**SMART AGRICULTURE USING IOT**

**A Project Report**

Submitted in partial fulfilment of the

Requirements for the award of the Degree of

# BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)

**By**

Dhruv Mandar Dave

KCTYBSCIT012

**Under the esteemed guidance of**

**Assistant Professor**

**Mrs. Pragati Thawani**

**&**

**Mr. Sadiq Batliwala**



**DEPARTMENT OF INFORMATION TECHNOLOGY**

**KISHINCHAND CHELLARAM COLLEGE**

***(Affiliated to HSNC University)* MUMBAI, 400020**

**MAHARASHTRA**

**2022-23**

PROFORMA FOR THE APPROVAL PROJECT PROPOSAL

PRN No.: 2021220210020021 Roll No: KCTYBSCIT012

**Name of the Student: Dhruv Mandar Dave**

**Title of the Project: Smart Agriculture using IOT**

**Name of the Guide: Mr. Sadiq Batliwala**

**Teaching experience of the Guide: 1.5 years**

Is this your first submission? Yes No



Signature of the Student Signature of the Guide

Date: ………………… Date: …………………….

Signature of the Co-Ordinator

Date: ………………

## KISHINCHAND CHELLARAM COLLEGE

***(Affiliated to HSNC University, Mumbai)***

**MUMBAI, MAHARASHTRA - 400020**

## DEPARTMENT OF INFORMATION TECHNOLOGY



**CERTIFICATE**

This is to certify that the project entitled “**SMART AGRICULTURE USING IOT”**, is bonafide work of Dhruv Dave bearing KCTYBSCIT012 submitted in partial fulfilment of the requirements for the award of degree of BACHELOR OF SCIENCE in INFORMATION TECHNOLOGY from HSNCU University.

**Internal Guide Co-Ordinator**

**External Examiner**

**Date: College Seal**

# ACKNOWLEDGEMENT

We would like to take this opportunity to express a deep sense of gratitude to Dr. Tejashree V. Shanbhag (Principal of K.C. College), Dr. Rakhi Gupta (H.O.D. of B.Sc. I.T. Department) and Mrs. Pragati V. Thawani (Asst. Professor of B.Sc. I.T. Department) for their cordial support, valuable information and guidance, which helped us in completing this task through various stages. We take this opportunity to express our profound gratitude and deep regards to our guide, Mr. Sadiq Batliwala for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by her time-to-time shall care us a long way in the journey of life on which we are about to embark.

We are obliged to staff members of K.C. College for the valuable information provided by them in their respective fields. We are grateful for their cooperation during the period of our project.

# DECLARATION

I hereby declare that the project entitled, **“Smart Agriculture using IOT”** done at **K.C. College** is done, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfillment of the requirements for the award of degree of

**BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as final semester project as part of our curriculum.

**Name and Signature of the Student**

TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Title** | **Page No** |
| 1 | Introduction | 08 |
|  | 1.1 Background | 09 |
|  | 1.2 Objectives | 09 |
|  | 1.3 Purpose, Scope and Applicability | 10 |
|  | 1.4 Achievements | 11 |
|  | 1.5 Organization of Report | 12 |
| 2 | Survey of Technology | 14 |
|  | 2.1 Front-End Features | 15 |
|  | 2.2 Back-End Features | 15 |
|  | 2.3 Justification | 15 |
| 3 | Requirement and Analysis | 16 |
|  | 3.1 Problem Definition | 17 |
|  | 3.2 Requirement Specification: Hardware, Software | 17 |
|  | 3.3 Planning and Scheduling | 19 |
|  | 3.4 Gantt Chart | 20 |
|  | 3.5 Preliminary Product Description | 20 |
| 4 | System Design | 21 |
|  | 4.1 Basic Modules | 22 |
|  | 4.2 Algorithmic Modules | 22 |
|  | 4.3 UML Diagrams | 24 |
|  | 4.4 Security Issues | 25 |
| 5 | Implementation and Testing | 26 |
|  | 5.1 Implementation Approaches | 27 |
|  | 5.2 Code Efficiency | 28 |
|  | 5.3 Testing Approach | 29 |
|  | 5.4 Modification and Improvements | 31 |
|  |  |  |
|  | 5.5 Test Cases | 31 |
| 6 | User Documentation | 32 |
| 7 | Cost Benefit Analysis | 36 |
| 8 | Conclusion | 39 |
|  | 8.1 Significance of the system  8.2 Limitations of the system  8.3 Future scope of the project | 40  41  41 |
| 9 | References | 42 |

