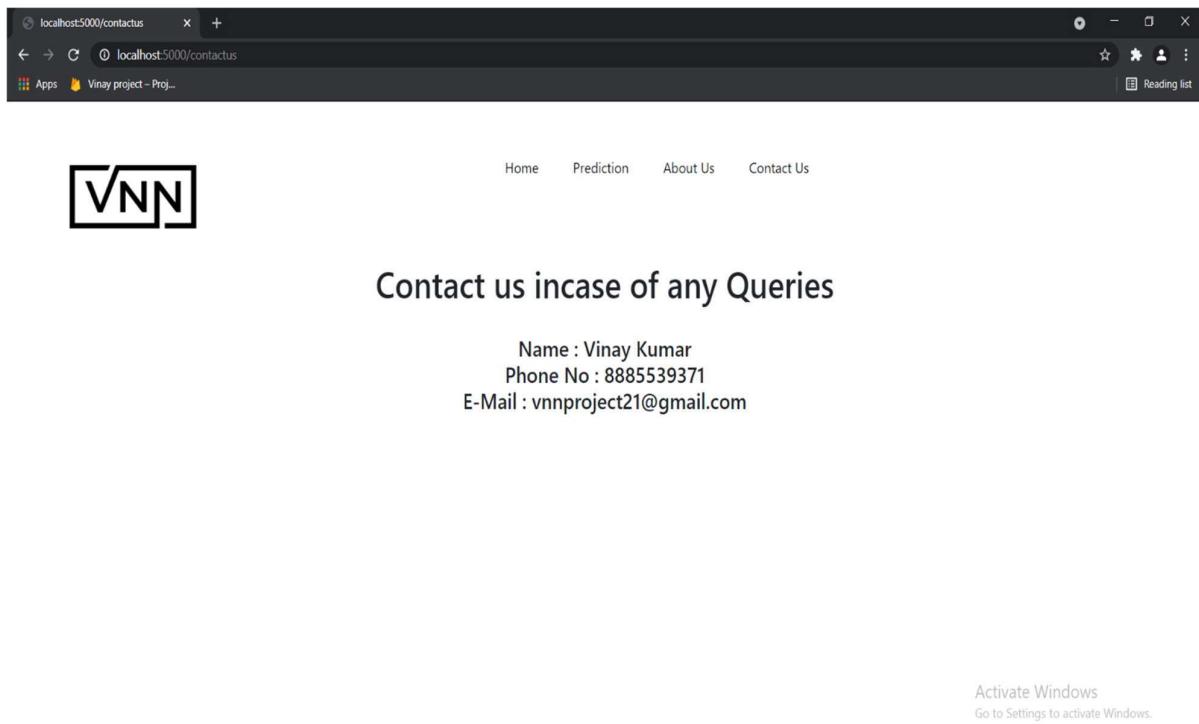


Figure.4.7 Contact us page

CHAPTER 5

FUTURE SCOPE

Future enhancement can be done in the proposed system by completely automating the system. This can be done by controlling different factors which affect the growth of crops such as Ph level of soil, temperature, humidity, soil moisture and CO₂ level inside the greenhouse. The values of these factors which are already being monitored by using sensors in our system can be compared to the required threshold conditions and can be automated to take the respective steps to maintain the threshold conditions. And furthermore, features like crop disease detection (using image processing techniques), crop rotation techniques can be included.

CONCLUSION

The ultimate goal of this project is to increase crop growth along with maximized yield, irrespective of the weather conditions.

Advantages:

- To increase agriculture productivity
- Prevents soil degradation
- Efficient use of water resources
- Dissemination of modern farm practices to improve quality, quantity and reduced cost of production
- Developing favorable attitudes
- Changes the socio-economic status of farmers.

BIBLIOGRAPHY

- [1] 2019 Global IoT Summit (GIoTS) by- Noriyoshi Anbai “Secure smart agriculture monitoring technique through isolation”
- [2] 2016 (ICGTSPICC) by- K. A. Patil “A model for smart agriculture using IoT”
- [3] 2018 (ICIRCA) by Rahul Dagar ; Subhranil Som ; Sunil Kumar Khatri “Smart Farming – IoT in Agriculture”[
- [4] 2018 (WiSPNET) G. Sushanth ; S. Sujatha “IOT Based Smart Agriculture System”[5]<https://www.projectsof8051.com/iot-based-greenhouse-monitoring-and-control-system-project/>
- [5] <https://www.projectsof8051.com/iot-based-greenhouse-monitoring-and-control-system-project/>
- [6] <https://krishijagran.com/agripedia/what-are-the-benefits-of-polyhouse-cultivation/>
- [7] M. Ayaz, M. Ammad-Uddin, Z. Sharif, A. Mansour and E. -H. M. Aggoune, "Internet-of-Things (IoT)-Based Smart Agriculture: Toward Making the Fields Talk," in IEEE Access, vol. 7, pp. 129551-129583, 2019, : 10.1109/ACCESS.2019.2932609.
- [8] December 2019 [**International Journal of Information Dissemination and Technology**](#) Smart Agriculture Using IOT: A Futuristic Approach
- [9] Agrawal, Kushagra 2019/12/30 Smart Agriculture Using IOT: A Futuristic Approach International Journal of Information Dissemination and Technology

