

**Chapter-1 INTRODUCTION**

## Chapter-1 INTRODUCTION

### General About-

India is a developing country whose more than 40% of peoples do farming. The main source of income for India is agriculture. So, India is an agriculture-based country. And for agriculture water is very important aspect. On earth, there is only 0.5% of water is useful for farming. As per the reports, there is lots of water wastage in the earth and it is a very dangerous problem so we have to find solution for smart and efficient way of irrigation. In the fast-moving world, the human requires that everything is done very quick and fast without manual inaction.

Our Lifestyle wants everything around us to be operated without any manual interaction. And without any manual interaction for that simply we can say automation. For the automation, Internet of Things (IoT) is the best technology. The IoT is nothing but creating connection of different intelligent and the self-configuring nodes. That all nodes are connected to each other in a network. IoT is a network of devices that are connected to each other, and then they communicate to perform given tasks. And all those tasks are done without any manual inaction i.e., without any human to human or human to computer disturbance.

IoT is new technology which is demanding indifferent sectors like agriculture, healthcare, retail, transport, environment, supply chain management, infrastructure monitoring, etc. The use of IoT in agriculture will be going to help the farmer. The objective of this project is to measure accurate temperature and humidity inside the greenhouse. And automate the fogger or cooling fan based on present humidity and temperature level so the atmosphere inside a greenhouse maintains. Also, automate irrigation by detecting the exact moisture level in the soil and give only the required amount of water to the plants. And it will be provided only irrigate nutrient diluted water to all crops. It will add nutrients in the water every time. And it will show the data of all sensors on the website and mobile application with good looking UI.

A greenhouse can be defined as a closed structure that is used to protect the plants from external factors such as climatic conditions, pollution, etc. Basic factors affecting plant growth are sunlight, the water content in the soil, temperature, humidity, etc. Manual irrigation using buckets, and watering cans, flood irrigation, drip irrigation, sprinkler irrigation is still being used today. The previous system has several limitations; leaching off soil nutrients, erosion due to flooding, loss of water from plant surfaces through evaporation, water wastage which can result in water scarcity in drought areas and production of unhealthy crops.

There are some devices are in the market which waters the soil from time to time. They do not sense the moisture inside. This problem can be rectified if we use Monitoring, and Controlling System using the Arduino Platform in which the irrigation, monitoring and controlling will be automated.

### Purpose of the project-

The purpose of greenhouse management system is to simplify the complexity of managing greenhouse’s environment, such as temperature, moisture, humidity, sunlight, water irrigation using sensors, network and cloud technologies. Also, it collects data to represent graph by which user can take essential decision and control multiple aspect of system.

### Scope of the project-

The aim of the project is to manage and control the greenhouse system that will remotely control greenhouse environment, using a microcontroller, sensors, cloud-based application which will collect the data like temperature, moisture, humidity, lux from greenhouse. The objective of this project is to implement a low cost, reliable and scalable greenhouse management system that can help to manage factors that affect crop growth. This project work is complete based on electronic sensors which are connected to cloud services and automatically control the components of greenhouse and also collect data and stream data to cloud on user’s dashboard.



**Chapter -2 LITERATURE SURVEY**

## Chapter -2 LITERATURE SURVEY

### Existing System-

This system is made up of Arduino microcontrollers. Arduino can receive input from various sensors and it can control motors, lights and other actuators devices. Some sensors, DHT11 sensor, LDR sensor, Soil moisture sensor, and pH sensor is used in this project. The soil moisture sensor is used to measures the water content inside the soil. pH sensor measures the pH of the soil. LDR sensor is used to measure light intensity.

Devices like a DHT11 sensor, LDR sensor, Soil moisture sensor, water pump, artificial light, and servo motor are also connected to the Arduino which help to maintain the to the mobile user, and the mobile user turns on the water pump by sending input. When the temperature comes to the normal range, the mobile user turns off the roofs by sending another input. When humidity exceeds a defined level, the system sends input to the mobile user, and the mobile user turns on the exhaust fan by sending another input. When the humidity comes to the normal range, the mobile user turns off the exhaust fan by sending another input.

When pH of the soil exceeds a defined level, the system sends input to the mobile user, and the mobile user turns on the motor pump which sprays acidic or alkaline solution by sending another input. Similarly, when light intensity is lower than a defined level, the system sends SMS to the mobile user, and the mobile user turns on the artificial lights by sending another input. Finally, when the soil moisture sensor does not sense moisture in the soil than the system sends input to the mobile user, and the mobile user turns on the water pump by sending another input.

To eliminate input charges, all environmental parameters are sending to the server through Ethernet and stored in the database. It has disadvantage that the water pump is going to be operated using Wi-Fi module through mobile, so controlling water pump user should carry his mobile phone, or any other device with internet connectivity.

### Proposed System-

The proposed system supports water management by sensing soil moisture and controlling the environment inside a greenhouse by measuring the parameters like temperature and humidity. The system continuously monitors the soil moisture and provides an accurate amount of water required to the crop by adding nutrients inside water automatically.

The system can also control the environment inside a greenhouse by sensing the humidity and temperature inside a greenhouse. The system works without any interrupt. It is a Low-cost system and effective with less power consumption and without any manual interaction. Users can monitor the system from a remote location through the website or mobile application. Cameras used to capture live videos of the greenhouse. By using these videos, the user able to see the real condition of the greenhouse and control the greenhouse remotely from any part of the world.