# ABSTRACT

Smart agriculture is the use of information and communication technologies (ICTs) to improve agricultural practices. IoT sensors are a key component of smart agriculture, as they can be used to collect data on a variety of factors, such as soil moisture, humidity, temperature, and crop health. This data can then be used to make informed decision about crop management, such as when to irrigate, fertilize, or apply pesticides. The project uses four sensors: soil moisture, humidity, temperature, and motion. The soil moisture sensor will be used to determine the moisture level in the soil, and the humidity sensor will be used to determine the humidity level in the air. The temperature sensor will be used to determine the temperature of the air, and the motion sensor will be used to detect any movement in the field. The data collected by the sensors will be stored in a software and accessed through a web application. The web application will allow farmers to view the data and make informed decisions about crop management. The results of this project will demonstrate the potential of IoT sensors to improve agricultural practices. The project will also identify any gaps in the literature on smart agriculture and suggest future research directions.

Keywords - **soil moisture, humidity, temperature, motion, sensors, crop management.**

# CHAPTER 1: Introduction

**1.1 Background**

Smart Agriculture has become a quintessential part of modern farming. Leveraging the power of the Internet of Things (IoT), it provides a more precise and data-driven approach to agriculture that addresses the crucial issue of providing sustainable agricultural production to feed the continuously growing global population.

IoT has brought an unprecedented transformation in the agriculture department by enabling farmers and agribusinesses to face significant challenges in various ways. Traditional farming practices lack efficiency and are reliant on predictable environmental conditions. This can lead to a loss in crop yield and, consequentially, economic deficits.

Through the use of various IoT technologies, smart farming applications can trigger actions, such as irrigation, fertilizing, or pest control, based on precise, real-time data, thus mitigating inefficient resource usage and improving crop yields.

Smart Agriculture captures data from the environment using multiple sensors embedded in farming fields and equipment. Information gleaned from soil moisture sensors, weather stations, and drones provides valuable insights to farmers on the specific needs of individual fields and crops. This system of interconnectivity is not restricted to local farms and fields, but through the Internet, provides a global system of agronomic data that can be analysed, modelled, and utilized for better agricultural practice.

**1.2 Objectives**

* My project aims to harness the power of IoT in the development of a Smart Agriculture system.
* Tailored to optimize and streamline the existing agricultural practices.
* Bring an increase in productivity of crops.
* Timely prevention of crops getting hampered.
* Reduction in wastage of resources.
* Therefore, by giving more control over the process of farming, the adoption of IoT in agriculture ensures that agriculture and technology go hand in hand towards a prosperous and sustainable future.

**1.3.1 Purpose**

This project promises to unfold the immense possibilities and capabilities of IoT within the agricultural sector, presenting a new era for farming practice, from traditional to connected Agriculture, from instinct-driven decisions to data-driven decisions, and better yielding of crops, thus helping to buttress the global demand.