CHAPTER 6

APPENDIX

/* Code for Arduino */

```
#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>
#include<dht.h>
#include<time.h>
dht DHT;

#define FIREBASE_HOST " vinay-project-5cbeb-default-rtdb.firebaseio.com "
#define FIREBASE_AUTH"416CFFc2VuVmKF6f5fLPMPOTkg0kj98ixzgM2e4d
#define WIFI_SSID "VINAY"
#define WIFI_PASSWORD "vinay@1999"

int timezone = 19800;
int dst = 0;

void setup() {
  Serial.begin(9600);
  pinMode(A0,INPUT);
  // connect to wifi.
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("connecting");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("connected: ");
  Serial.println(WiFi.localIP());

  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
  configTime(timezone, dst, "pool.ntp.org","time.nist.gov");
  Serial.println("\n Waiting for Internet time");

  while(!time(nullptr)){
    Serial.print("*");
    delay(1000);
  }

  Serial.println("Time response ....OK");

}

int n = 0;
```

```

void loop() {
    if(n!=0){                                // if n==0 if prints 1970 date

        time_t now = time(nullptr);           // Date Time
        struct tm* p_tm = localtime(&now);
        int x1=p_tm->tm_mday;
        int y1=p_tm->tm_mon + 1;
        int z1=p_tm->tm_year + 1900;
        String s1=String(x1)+"-"+String(y1)+"-"+String(z1);
        int x2=p_tm->tm_hour;
        int y2=p_tm->tm_min;
        int z2=p_tm->tm_sec;
        String s2=String(x2)+":"+String(y2)+":"+String(z2);
        String s3=s1+" "+s2;
        Serial.println(s3);

        int x=DHT.read11(4);                  // DHT
        Serial.println("temperature");
        Serial.println(DHT.temperature,1);
        Serial.println("humidity");

        Serial.println(DHT.humidity,1);

        float moist;                         // Soil Moisture
        int anlread= analogRead(A0);
        moist=(100-((anlread/1024.00)*100));
        Serial.println("Moisture Percentage");
        Serial.println(moist);
        Serial.println();

        Firebase.pushFloat(s1+"/"+s2+"/Temperature", DHT.temperature);
        // handle error
        if(Firebase.failed()) {
            Serial.print("setting /number failed:");
            Serial.println(Firebase.error());
            return;
        }

        Firebase.pushFloat(s1+"/"+s2+"/Humidity", DHT.humidity);
        // handle error
        if(Firebase.failed()) {
            Serial.print("setting /number failed:");
            Serial.println(Firebase.error());
            return;
        }

        Firebase.pushFloat(s1+"/"+s2+"/SoilMoist", moist);
        // handle error
        if(Firebase.failed()) {
    }
}

```

An IoT and Machine Learning based model for smart agriculture monitoring and alerting

```
Serial.print("setting /number failed:");
Serial.println(Firebase.error());
return;
}

}
n=n+1;
delay(1000);

/* Code for backend */

from flask import Flask,request,redirect,url_for,render_template
from flask_mail import Mail, Message
import os, random, string
#from datetime import date
import time
import pandas as pd

from firebase import firebase
conn = firebase.FirebaseApplication("https://arduinodata-83659.firebaseio.com/")

app = Flask(__name__, static_url_path='/static')

app.config['MAIL_SERVER']='smtp.gmail.com'
app.config['MAIL_PORT'] = 465
app.config['MAIL_USERNAME'] = 'vnnproject21@gmail.com'
app.config['MAIL_PASSWORD'] = '*****'
app.config['MAIL_USE_TLS'] = False
app.config['MAIL_USE_SSL'] = True
mail = Mail(app)

#result = conn.get("/27-10-2019/18:23:12/", None)

@app.route('/')
def fun():
    return render_template('landingPage.html')




@app.route("/sec")
def fun44():
    return redirect(url_for('all'))

@app.route("/login")
def fun444():
    return render_template("login.html")
```

An IoT and Machine Learning based model for smart agriculture monitoring and alerting

```
@app.route('/aboutus')
def fun626():
    return render_template('aboutus.html')

@app.route('/contactus')
def fun726():
    return render_template('contactus.html')

@app.route('/subscrib')
def fun826():
    return render_template('subscribe.html')

@app.route("/mail")
def index():
    msg = Message('IOT Alert', sender = 'vnnproject21@gmail.com', recipients =
['vinay.telukanti@gmail.com'])
    x = "35"
    msg.body = "Dear Customer, \n Your Temperature is exceed than optimal.\n It's value is %s
. Please Take necesary precautions to avoid further damage" % x
    mail.send(msg)
    return "Successfully send"

@app.route("/monitoring")
def fun56():
    x = conn.get('/', None)
    dates = list(x.keys())
    select=dates[-1]

#select='6-11-2019'
date = dict(conn.get(f'{select}', None))
#date = conn.get("//"+z, None)
times = list((date.keys()))
pre=times[-1]
result=conn.get("//"+select+"/"+pre,None)
d = {}
for i in result.keys():
    for j in result[i]:
        # print(i, end=" : ")
        # print(result[i][j])
        d[i] = result[i][j]

#     msg = Message('IOT Alert', sender='vnnproject21@gmail.com',
recipients=['vinay.telukanti@gmail.com'])
    x = d["Temperature"]
    y = d["Humidity"]
    z = d["SoilMoist"]
    if(25<=x<34 and 60<=y<85 and 20<=z<50):
```