**Chapter-3**

**PROJECT PLAN/ACTION PLAN**

# Chapter-3

**PROJECT PLAN/ACTION PLAN**

Table No. 3.1: Plan of work

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **No. of Weeks** | **From Date** | **To Date** |
| Project Planning | 4 | 23-Mar-21 | 01-Apr-21 |
| Designing | 2 | 02-Apr-21 | 15-Apr-21 |
| Coding | 4 | 25-Apr-21 | 10-May-21 |
| Testing | 3 | 19-May-21 | 30-May-21 |
| Documentation | 1 | 01-Jun-21 | 08-Jun-21 |

Table No. 3.2: Gantt chart

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Activity Name** | **Month** | **March** | | | | **April** | | | | **May** | | | | **June** | | | |
| **Week** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** |
| **Plan of Project** | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Designing and module Formation** | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Coding** | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Testing** | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Final reporting writing/**  **Documentation** | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |



**Chapter-4**

**SYSTEM REQUAREMENTS**

## Chapter-4 SYSTEM REQUAREMENTS

### Software Requirements-

Table No. 4.1: Software Requirements

|  |  |
| --- | --- |
| **SR NO.** | **SOFTWARE** |
| **1** | Arduino IOT Cloud |
| **2** | Arduino IDE |
| **3** | Browser |

### Hardware Requirements

Table No. 4.2: Hardware Requirements

|  |  |
| --- | --- |
| **SR NO.** | **HARDWARE COMPONENTS** |
| **1** | MKR 1000 WIFI (Micro-controller) |
| **2** | DHT 11 (Temperature and humidity sensor) |
| **3** | MQ 135 (Air quality sensor) |
| **4** | LDR (Light Density Resistor) |
| **5** | Soil Moisture Sensor Module |
| **6** | Battery |
| **7** | Stepper Motor |
| **8** | Arduino UNO (Microcontroller) |



**Chapter-5 SYSTEM ANALYSIS**

## Chapter-5

## SYSTEM ANALYSIS

### Activity Diagram-

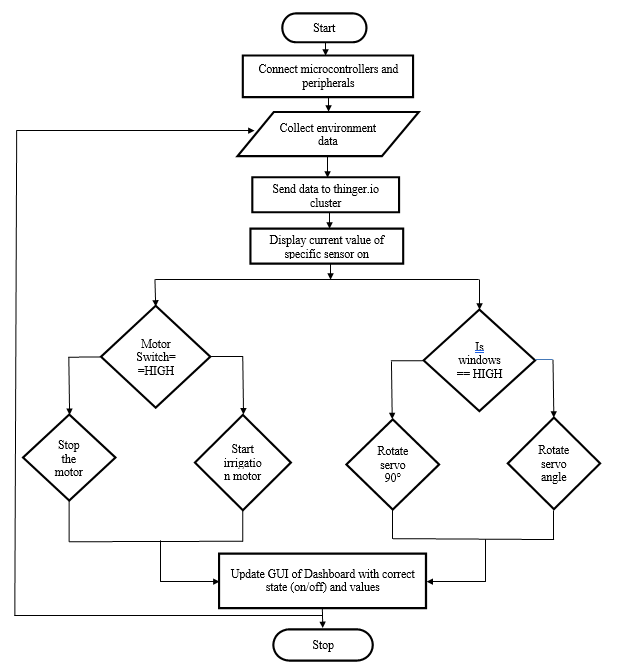


Figure No. 5.1: Activity Diagram

### Data Flow Diagram-

Level 0:

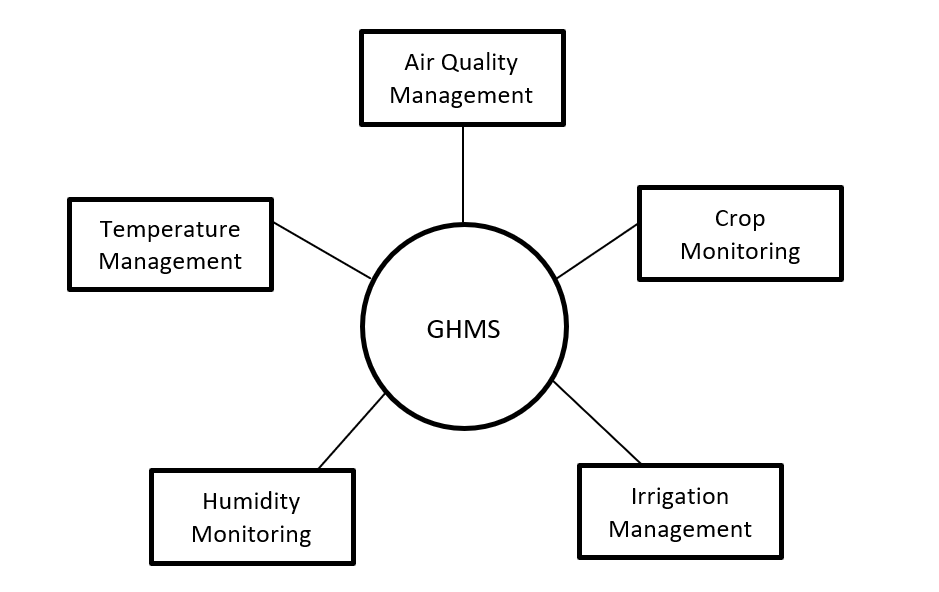
****

Figure No. 5.2: Data Flow Diagram Zero Level

Level 1:

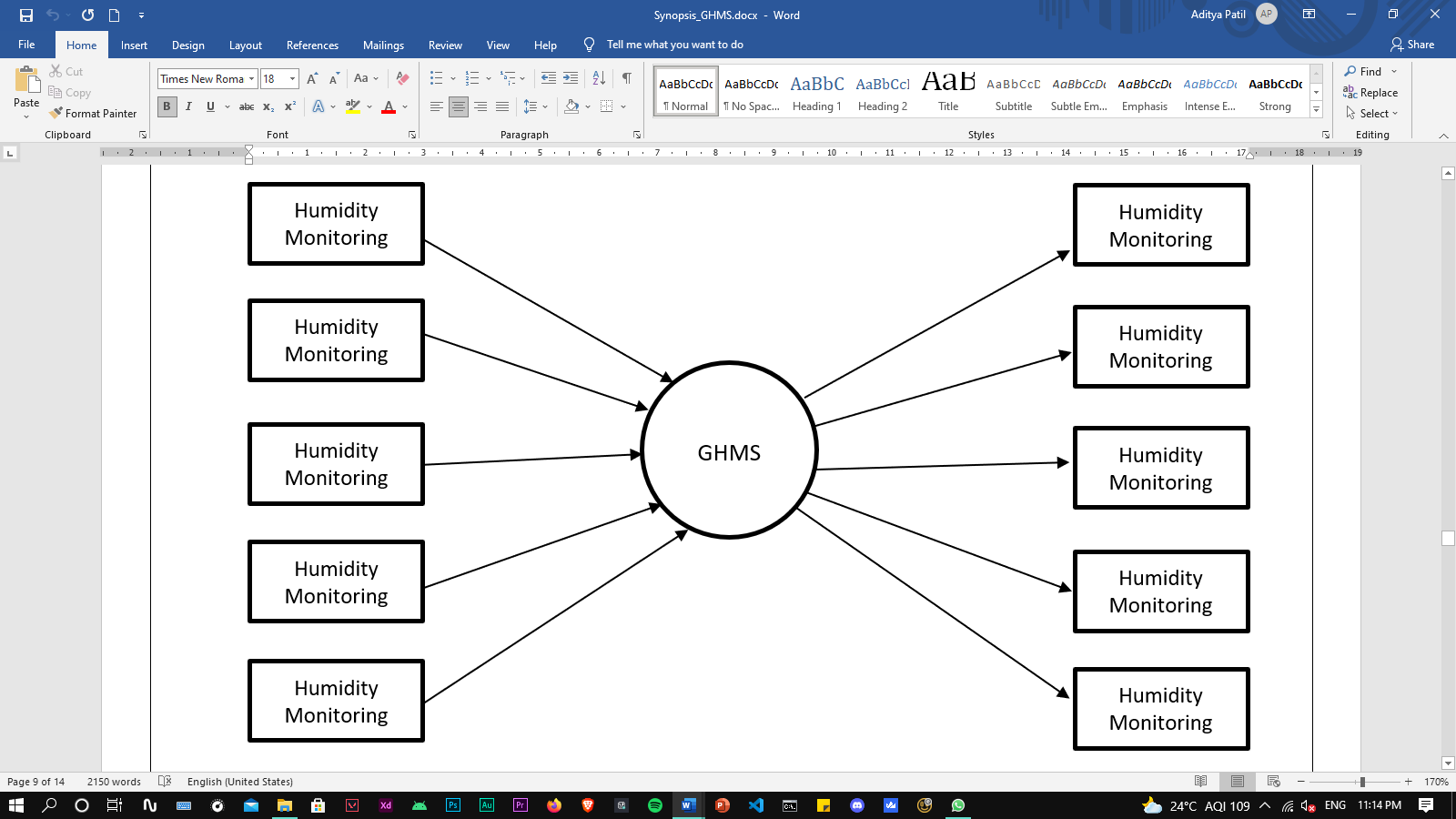


Figure No. 5.3: Data Flow Diagram One Level

Figure No. 5.4: Data Flow Diagram One Level

### Use Case Diagram-

Data management service

Figure No. 5.4: Use Case Diagram

Greenhouse Management System

User

Micro-

Controller



**Chapter-6 FEASIBILITY STUDY**

## Chapter-6 FEASIBILITY STUDY

### Introduction to Feasibility-

A feasibility study is an analysis that takes all of a project’s relevant factors into account—including economic, technical, legal, and scheduling considerations—to ascertain the likelihood of completing the project successfully. This project can be implemented using affordable electronic and software technology making it economically, technically and operationally feasible. A feasibility study aims to objectively and rationally uncover the strengths and weaknesses of an existing business or proposed venture, opportunities and threats present in the environment, the resources required to carry through, and ultimately the prospects for success. In its simplest terms, the two criteria to judge feasibility are cost required and value to be attained. A well-designed feasibility study should provide a historical background of the business or project, a description of the product or service, accounting statements, details of the operations and management, marketing research and policies, financial data, legal requirements and tax obligations. Generally, feasibility studies precede technical development and project implementation. A feasibility study evaluates the project's potential for success; therefore, perceived objectivity is an important factor in the credibility of the study for potential investors and lending institutions. It must therefore be conducted with an objective, unbiased approach to provide information upon which decisions can be based.

### Technical Feasibility-

This project is based on fingerprint scanners which are reasonably in phase with currently used technology. Therefore, it is very much favored by the technology. The technical feasibility assessment focuses on the degree to which the proposed development projects fits in with the existing business environment and objectives with regard to development schedule, delivery date, corporate culture, and existing business processes. To develop this project, we are using windows operating system. All operations done by using Ardiuno. Technically this project is feasible because it can handle the all problems related to existing system. This assessment is based on an outline design of system requirements.

To develop our system, we required knowledge of Arduino. There are easily available all the related software to develop this project. Ardiuno is easy to implement. So that, our system is technically feasible. This is platform independent.

### Operational Feasibility-

Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. Proposed is said to be operationally feasible as it provide the user friendly Graphical User Interface (GUI). User will able to understand the system clearly and correctly and can use the system with ease. User does not need any special training for operating this application. This application will be designed more user friendly as considering from the point of view of user. The Application will reduce the time consumed to maintain manual records and is not tiresome and cumbersome to maintain the records. Hence operational feasibility is assured.

### Economical Feasibility-

The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide. It includes quantification and identification of all the benefits expected. This assessment typically involves a cost/ benefits analysis. We know that, Android implementations are open source. We can get all the things related to android from internet. All the software required to develop this project are available on internet. So that, we can easily get these software to develop our system. We did not require buying any software for this project. So that, our system is economically feasible.