**1.3.2 Scope**

Traditional farming often relies on generalized resource allocation, leading to inefficiencies in water and fertilizer usage. The project aims to use IoT sensors to collect real-time data on soil moisture, temperature, and other relevant factors. By integrating this data with analytics, farmers can make precise decisions about irrigation and nutrient application based on the actual needs of crops. This approach minimizes resource wastage and ensures optimal growth conditions. Automated irrigation systems can be designed to respond to the specific moisture levels of the soil. IoT sensors placed in the field continuously monitor soil moisture content. When the moisture level falls below a certain threshold, the system triggers irrigation, delivering water precisely when and where it's needed. This targeted irrigation approach reduces water consumption, prevents overwatering, and maintains optimal soil conditions for plant growth. Sensors can be deployed to monitor environmental conditions that pests thrive in, such as temperature and humidity. Any deviations from normal conditions can trigger alerts to farmers, indicating the potential onset of pest infestations. By detecting pests at an early stage, farmers can implement timely interventions like organic pesticides, reducing crop damage and the need for excessive chemical treatments. The project contributes to environmental sustainability by promoting responsible resource management. By using IoT-driven data to allocate water and nutrients accurately, the project reduces the environmental impact associated with excessive water usage and chemical runoff. This approach aligns with sustainable farming practices, safeguarding soil quality and surrounding ecosystems. IoT-generated data is processed through analytics to provide farmers with valuable insights. Historical data combined with real-time information can offer trends, patterns, and predictive analyses. Farmers can receive recommendations on optimal planting times, irrigation schedules, and pest control strategies. These insights empower farmers to make informed decisions that maximize crop yield and minimize resource waste. Through a user-friendly mobile application or web interface, farmers can remotely monitor their fields. This feature grants farmers the flexibility to keep track of critical parameters even when they are not physically present on the farm. Remote monitoring enables prompt responses to unexpected changes and allows farmers to adjust irrigation and other factors remotely. The project design allows for scalability, accommodating the diverse needs of different farming operations. Farmers can expand the system to cover larger fields or adapt it for various types of crops. This scalability ensures that the benefits of IoT-enabled smart agriculture can be extended to farms of various sizes and requirements.

**1.3.3 Applicability**

1**. Precision Agriculture**: Through IoT-based systems, farmers can monitor the needs of individual fields and crops in real-time. This equips them with useful insights on when to water, when to plant, when to fertilize, and when to harvest, enabling exact resource application at the right time.

2. **Greenhouse Control Systems**: The IoT can be applied to monitor environmental conditions in greenhouses effectively. Changes in conditions are communicated in real-time, enabling farmers to optimize crop conditions and growth.

3. **Crop Management**: IoT technology can be used to predict the best time to plant, spray, and harvest crops based on weather forecasting and real-time field data.

4. **Irrigation Management**: Smart farming solutions enable controlled irrigation, ensuring water is conserved by providing specific amounts where necessary, reducing water waste and promoting healthier crop growth.

5**. Soil and Crop Monitoring**: Advanced IoT systems can use sensors to identify changes in climate conditions and soil health, which could influence crop growth. This data helps farmers make timely and crucial decisions.

**1.3 Achievements**

1. **Increased Crop Yield**: Due to the precise usage of water, fertilizers, and optimal cropping patterns provided by IoT systems, a noticeable increase in crop yield can be achieved.

2**. Resource Conservation**: Through smart irrigation systems, there has been significant conservation of water resources. Similarly, using specific quantities of fertilizers and pesticides as needed has led to reduced wastage of these resources.

3. **Reduction in Operational Costs**: Automated farming operations have helped minimize labour expenses. Also, by eliminating unnecessary usage of water, power, and fertilizers, the overall operational cost gets reduced.

4. **Enhanced Quality of Produce**: IoT systems enable watching the crop growth in real-time, thereby effectively controlling the quantity and quality of the yield. Early alerts on disease or pest infestation help to ensure the quality of the produce.

5. **Sustainability**: By employing waste management tactics, soil conservation techniques, and optimization of water resources, IoT in agriculture has significantly contributed towards sustainable farming and environmental preservation.

6. **Empowered Decision Making**: The accessibility to real-time data through IoT devices has enhanced the decision-making capacity of farmers, helping them to make well-informed and timely decisions leading to better results in crop yield and overall farm management.

**1.5 Organisation of Report**

• Chapter 2 will summarize the details of the technologies necessary to complete the project.

• In chapter 3 problem statement will be defined which will be divided into subproblems. Requirement specifications will describe the things in the system and the actions that can be done on these things.

• Chapter 4 describes desired features and operations in detail including screen layout, business rules, process diagrams, and other documentation

• Basic Modules: The students should follow the divide and conquer theory, so divide the overall problem into more manageable parts and develop each part or module separately.

• Data Design: Data design will consist of how data is organized, managed, and manipulated.