An IoT and Machine Learning based model for smart agriculture monitoring and alerting

```
pass
else:
    msg = Message('IOT Alert', sender='vnnproject21@gmail.com',
recipients=['vinay.telukanti@gmail.com'])

    msg.body = "Dear Customer,\n\nYour environmental conditions are exceed than optimal at
time {0}.\n\nTemperture = {1} Normal Range(25,34) \nHumidity = {2} Normal Range(60,85)
\nSoil Moisture = {3} Normal Range(20,50).\n\nPlease Take necesary precautions to avoid further
damage to your Crop\n\nThanks and Regards,\nManohar Reddy".format(pre,x,y,z)

    mail.send(msg)
return d
```

```
@app.route("/getData",methods = ['POST', 'GET'])
```

```
def fun1():
    if request.method == 'POST':
        d = request.form['date']
        t = request.form['time']
```

```
result = conn.get("/{}/{}{}".format(d,t, None))
```

```
d={}
for i in result.keys():
    for j in result[i]:
        #print(i, end=" : ")
        #print(result[i][j])
        d[i]=result[i][j]
```

```
#msg = Message('IOT Alert', sender='vnnproject21@gmail.com',
recipients=['vinay.telukanti@gmail.com'])
```

```
x=d["Temperature"]
```

```
y=d["Humidity"]
```

```
z=d["SoilMoist"]
```

```
#if(25<=x<34 and 60<=y<85 and 20<=z<50):
```

```
# pass
```

```
#else:
```

```
#     msg = Message('IOT Alert', sender='vnnproject21@gmail.com',
recipients=['vinay.telukanti@gmail.com'])
```

```
#     msg.body = "Dear Customer,\n\nYour environmental conditions are exceed than
optimal.\n\nTemperture = {0} Normal Range(25,34) \nHumidity = {1} Normal Range(60,85)
\nSoil Moisture = {2} Normal Range(20,50).\n\nPlease Take necesary precautions to avoid further
damage to your Crop\n\nThanks and Regards,\nManohar Reddy".format(x,y,z)
```

```
#     mail.send(msg)
```

```
return render_template('secondpage.html',result=d)
```

```
@app.route("/demo")
```

```
def fun55():
    result = conn.get("/",None)
```

```
h={}
for i in result.keys():
    h[i]=i
return render_template('thirdpage.html',result=h)

@app.route("/getone",methods = ['POST', 'GET'])
def fun111():
    if request.method == 'POST':
        d = request.form['date']

    result = conn.get("/{}/{}+d, None)
    f = {}
    for i in result.keys():
        f[i] = result[i]

    #x = d["Temperature"]
    #y = d["Humidity"]
    #z = d["SoilMoist"]
    return render_template('thirdpage.html', result=f)
    #return f

@app.route("/subscribe",methods=['POST','GET'])
def all1():
    if request.method=="POST":
        t1=request.form["temp"]
        t3=request.form["hum"]
        t2=request.form["soilmoist"]
        t1 = int(t1)
        t2 = int(t2)
        t3 = int(t3)
        x = pd.read_csv('out.csv')
        X = x.iloc[:, :3]
        y = x.iloc[:, 3]
        from sklearn.preprocessing import LabelEncoder
        le = LabelEncoder()
        y = le.fit_transform(y)

    from sklearn.model_selection import train_test_split
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=123)

from sklearn.tree import DecisionTreeClassifier
    trc = DecisionTreeClassifier(criterion='entropy', random_state=0)
    trc.fit(X_train, y_train)

    y_pred = trc.predict([[t1, t2, t3]])
    print(type(t1))
```

An IoT and Machine Learning based model for smart agriculture monitoring and alerting

```
print(y_pred[0])
if (y_pred[0] == 1):
    z="Favourable for growth"
else:
    z="Difficult to Sustain"
return render_template('subscribe.html',value=1,z=z)
else:
    return render_template('subscribe.html',value=0)

@app.route("/new",methods=['POST','GET'])
def all():
    x = conn.get('/', None)
    dates = list(x.keys())
    if request.method=="POST":
        select=request.form['select']
        #print(select)
        d = []
        f = []
        p = []
        t = []
        tm=[]
        hum=[]
        sm=[]
        date = dict(conn.get(f'{select}', None))
        times = list((date.keys()))
        #print(times)
        for i in times:
            time = date[i]

            print(time)
            di = (list(time['Temperature'].keys()))
            di=di[0]
            #print(di)
            tm.append(time['Temperature'][di])
            di = (list(time['Humidity'].keys()))
            di=di[0]
            hum.append(time['Humidity'][di])
            di = (list(time['SoilMoist'].keys()))
            di=di[0]
            sm.append(time['SoilMoist'][di])
            #print(sm)
        #return di
        return render_template('hhnew.html',value=1,length=[i for i in range(len(times))],dates=dates,select=select,times=times,tm=tm,hum=hum,sm=sm)
    else:
        return render_template('hhnew.html',value=0,dates=dates)
```

```
if __name__ == "__main__":
    app.run(debug=True)
```