Economic feasibility of the application will be determined by:

• The Application will reduce a lot of labor work. Hence the Efforts will be reduced.

• Our Application will reduce the time that is wasted in manual processes.

• The storage and handling problems of the registers will be solved.

### 

### Legal Feasibility-

Determines whether the proposed system conflicts with legal requirements. We are developing this project for academic purpose and for solving the problems faced by peoples. This project will be useful for government and peoples. So that, our system is legally feasible.



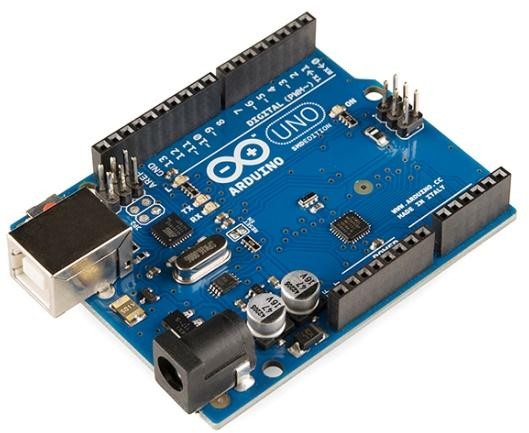
**Chapter-7 TECHNICAL OVERVIEW**

## Chapter-7 TECHNICAL OVERVIEW

### Arduino Uno-

The Arduino Uno is an [open-source](https://en.wikipedia.org/wiki/Open-source) [microcontroller boa*r*d](https://en.wikipedia.org/wiki/Microcontroller_board) based on the [Microchip](https://en.wikipedia.org/wiki/Microchip_Technology) [ATmega328P](https://en.wikipedia.org/wiki/ATmega328P) microcontroller and developed by Arduino. The board is equipped with sets of digital and analog [input/output](https://en.wikipedia.org/wiki/Input/output) (I/O) pins that may be interfaced to various [expansion boards](https://en.wikipedia.org/wiki/Expansion_board) (shields) and other circuits. [The board has

14 digital I/O pins (six capable of [PWM](https://en.wikipedia.org/wiki/Pulse-width_modulation) output), 6 analog I/O pins, and is programmable with the [Arduino IDE](https://en.wikipedia.org/wiki/Arduino#Software) (Integrated Development Environment), via a type B [USB cable.](https://en.wikipedia.org/wiki/USB_cable) It can be powered by the USB cable or by an external [9-volt](https://en.wikipedia.org/wiki/9-volt_battery) [battery,](https://en.wikipedia.org/wiki/9-volt_battery) though it accepts voltages between 7 and 20 volts. It is also similar to the [Arduino Nano.](https://en.wikipedia.org/wiki/Arduino_Nano)



**Figure No. 7.1:** Arduino Uno

**Application:**

* + - Prototyping of Electronics Products and Systems
    - Multiple DIY Projects.
    - Easy to use for beginner level DIYers and makers.
    - Projects requiring Multiple I/O interfaces and communications.

### DHT 11 (temperature and moisture sensor)-

The **DHT11**is a commonly used **Temperature and humidity sensor.** The sensor comes with a dedicated NTC to measure temperature and an 8-bit microcontroller to output the values of temperature and humidity as serial data. The sensor is also factory calibrated and hence easy to interface with other microcontrollers.

The sensor can measure temperature from 0°C to 50°C and humidity from 20% to 90% with an accuracy of ±1°C and ±1%. So, if you are looking to measure in this range then this sensor might be the right choice for you.

### 

**Figure No. 7.2:** DHT 11 Sensor

### Application:

* DHT11 Relative Humidity and Temperature Sensor can be used in many applications like:
* HVAC (Heating, Ventilation and Air Conditioning) Systems
* Weather Stations
* Medical Equipment for measuring humidity
* Home Automation Systems
* Automotive and other weather control applications

### Arduino MKR WiFi 1010 :-

The Arduino MKR WiFi 1010 is the easiest point of entry to basic IoT and pico-network application design. Whether you are looking at building a sensor network connected to your office or home router, or if you want to create a BLE device sending data to a cell phone, the MKR WiFi 1010 is your one-stop-solution for many of the basic IoT application scenarios.

### 

Figure No.**7.3**: Arduino MKR WiFi 1010

### Application:

* DHT11 Relative Humidity and Temperature Sensor can be used in many applications like:
* HVAC (Heating, Ventilation and Air Conditioning) Systems
* Weather Stations
* Medical Equipment for measuring humidity
* Home Automation Systems
* Automotive and other weather control applications

**LDR :-**

A light dependent resistor is an electronic component that is sensitive to light. When light falls upon it, then the resistance changes. Values of the resistance of the LDR may change over many orders of magnitude the value of the resistance falling as the level of light increases.



**Figure No. 7.4** : LDR

**Application:**

* The [LDR](https://www.semiconductorforu.com/applications-ldr-light-dependent-resistors/) is used in the infrared astronomy.
* The LDR is used in light failure alarm circuits and used in light meter.
* The LDR used in smoke detectors.
* It is used for automatic contrast and brightness control in television receivers.
* It is used in photosensitive [relay](https://www.semiconductorforu.com/applications-ldr-light-dependent-resistors/)
* It is used in optical coding.
* It is used in street light control circuits.
* It is used in camera light meters.
* It is used in the security alarm.
* It is used as a proximity switch.
* It is used in light activated control circuits.

**SG90 Servo Motor :-**

Most of the hobby Servo motors operates from 4.8V to 6.5V, the higher the voltage higher the torque we can achieve, but most commonly they are operated at +5V.  Almost all hobby servo motors can rotate only from 0° to 180° due to their gear arrangement so make sure you project can live with the half circle if no, you can prefer for a 0° to 360° motor or modify the motor to make a full circle. The gears in the motors are easily subjected to wear and tear, so if your application requires stronger and long running motors you can go with metal gears or just stick with normal plastic gear.