• Logic Diagrams: Define the systematical flow of the procedure that improves its comprehension and helps the programmer during implementation. e.g., Control Flow Chart, Process Diagrams, etc.

• We will also write about Test case design.

• Chapter 5 describes different types of 8testing like unit, integration, system testing, etc.

• Chapter 6 describes the User documentation which includes all screenshots of the project

• Chapter 7 describes the cost and benefit analysis

• Chapter 8 describes the significance of the system, its limitations of the system and also future enhancement.

# CHAPTER 2: Survey of Technologies

**2.1 Features of Front end**

Parts used in the project:

* ESP 8266
* ESP Camera
* Motion, Moisture, Temperature, Humidity Sensors
* Jumper wires
* USB Wires

**2.2 Features of Backend**

Blynk Software will be used to connect to the sensors and provide realtime live data to the farmers. This is an app which allows users to build their own functions based on the items they’re using. This operates on C++ language.

* User-Friendly App Builder
* Cross-Platform Compatibility
* Wide Range of Widgets
* IoT Hardware Integration

Fritzing/Proteus will be used to simulate the entire connection virtually.

**2.3 Justification of technologies used**

These parts provide the best functionality and efficiency required to function and get desired result and output. Blynk Software used is very easy to code and easy to access which provides a lot of functionalities required for the task. It has a great framework specifically aimed at allowing the user to have a good experience with their work. It has a user-friendly interface, cross-platform compatibility. It gives wide hardware support, cloud connectivity. It also allows for customization and data visualization.

# CHAPTER 3: Requirements and Analysis

**3.1 Problem Definition**

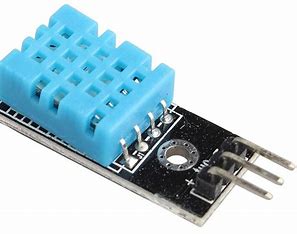
Traditional agricultural practices face inefficiencies in resource management, monitoring, and decision-making, resulting in suboptimal yield, increased resource consumption, and potential environmental harm. The lack of real-time data and automated control systems hinders farmers from making informed decisions about irrigation, fertilization, pest control, and overall crop health. This not only affects the profitability of farmers but also contributes to water wastage and chemical overuse, impacting the environment.

**3.2 Requirement Specification:**

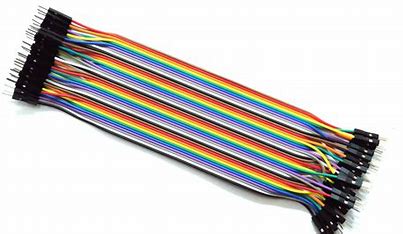
**3.2.1 Hardware Specification**

****

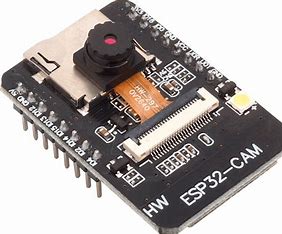
**ESP 8266:->** Is a low-cost Wi-Fi microchip with full TCP/IP stack and microcontroller capability.

**   **

**Motion, Moisture, Temperature, Humidity Sensors:->** These are devices that detect events or changes in their environment and send the information to other electronics.

****

**Jumper wires:->** Are used to establish connections between different points on a breadboard, circuit board, or electronic components.

****

**ESP32 Camera:->** An ESP camera, or ESP32 camera, is a type of Wi-Fi enabled smart camera that combines the functionality of a webcam with advanced computing and network connectivity capabilities. It can be used for a variety of purposes, including security surveillance, remote monitoring, and live streaming.

**3.2.2 Software Requirements**

**1. Blynk Software:->** Blynk is a popular Internet of Things (IoT) platform that allows users to create custom mobile applications to control and monitor IoT devices remotely. It's designed to simplify the process of connecting hardware (such as microcontrollers) with the cloud and mobile apps, enabling seamless interaction and control of smart devices. Blynk provides a user-friendly interface and a variety of features that make it a preferred choice for DIY enthusiasts, hobbyists, and developers entering the IoT space.