/* Code for Landing Page */

```
<!DOCTYPE html>
<html>
<head>
    <title></title>
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-  
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
    <link rel="import" href="https://fonts.googleapis.com/css?">

<script type="text/javascript" src="https://www.48hourslogo.com/48hourslogo_data/2018/05/08/73072_1525787484.png  
BB3FDD723AC1/main.js?attr=zxSTxOkyeI1jpQynynVmwmHrlxL5QNFx8It9hymezqNVK7cJn1  
4QisxyyUVpef8Pprn_vZrV3kUM66v9oG8t9A" charset="UTF-8"></script><style>
    .subcon {
        width:350px;
        margin-left: 5%;
        height: 400px;
        border-radius: 5px;
        background:white;
        box-shadow: 0px 5px;
    }
    .maincon {
        margin-top: 100px;
        margin-left:2%;
    }
    .subcon img{
        width:350px;
        border-radius: 5px;
```

```
        }
    body{
        background:white;
    }
    .nav1{
        display:flex;
        background:transparent;
        height: 80px;
    }
    .a1 a{
        text-decoration: none;
        color:grey;
        margin-left: 10px;
    }
    .a1 {
        padding-top: 15px;
    }

    height:50px;
    margin-left: 20px;
    width: 276px;
    margin-top: 15px;
}
.navtitle{
    font-size: 28px;
    text-align: center;
    margin-top: 5px;
}

height:50px;
padding-left: 12px;
width: 400px;
}
.log{
    margin-left: 200px;
    padding-top: 40px;
    height: 50px;
    margin-top: 20px;
    color:black;
}
```

```
        }  
        .log a{  
            text-decoration: none;  
            color:black;  
            margin-left: 40px;  
  
        }  
        .fixbut{  
            position: fixed;  
            bottom: 15px;  
            right:20px;  
            background:#17a2b8;  
            color: white;  
            border-radius: 8px;  
  
            outline: none;  
            border:none;  
        }  
        .supre{  
            text-overflow: ellipsis;  
            overflow: hidden;  
            line-height: 16px;  
        }  
        #img1{  
            width: 45%;  
  
            height: 350px;  
            float: right;  
            /* right: 9%; */  
            margin-right: 80px;  
            margin-top:30px;  
        }  
        #img2{  
            width: 300px;  
  
            height: 200px;  
            margin-left: 0px;
```

```
margin-right: 990px;
}

</style>

<meta name="viewport" content="width=device-width">

</head>

<body>

<!--<button class="fixbut" style="text-overflow:inherit;">THIS IS FIXED</button>-->

<div class="nav1">

    <div class="navtitle"><!--<span
style="font-weight:600; color:#14d5e2">VIDHYACONNECT</span>--></div>

    <!--<div class="a1">
        <a href="#">HOME</a>
        <a href="#">SUPPORT</a>
        <a href="#">DETAILS</a>
    </div>-->

    <div class="log">
        <a href="http://localhost:5000/">Home</a>
        <a href="http://localhost:5000/subscribe">Prediction</a>
        <a href="http://localhost:5000/aboutus">About Us</a>
        <a href="http://localhost:5000/contactus">Contact Us</a>
    </div>

    <!--
        <a href="#">DETAILS</a><a href="#">LOGIN</a>
    --></div>

</div>

<div class="jumbotro" style="position: relative; margin-top: 90px;">

    <span style="/*position: absolute; left: 5%; width: 550px; top: 30%; font-size:
28px; font-weight: 3px; color: black; */
    /*position: absolute; left: 5%; width: 467px; top: 30%; font-size: 28px; margin-top:
50px;
    color: #d11814; text-align: left; font-family: 'Roboto', sans-serif; */


```

An IoT and Machine Learning based model for smart agriculture monitoring and alerting

font-weight:600;">Smart Agriculture using IOT

```
<span style="/*position: absolute;left:5%;width:550px;top:30%;font-size:28px;font-weight: 3px;color: black;
```

```
*/position: absolute;left: 5%;width: 567px;top: 30%;font-size: 20px;font-weight: 3px;margin-top: 130px;
```

color: black;text-align: left;font-family: 'Roboto',sans-serif;