Next comes the most important parameter, which is the **torque** at which the motor operates. Again, there are many choices here but the commonly available one is the 2.5kg/cm torque which comes with the Towerpro SG90 Motor. This 2.5kg/cm torque means that the motor can pull a weight of 2.5kg when it is suspended at a distance of 1cm

****

**Figure No. 7.5**: SG90 Servo Motor

**Application:**

* This motor is used to activate movements in robotics for giving the arm to its exact angle.
* This motor is used to start, move and also stop the conveyor belts carrying the product along with many stages. For example, bottling, packaging and product labeling,
* This motor can be used in the camera to set a lens of the camera to enhance the focus of images
* This motor is built into the camera to correct a lens of the camera to improve out of focus images.
* This motor can be used to control the robotic vehicle by controlling robot wheels, speed, generating plenty torque to move and also start and stop the robot.
* It can be [used in solar](https://www.efxkits.us/solar-energy-system-working-with-applications/) tracking system to set the angle of the solar panel so that panel stays to face the sun.
* This motor is used in milling machines for metal cutting and forming to provide specific motion.
* This is used in Textile industries to control and knitting machines like spinning, weaving.
* This motor is used in automatic door opening and closing in public places like hospitals, theaters and supermarkets.

**Soil Moisture Sensor :-**

This **soil moisture sensor module** is used to detect the moisture of the soil. It measures the volumetric content of water inside the soil and gives us the moisture level as output. The module has both digital and analog outputs and a potentiometer to adjust the threshold level.



**Figure No. 7.6**: Soil Moisture Sensor

**Application:**

* Agriculture.
* Landscape irrigation.
* Research.
* Simple sensors for gardeners.

**Submersible Mini Water Pump :-**

The water pump can be defined as a pump which uses the principles like mechanical as well as hydraulic throughout a piping system and to make sufficient force for its future use. They have been approximately in one structure otherwise another because of early civilization. At present these pumps are utilized within a wide range of housing, farming, municipal, and manufacturing applications.

****

**Figure No. 7.7**: Submersible Mini Water Pump

**Application:**

* Used to push fluid with its rotor energy to the surface.

**MQ135:-**

The MQ-135 Gas sensors are used in air quality control equipment and are suitable for detecting or measuring of NH3, NOx, Alcohol, Benzene, Smoke, CO2. The MQ-135 sensor module comes with a Digital Pin which makes this sensor to operate even without a microcontroller and that comes in handy when you are only trying to detect one particular gas.  If you need to measure the gases in PPM the analog pin need to be used. The analog pin is TTL driven and works on 5V and so can be used with most common microcontrollers.

If you are looking for a sensor to detect or measure common air quality gases such as CO2, Smoke, NH3, NOx, Alcohol, Benzene then this sensor might be the right choice for you.



**Figure No. 7.8**: MQ135

**Application:**

* Air Quality Monitor
* Detection of harmful gases.
* Domestic air pollution detection.
* Industrial pollution detection.
* Portable air pollution detection.

### Battery -

A **battery** is a [power source](https://en.wikipedia.org/wiki/Power_source) consisting of one or more [electrochemical cells](https://en.wikipedia.org/wiki/Electrochemical_cell) with external connections for powering [electrical](https://en.wikipedia.org/wiki/Electricity) devices such as [flashlights](https://en.wikipedia.org/wiki/Flashlight), [mobile phones](https://en.wikipedia.org/wiki/Mobile_phone), and [electric cars](https://en.wikipedia.org/wiki/Electric_car). When a battery is supplying [electric power](https://en.wikipedia.org/wiki/Electric_power), its positive terminal is the [cathode](https://en.wikipedia.org/wiki/Cathode) and its negative terminal is the [anode](https://en.wikipedia.org/wiki/Anode). The terminal marked negative is the source of electrons that will flow through an external electric circuit to the positive terminal...



**Figure No. 7.9**: Battery

**Application:**

* House
* Health Instruments
* Medical
* Logistics and construction
* Firefighting and Emergency
* Military



**Chapter-8**

**SYSTEM DESIGN AND SNAPSHOT**

## Chapter-8

**SYSTEM DESIGN AND SNAPSHOT**

### 8.1 Source Code:

### Arduino MKR 1010 WiFi :-

#define \_DISABLE\_TLS\_

#include <ThingerWiFiNINA.h>

#define USERNAME "adityapatil688"

#define DEVICE\_ID "mkr1010"

#define DEVICE\_CREDENTIAL "mkr1010"

#define SSID "PC"

#define SSID\_PASSWORD "PASS"

#include <Servo.h>

Servo myservo1;

Servo myservo2;

#include "DHT.h"

#define DHTPIN 5 // Temperature and Humidity

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

ThingerWiFiNINA thing(USERNAME, DEVICE\_ID, DEVICE\_CREDENTIAL);

#include <MQUnifiedsensor.h>

#define placa "Arduino MKR"

#define Voltage\_Resolution 5

#define pin A2 //Analog input 0 of your arduino

#define type "MQ-135" //MQ135

#define ADC\_Bit\_Resolution 10 // For arduino UNO/MEGA/NANO

#define RatioMQ135CleanAir 3.6//RS / R0 = 3.6 ppm

//#define calibration\_button 13 //Pin to calibrate your sensor

MQUnifiedsensor MQ135(placa, Voltage\_Resolution, ADC\_Bit\_Resolution, pin, type);

float soil\_moisture;

int Servo\_;

void setup()