Predictive Analytics and Green House Automation using Internet of Things for Remote Monitoring and Alert
 Generation is about modern agriculture techniques. It offers monitoring and managing crop production within a synthetic environment provided by green house.

```
</span>
```

```
<button onclick="location.href='http://localhost:5000/login'" class="btn" style="/*position: absolute;top:35%;left:41%*/position: absolute;top: 35%;margin-top: 351px;left: 8%;background:#ffa100;border-radius:25px;width:233px;color:white;height:50px;font-
```

```
weight:800;">GET STARTED</button>
```

```
</div>
```

```
</body>
```

```
</html>
```

/* Code for Login page */

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
    <title></title>
```

```
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCBMQu3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
```

```
    <link rel="import" href="https://fonts.googleapis.com/css?">
```

```
<script type="text/javascript" src="https://www.48hourslogo.com/48hourslogo_data/2018/05/08/73072_1525787484.png" data-bbox="144 62 890 967">  
BB3FDD723AC1/main.js?attr=zxSTxOkyeI1jpQynynVmwMHrlxL5QNFx8It9hymezqNVK7cJn1  
4QisxxyUVpef8Pprn_vZrV3kUM66v9oG8t9A" charset="UTF-8"></script><style>  
    .subcon {  
        width:350px;  
        margin-left: 5%;  
        height: 400px;  
        border-radius: 5px;  
        background:white;  
        box-shadow: 0px 5px;  
    }  
    .maincon {  
        margin-top: 100px;  
        margin-left:2%;  
    }  
    .subcon img {  
        width:350px;  
        border-radius: 5px;  
    }  
    body {  
        background:white;  
    }  
    .nav1 {  
        display:flex;  
        background:transparent;  
        height: 80px;  
    }  
    .a1 a {  
        text-decoration: none;  
        color:grey;  
        margin-left: 10px;  
    }  
    .a1 {
```

```
padding-top: 15px;  
height:50px;  
margin-left: 20px;  
width: 276px;  
margin-top: 15px;  
}  
.navtitle{  
font-size: 28px;  
text-align: center;  
margin-top: 5px;  
  
height:50px;  
padding-left: 12px;  
width: 400px;  
}  
.log{  
margin-left: 200px;  
  
padding-top: 40px;  
height: 50px;  
margin-top: 20px;  
color:black;  
  
}  
.log a{  
text-decoration: none;  
color:black;  
margin-left: 40px;  
  
}  
.fixbut{  
position: fixed;  
bottom: 15px;  
right:20px;  
background:#17a2b8;  
color: white;  
border-radius: 8px;  
outline: none;
```

```
border:none;
}
.supre{
    text-overflow: ellipsis;
    overflow: hidden;
    line-height: 16px;
}
#img1{
    width: 45%;
height: 300px;
float: right;
/* right: 9%; */

margin-right: 100px;
}
#img2{
    width: 300px;
height: 200px;
margin-left: 0px;
margin-right: 990px;
}
</style>
<meta name="viewport" content="width=device-width">
</head>
<body>
<div class="nav1">
    <div class="navtitle"><!--<span
style="font-weight:600; color:#14d5e2">VIDHYACONNECT</span>--></div>
    <!--<div class="a1">
        <a href="#">HOME</a>
        <a href="#">SUPPORT</a>
        <a href="#">DETAILS</a>

    </div>-->
    <div class="log">
        <a href="http://localhost:5000/">Home</a>

```

An IoT and Machine Learning based model for smart agriculture monitoring and alerting

```
<a href="http://localhost:5000/subscribe">Prediction</a>
    <a href="http://localhost:5000/aboutus">About Us</a>
    <a href="http://localhost:5000/contactus">Contact Us</a>

    <!--
        <a href="#">DETAILS</a><a href="#">LOGIN</a>
    --></div>
</div>

<center><br><br><br><br><br><br>
    <h2>Enter the below details </h2>
<form action="http://localhost:5000/new" method="get">
    Username:<br>
    <input type="text" name="date" >
    <br>
    Password:<br>
    <input type="password" name="time" >
    <br><br>
    <input type="submit" onclick="location.href='http://localhost:5000/sec'" value="Submit">
</form>
    </center>
</body>
</html>
```

/* Code for Main Page */

```
<html>
<head>
    <title></title>
    <link rel="stylesheet"
        href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
        integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
        crossorigin="anonymous">
    <link rel="import" href="https://fonts.googleapis.com/css?">
```

```
<script type="text/javascript" src="https://www.48hourslogo.com/48hourslogo_data/2018/05/08/73072_1525787484.png BB3FDD723AC1/main.js?attr=zxSTxOkyeI1jpQynynVmwmHrlxL5QNFx8It9hymeZqNVK7cJn1 4QisxyyUVpef8Pprn_vZrV3kUM66v9oG8t9A" charset="UTF-8"></script><style>
```

```
.subcon{  
    width:350px;  
    margin-left: 5%;  
    height: 400px;  
    border-radius: 5px;  
    background:white;  
    box-shadow: 0px 5px;  
}  
.maincon{  
    margin-top: 100px;  
    margin-left:2%;  
}  
.subcon img{  
    width:350px;  
    border-radius: 5px;  
}  
body{  
    background:white;  
}  
.nav1{  
    display:flex;  
    background:transparent;  
    height: 80px;  
}  
.a1 a{  
    text-decoration: none;  
    color:grey;  
    margin-left: 10px;  
}  
.a1{  
    padding-top: 15px;
```

```
height:50px;  
margin-left: 20px;  
width: 276px;  
margin-top: 15px;  
  
}  
  
.navtitle{  
    font-size: 28px;  
    text-align: center;  
    margin-top: 5px;  
  
    height:50px;  
    padding-left: 12px;  
    width: 400px;  
}  
  
.log{  
    margin-left: 200px;  
padding-top: 40px;  
height: 50px;  
margin-top: 20px;  
color:black;  
  
}  
  
.log a{  
    text-decoration: none;  
    color:black;  
    margin-left: 40px;  
  
}  
  
}  
  
.fixbut{  
    position: fixed;  
    bottom: 15px;  
    right:20px;  
    background:#17a2b8;  
    color: white;  
    border-radius: 8px;
```

```
outline: none;  
border:none;  
}  
.supre {  
  
  
text-overflow: ellipsis;  
overflow: hidden;  
line-height: 16px;  
}  
  
#img1 {  
width: 45%;  
height: 300px;  
float: right;  
/* right: 9%; */  
margin-right: 100px;  
}  
  
#img2 {  
width: 300px;  
height: 200px;  
margin-left: 0px;  
margin-right: 990px;  
}  
</style>  
<meta name="viewport" content="width=device-width">  
</head>  
<body>  
<div class="nav1">  
    <div class="navtitle"><!--<span  
style="font-weight:600; color:#14d5e2">VIDHYACONNECT</span>--></div>  
    <!--<div class="a1">  
        <a href="#">HOME</a>  
        <a href="#">SUPPORT</a>  
        <a href="#">DETAILS</a>  
  
    </div>-->  
    <div class="log">
```

An IoT and Machine Learning based model for smart agriculture monitoring and alerting

```
<a href="http://localhost:5000/">Home</a>
<a href="http://localhost:5000/subscribe">Prediction</a>
<a href="http://localhost:5000/aboutus">About Us</a>
<a href="http://localhost:5000/contactus">Contact Us</a>

<!--
    <a href="#">DETAILS</a><a href="#">LOGIN</a>
--></div>
</div>
<br><br>
<center>
<br><br>
<marquee width="50%" direction="left" height="5%" bgcolor="white">
Select any particular date to view all the environment values.</marquee><br><br>
<form action="/new" method="POST">
<select name="select">
    {%for i in dates%}
        <option>{{i}}</option>
    {%endfor%}
</select>
<input type="submit" value="Get Them">
</form>
{%if value==1%}
{%for i in length%}
    Date :{{select}}
    <br>Time: {{times[i]}}
    <br>Temperature:{{tm[i]}}
    <br>Humidity:{{hum[i]}}
    <br> Soil Moisture:{{sm[i]}}
<br><br><br>
{%endfor%}
{%endif%}
</center>
</body>
</html>
```

/* Code for Aboutus Page */

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title></title>
    <link rel="stylesheet"
        href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
        integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
        crossorigin="anonymous">
    <link rel="import" href="https://fonts.googleapis.com/css?">

    <script type="text/javascript" src="
https://www.48hourslogo.com/48hourslogo_data/2018/05/08/73072_1525787484.png
BB3FDD723AC1/main.js?attr=zxSTxOkyeI1jpQynynVmwmHrlxL5QNFx8It9hymezqNVK7cJn1
4QisxyyUVpef8Pprn_vZrV3kUM66v9oG8t9A" charset="UTF-8"></script><style>
.subcon {
    width:350px;
    margin-left: 5%;
    height: 400px;
    border-radius: 5px;
    background:white;
    box-shadow: 0px 5px;
}
.maincon {
    margin-top: 100px;
    margin-left:2%;
}
.subcon img{
    width:350px;
    border-radius: 5px;
}
```

```
body{  
    background:white;  
}  
.nav1{  
    display:flex;  
    background:transparent;  
    height: 80px;  
}  
.a1 a{  
    text-decoration: none;  
    color:grey;  
    margin-left: 10px;  
}  
.a1 {  
    padding-top: 15px;  
    height:50px;  
    margin-left: 20px;  
    width: 276px;  
    margin-top: 15px;  
}  
.navtitle{  
    font-size: 28px;  
    text-align: center;  
    margin-top: 5px;  
  
    height:50px;  
    padding-left: 12px;  
    width: 400px;  
}  
.log{  
    margin-left: 200px;  
  
padding-top: 40px;  
height: 50px;  
margin-top: 20px;  
color:black;
```

```
        }  
        .log a{  
            text-decoration: none;  
            color:black;  
            margin-left: 40px;  
  
        }  
        .fixbut{  
            position: fixed;  
            bottom: 15px;  
            right:20px;  
            background:#17a2b8;  
            color: white;  
            border-radius: 8px;  
            outline: none;  
            border:none;  
        }  
        .supre{  
            text-overflow: ellipsis;  
            overflow: hidden;  
            line-height: 16px;  
        }  
        #img1{  
            width: 45%;  
            height: 300px;  
            float: right;  
            /* right: 9%; */  
            margin-right: 100px;  
        }  
        #img2{  
            width: 300px;  
            height: 200px;  
            margin-left: 0px;  
            margin-right: 990px;  
        }
```

```
</style>
<meta name="viewport" content="width=device-width">
</head>
<body>
<div class="nav1">
    <div class="navtitle"><!--<span
style="font-weight:600; color:#14d5e2">VIDHYACONNECT</span>--></div>
    <!--<div class="a1">
        <a href="#">HOME</a>
        <a href="#">SUPPORT</a>
        <a href="#">DETAILS</a>

    </div>-->
    <div class="log">
        <a href="http://localhost:5000/">Home</a>
        <a href="http://localhost:5000/subscribe">Prediction</a>
        <a href="http://localhost:5000/aboutus">About Us</a>
        <a href="http://localhost:5000/contactus">Contact Us</a>

    <!--
        <a href="#">DETAILS</a><a href="#">LOGIN</a>
    --></div>
</div>
<br><br><br><br>
<div>
    <span style="font-family: 'Roboto', sans-serif; font-weight: 500;"> Agriculture is
done in every country from ages. Agriculture is the science and art of cultivating plants.
Agriculture was the key development in the rise of sedentary human civilization. Agriculture is
done manually from ages. As the world is trending into new technologies and implementations it is
a necessary goal to trend up with agriculture also. IOT plays a very important role in smart
agriculture. IOT sensors are capable of providing information about agriculture fields. <br><br>
We have proposed an IOT and smart agriculture system . This IOT based Agriculture monitoring
system makes use of sensor networks that collects data from different sensors deployed at various
nodes and sends it through the wireless protocol.<br> This smart agriculture using IOT system is
powered by Arduino, it consists of Temperature sensor, Moisture sensor, Humidity sensor. When
```