{ //27

Serial.begin(9600);

thing.add\_wifi(SSID, SSID\_PASSWORD);

pinMode(A0, INPUT); // Light

pinMode(A1, INPUT); // Soil Moisture

pinMode(A2, INPUT); // CO2

pinMode(10, OUTPUT);// Water Pump

pinMode(2, OUTPUT); // Servo Motor

myservo1.attach(1);

myservo2.attach(2);

dht.begin();

MQ135.setRegressionMethod(1);

MQ135.init();

float calcR0 = 0;

for(int i = 1; i<=10; i ++) //40

{

MQ135.update();

calcR0 += MQ135.calibrate(RatioMQ135CleanAir);

}

MQ135.setR0(calcR0/10);

}

void loop()

{ //44

thing.handle();

thing["Water pump"] << digitalPin(10);

thing["Temperature"] >> outputValue(dht.readTemperature());

thing["Humidity"] >> outputValue(dht.readHumidity());

soil\_moisture = 100 - ( (analogRead(A1)/1023.00) \* 100 );

thing["Soil Moisture"] >> outputValue(soil\_moisture);

thing["Light"] >> outputValue(analogRead(A0));

thing ["Servo\_"] << [](pson &in){

if(in.is\_empty())

in = (int)Servo\_;

else

{

Servo\_ = (int)in;

myservo1.write((int)in);

myservo2.write((int)in);

delay(500);

}

Servo\_ = in;

}; //60

MQ135.update();

MQ135.setA(110.47); MQ135.setB(-2.862);

float CO2 = MQ135.readSensor();

Serial.print("| ");

Serial.print(CO2);

Serial.println(" |");

thing["CO2"] >> outputValue(MQ135.readSensor() + 400); //67

}

### Arduino UNO :-

const int feedback = 5;

const int motorPin = 6;

int feedbackState = 0;

void setup() {

Serial.begin(9600);

pinMode(motorPin, OUTPUT);

pinMode(feedback, INPUT);

}

void loop() {

feedbackState = digitalRead(feedback);

if (feedbackState == HIGH) {

digitalWrite(motorPin, HIGH);

Serial.println("HIGH");

} else {

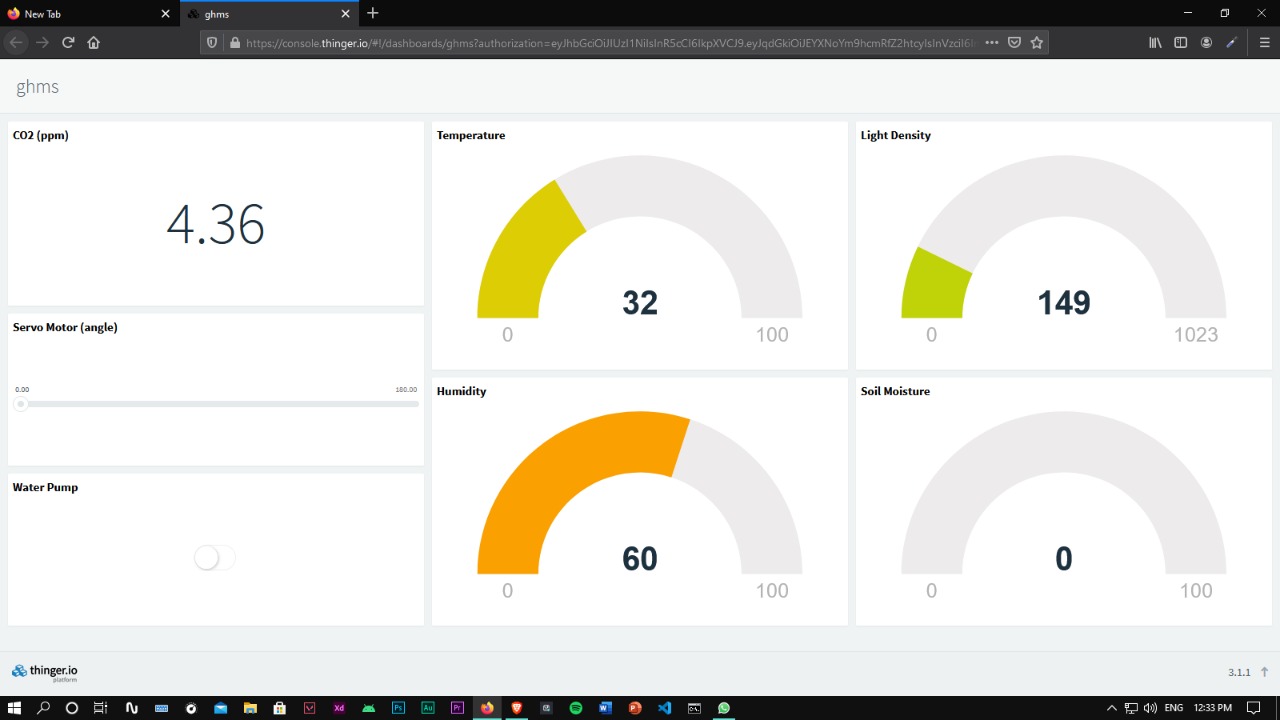
digitalWrite(motorPin, LOW);