the IOT based agriculture monitoring system starts it checks the Temperature, humidity and moisture level. It sends E-mail alert to the customer about the levels.

This is system is also used to make predictions about the plant sustainability depending on the sensor values.

Also the data from these sensors can be used for data mining for future purposes.

The Cost for devices and installation for this system is very less compared to the benefits which one can make out of it.

Thanks and Regards,

VNN

</div>

</body>

</html>

/* Code for contact Page */

<!DOCTYPE html>

<html lang="en">

<head>

<title></title>

<link rel="stylesheet"

href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"

integrity="sha384-

ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"

crossorigin="anonymous">

<link rel="import" href="https://fonts.googleapis.com/css?">

<script type="text/javascript" src="

https://www.48hourslogo.com/48hourslogo_data/2018/05/08/BB3FDD723AC1/main.js?attr=zxST

xOkyeI1jpQynynVmwmMHrlxL5QNFx8It9hymezqNVK7cJn14QisxyyUVpef8Pprn_vZrV3kUM66

v9oG8t9A" charset="UTF-8"></script><style>

.subcon {

```
width:350px;  
margin-left: 5%;  
height: 400px;  
border-radius: 5px;  
background:white;  
box-shadow: 0px 5px  
}  
.maincon{  
margin-top: 100px;  
margin-left:2%;  
}  
.subcon img{  
width:350px;  
border-radius: 5px;  
}  
body{  
background:white;  
}  
.nav1{  
display:flex;  
background:transparent;  
height: 80px;  
}  
.a1 a{  
text-decoration: none;  
color:grey;  
margin-left: 10px;  
}  
.a1{  
padding-top: 15px;  
  
height:50px;  
margin-left: 20px;  
width: 276px;  
margin-top: 15px;  
}  
.navtitle{
```

```
font-size: 28px;  
text-align: center;  
margin-top: 5px;  
  
height:50px;  
padding-left: 12px;  
width: 400px;  
}  
.log{  
margin-left: 200px;  
padding-top: 40px;  
height: 50px;  
margin-top: 20px;  
color:black;  
  
}  
.log a{  
text-decoration: none;  
color:black;  
margin-left: 40px;  
  
}  
.fixbut{  
position: fixed;  
bottom: 15px;  
right:20px;  
background:#17a2b8;  
color: white;  
border-radius: 8px;  
  
outline: none;  
border:none;  
}  
.supre{  
text-overflow: ellipsis;  
overflow: hidden;
```

```
        line-height: 16px;  
    }  
  
    #img1{  
        width: 45%;  
        height: 300px;  
        float: right;  
        /* right: 9%; */  
        margin-right: 100px;  
    }  
  
    #img2{  
        width: 300px;  
        height: 200px;  
        margin-left: 0px;  
        margin-right: 990px;  
    }  
}</style>  
<meta name="viewport" content="width=device-width">  
</head>  
<body>  
<div class="nav1">  
    <div class="navtitle"><!--<span style="font-weight:600; color:#14d5e2">VIDHYACONNECT</span>--></div>  
    <!--<div class="a1">  
        <a href="#">HOME</a>  
        <a href="#">SUPPORT</a>  
        <a href="#">DETAILS</a>  
    </div>-->  
    <div class="log">  
        <a href="http://localhost:5000/">Home</a>  
        <a href="http://localhost:5000/subscribe">Prediction</a>  
        <a href="http://localhost:5000/aboutus">About Us</a>  
        <a href="http://localhost:5000/contactus">Contact Us</a>
```