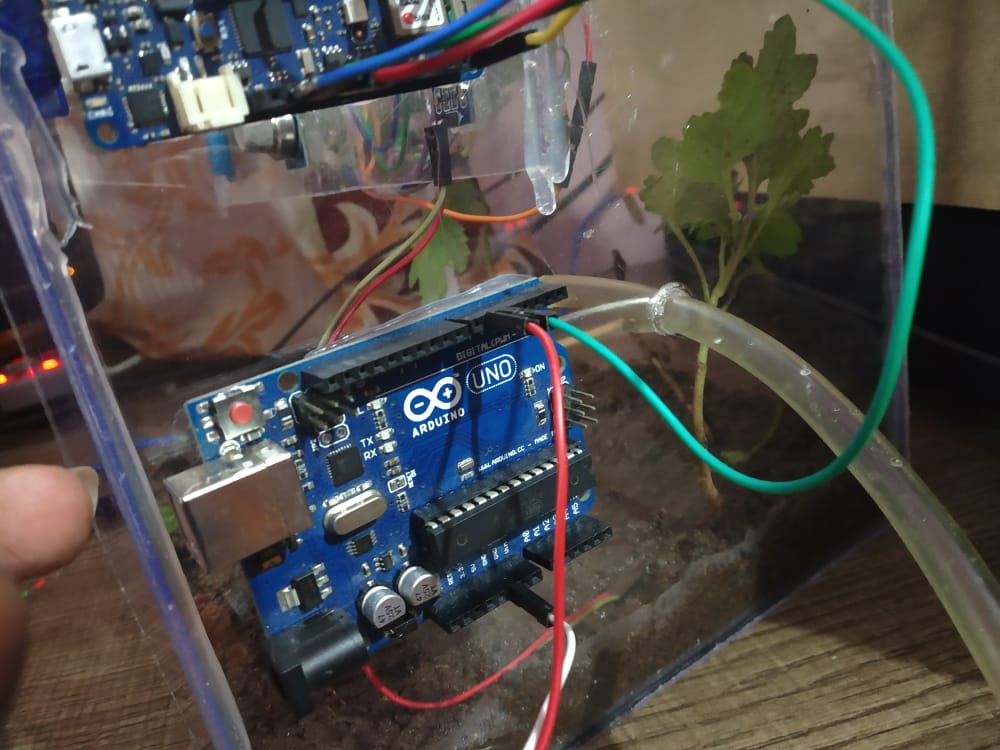
Serial.println("LOW");

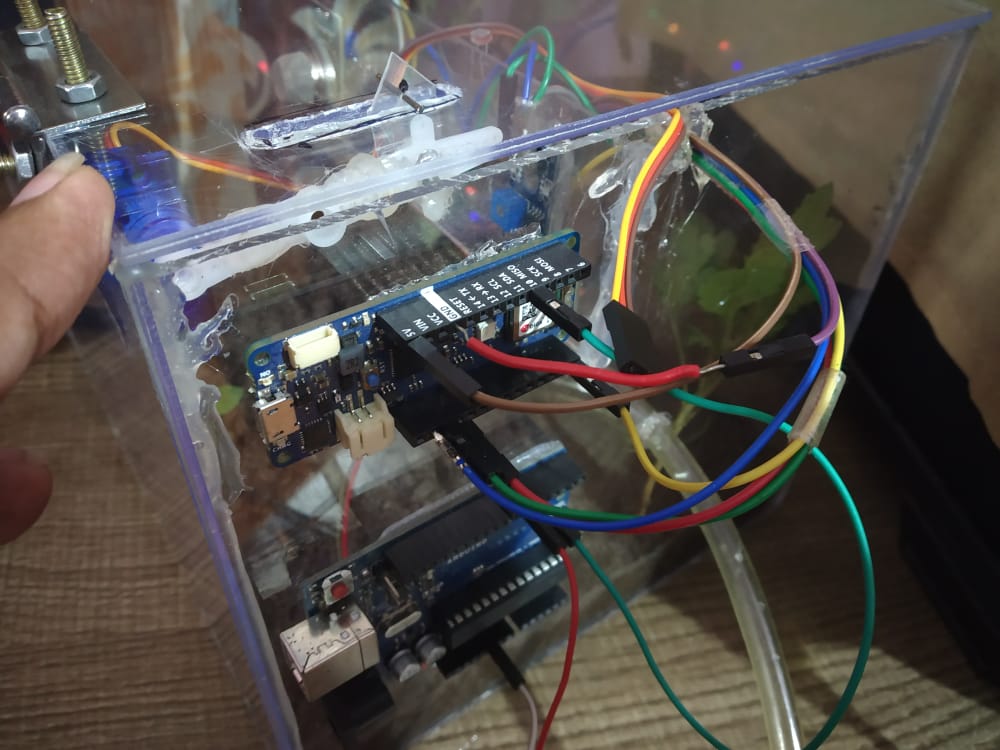
}

}

* 1. **Snapshot-**

****







**Chapter-9 TESTING PLAN**

# Chapter-9 TESTING PLAN

### Basic of Testing-

Software testing can be stated as the process of verifying and validating that a software or application is bug free, meets the technical requirements as guided by it’s design and development and meets the user requirements effectively and efficiently with handling all the exceptional and boundary cases.

The process of software testing aims not only at finding faults in the existing software but also at finding measures to improve the software in terms of efficiency, accuracy and usability. It mainly aims at measuring specification, functionality and performance of a software program or application.

### Test Cases-

Table No. 9.1: First test Scenario

**Test Scenario**: Verify on entering valid user id and password, the customer can login and Application Working.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr no. | Test Case | Expected result | Actual result | Pass/Fail |
| 1 | Check proper connection for all sensors | LED should turn on | As expected | Pass |
| 2 | Take reading(temp) | Reading should be displayed on cloud dashboard | As expected | Pass |
| 3 | Take reading(humidity) | Reading should be displayed on cloud dashboard | As expected | Pass |
| 4 | Take reading of gas in ppm | Reading should be displayed on cloud dashboard | As expected | Pass |
| 5 | Take reading of light | Reading should be displayed on cloud dashboard | As expected | Pass |
| 6 | Put the sensor in soil and take reading | Reading should be displayed on cloud dashboard | As expected | Pass |
| 7 | Check whether water pump motor turns on after clicking the button on dashboard | Motor should turn on | As expected | Pass |
| 8 | Check whether water pump motor turns off after clicking the button on dashboard | Motor should turn off | As expected | Pass |
| 9 | Check whether water passes through motor | Water should passes through motor | As expected | Pass |
| 10 | Check whether motor opens the roof, after sliding the slider to right | Motor should open the roof | As expected | Pass |
| 11 | Check whether motor closes the roof, after sliding the slider to right | Motor should close the roof | As expected | Pass |



**Chapter-10 APPLICATION AND ADVANTAGES**

**Chapter-10 APPLICATION AND ADVANTAGES**

### Applications:

1. Hydroponics.
2. Aquaponics.
3. Aeroponics.
4. Precise farming.
5. Vertical farming.

**Advantages:**

1. Maintain Ideal Micro-Climate Conditions
2. Enhance Irrigation and Fertilization Practices
3. Control Infection and Avoid Disease Outbreak
4. Prevent Thefts and Improve Security



**Chapter-11 FUTURE ENHANCEMENT**

## Chapter-11 FUTURE ENHANCEMENT

* Advancements in greenhouse management system are moving so fast, In future we will make advancement and multi functions like SMS alert if authorized person try to monitor their greenhouse weather .Image recognizing process system and AI system based.
* The drone technology will give a high- technology makeover to the agriculture industry with strategy and planning based on real-time data collection and processing, there are several ways in which ground-based and aerial drones will be utilized throughout the crop cycle. The common use of agricultural drones is surveying or mapping. The aerial perspective that drones provide decreases the need to go row by row to check the crop health.
* Accessing this system can be more easy by the integration of webhooks which helps to operate system using voice command with AI Assistants such as Google assistant, Amazon Alexa



**Chapter-12 COST OF PROJECT**

## Chapter-12 COST OF PROJECT

### LOC Based Estimation:

Table No. 12.1: line of code estimation

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Software Module** | **LOC** |
| 1. | Arduino MKR 1010 | 63 |
| 2. | Arduino UNO | 20 |
| **Total Estimated LOC** | | **83** |

We calculate the project cost on the basis of No of People, No of Working days and System requirement for software products.

|  |  |  |
| --- | --- | --- |
| Month required for Project development  Total Number of members in project team | =  = | **3.2 Months/100 days**  **04** |
| Total Estimated LOC | = | **83** |

### Using CCOMO (Constructive Cost Model) Model:

Given

|  |  |  |
| --- | --- | --- |
| Ab | = | 3.0 |
| Bb | = | 1.12 |
| Cb | = | 2.5 |
| Db | = | 0.35 |
| (Coefficient factors for Semidetached Project catageory) | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Efforts Estimation[E**] | = | [Ab\*(KLOC)^Bb] | |
|  | = | [3.0\*(0.083)^1.12] | |
|  | = | 3.O\*0.06 | |
|  | = | 0.18 Person-Month | |
|  | = | 1 Person-Month | |
| **Duration[D]** | **=** | Cb(E)^Db | |
| = | | 2.5(1)^0.35 |
| = | | 2.5\*1 |
| = | | 2.5 Month |
| = | | 2 Months |
| ]  **Person Estimation[p]** = | | E/D |
| = | | 1/6 |
| = | | 0.16 |
| = | | 0.2 Person |

**Cost Calculation:** As per the we assume that the software engineering cost for each 100 LOC is $10, therefore overall cost to develop the project is

= (Total LOC /100)\*rate

= (83/100)\*10

= 0.83 \*10

= $8.3

Table No. 12.1: Hardware Based Estimation

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sr. No.** | | **Component** | | **Quantity** | | **Price** | **Total Amt.** |
| 1. | | Arduino Uno | | 1 | | 330 | 330 |
| 2. | | Arduino MKR1010 | | 1 | | 3149 | 3149 |
| 3. | | Submersible water pump | | 1 | | 44 | 44 |
| 4. | | Mini servo motor | | 2 | | 74 | 148 |
| 5. | MQ2 gas sensor | | 1 | | 93 | | 93 |
| 6. | LDR | | 1 | | 6 | | 6 |
| 7. | DHT11 sensor | | 1 | | 73 | | 73 |
| 8. | Soil Moisture sensor | | 1 | | 58 | | 58 |
| **Total** | | | | | | | 3901 |



**Chapter-13 CONCLUSION**

## Chapter-13 CONCLUSION

The advantage of Smart Greenhouse over conventional farming is that we were able to produce insecticide and pesticide free crops and create a climate for the proper growth of plants and even provides alternative source of income through apiculture, selling tube well water etc. Moreover, this system can be installed by any individual in his house (Rooftop greenhouse), who doesn’t have knowledge about farming. Since one can maintain any climatic condition in this type of Greenhouse, it is possible to cultivate any type of crop. Hence, we grow plants like Hibiscus which are imported to India. We can reduce 70%-80% water requirement. It also increases yield and rate of growth and produces organic agricultural products. Most importantly, we are able to connect farmer directly to consumer using IoT, which can save him from the clutches of middlemen. It reduces effort and time of farmer and makes farming efficient and profitable activity.



**Chapter-14**

**BIBLIOGRAPHY**

**Chapter - 14**

**BIBLIOGRAPHY**

Reference books and websites used during the entire project.

### Book Reference-

* 1. IoT and Analytics for agriculture – Prashant Kumar Pattnaik
  2. Getting Started with Internet of Things – Cuno Pfister
  3. Iot And Edge Computing – Perry Lea

### Web Reference-

1. https://nevonprojects.com/greenhouse-monitoring-and-control-system-using-iot-project/www.usautomatic.com
2. https://www.iotsworldcongress.com/iot-transforming-the-future-of-agriculture/www.microchip.co.uk
3. https://www.biz4intellia.com/blog/5-applications-of-iot-in-agriculture/www.engineeringproject.com
4. https://www.digiteum.com/iot-agriculture/www.electronicsworldforyou.com
5. https://en.wikipedia.org/wiki/Greenhouse