An IoT and Machine Learning based model for smart agriculture monitoring and alerting

```
<a href="#">DETAILS</a><a href="#">LOGIN</a>
--></div>
</div>
<br><br>
<center>
<br><br>
<h1>
    Contact us incase of any Queries
</h1>
<br>
<h4>
    Name : Vinay Kumar<br>
    Phone No : 8885539371<br>
    E-Mail : vnnproject21@gmail.com<br>
</h4>
</body>
</html>
```

/* Code for Prediction Page */

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title></title>
    <link rel="stylesheet"
        href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
        integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
        crossorigin="anonymous">
    <link rel="import" href="https://fonts.googleapis.com/css?">

    <script type="text/javascript" src="http://gc.kis.v2.scr.kaspersky-labs.com/FD126C42-
EBFA-4E12-B309-
BB3FDD723AC1/main.js?attr=zxSTxOkyeI1jpQynynVmwmHrlxL5QNFx8It9hymezqNVK7cJn1
        crossorigin="anonymous">
```

4QisxxyUVpef8Pprn_vZrV3kUM66v9oG8t9A" charset="UTF-8">></script><style>

```
.subcon {  
    width:350px;  
    margin-left: 5%;  
    height: 400px;  
    border-radius: 5px;  
    background:white;  
    box-shadow: 0px 0px 5px;  
}  
.maincon {  
    margin-top: 100px;  
    margin-left:2%;  
}  
.subcon img{  
    width:350px;  
    border-radius: 5px;  
}  
body{  
    background:white;  
}  
.nav1{  
    display:flex;  
    background:transparent;  
    height: 80px;  
}  
.a1 a{  
    text-decoration: none;  
    color:grey;  
    margin-left: 10px;  
}  
.  
.a1 {  
    padding-top: 15px;  
    height:50px;  
    margin-left: 20px;  
    width: 276px;
```

```
margin-top: 15px;  
}  
.navtitle{  
    font-size: 28px;  
    text-align: center;  
    margin-top: 5px;  
  
    height:50px;  
    padding-left: 12px;  
    width: 400px;  
}  
.log{  
    margin-left: 200px;  
padding-top: 40px;  
height: 50px;  
margin-top: 20px;  
color:black;  
  
}  
.log a{  
    text-decoration: none;  
    color:black;  
    margin-left: 40px;  
  
}  
.fixbut{  
    position: fixed;  
    bottom: 15px;  
    right:20px;  
  
background:#17a2b8;  
color: white;  
border-radius: 8px;  
outline: none;  
border:none;  
}
```

```
.supre{  
    text-overflow: ellipsis;  
    overflow: hidden;  
    line-height: 16px;  
}  
  
#img1{  
    width: 45%;  
  
height: 300px;  
float: right;  
/* right: 9%; */  
margin-right: 100px;  
}  
  
#img2{  
    width: 300px;  
  
height: 200px;  
margin-left: 0px;  
margin-right: 990px;  
}  
  
</style>  
  
<meta name="viewport" content="width=device-width">  
</head>  
  
<body>  
  
<div class="nav1">  
    <div class="navtitle"><!--<span style="font-weight:600; color:#14d5e2">VIDHYACONNECT</span>--></div>  
    <!--<div class="a1">  
        <a href="#">HOME</a>  
        <a href="#">SUPPORT</a>  
        <a href="#">DETAILS</a>  
    </div>-->  
    <div class="log">  
        <a href="http://localhost:5000/">Home</a>  
        <a href="http://localhost:5000/subscribe">Prediction</a>  
        <a href="http://localhost:5000/aboutus">About Us</a>  
        <a href="http://localhost:5000/contactus">Contact Us</a>  
    </div>  
</div>
```

```
<!--  
<a href="#">DETAILS</a><a href="#">LOGIN</a>  
--></div>  
</div>  
<br><br><br><br>  
<center>  
    <h2>Enter the below details </h2>  
<form action="http://localhost:5000/subscribe" method="post">  
    Temperature:<br>  
    <input type="text" name="temp" >  
    <br>  
    Humidity:<br>  
    <input type="text" name="hum">  
    <br>  
    SoilMoisture:<br>  
    <input type="text" name="soilmoots" >  
    <br><br>  
    <input type="submit" value="Submit">  
</form>  
<br><br>  
    {%if value==1%}  
        Output : {{z}}  
  
<br><br>  
    {%endif%}  
</center>  
</body>  
</html>
```