****

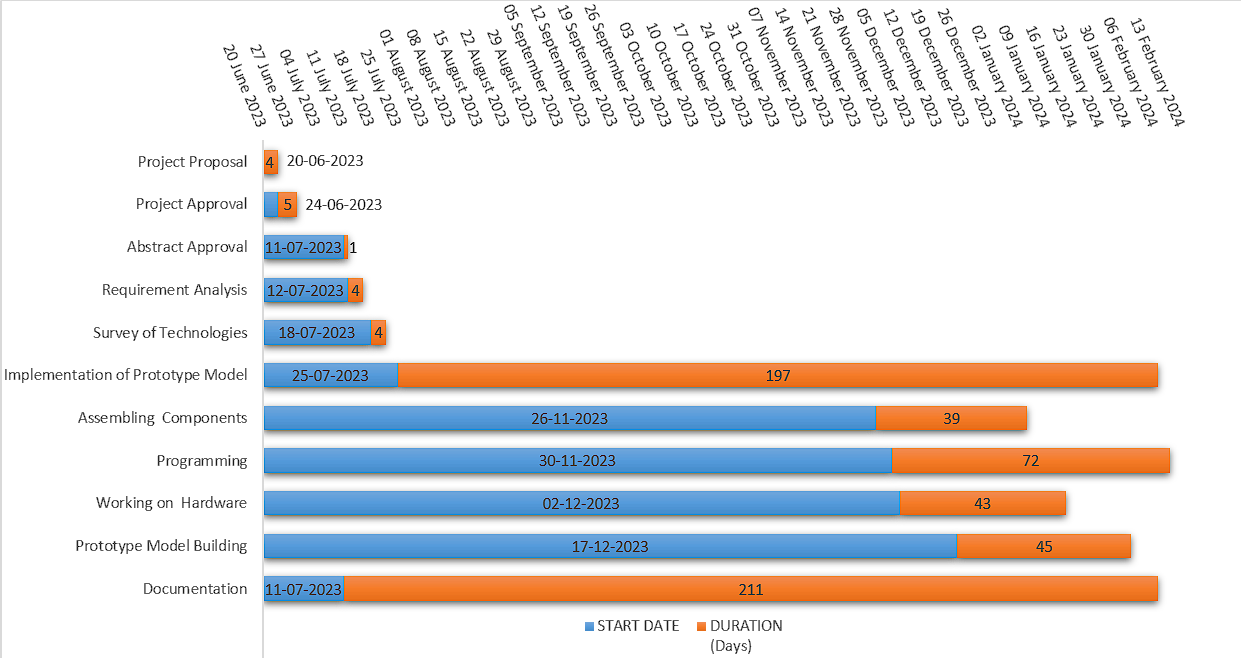
**2. Arduino IDE:** Arduino IDE, also known as Arduino Integrated Development Environment, is an open-source software used for programming Arduino-based electronic devices. It is designed to be user-friendly and intuitive, with a simple text-based interface that allows users to rapidly prototype and build interactive objects using Arduino boards. The IDE supports multiple programming languages, including C++, Python, and Ruby, and has built-in tools for testing and debugging code, monitoring serial communications, and managing libraries.



**3.3 Planning and Schedule**

|  |  |  |
| --- | --- | --- |
| **Task** | **Start date** | **End date** |
|  |  |  |
| **Project proposal** | **20/06/23** | **24/06/23** |
| **Project approval** | **24/06/23** | **29/06/23** |
| **Abstract approval** | **11/07/23** | **12/07/23** |
| **Requirement analysis** | **12/07/23** | **16/07/23** |
| **Survey of technologies** | **18/07/23** | **22/07/23** |
| **Implementation of prototype model** | **25/07/23** | **12/02/24** |
| **Assembling components** | **26/11/23** | **05/01/24** |
| **Programming** | **30/11/23** | **12/02/24** |
| **Working on hardware** | **02/12/23** | **15/01/24** |
| **Prototype model building** | **17/12/23** | **02/02/24** |
| **Documentation** | **11/07/23** | **12/02/24** |

**3.4 Gantt chart**

****

**3.5 Preliminary Product Description**

**This will be made on a big cardboard chart where all the part will be stuck on it and connected through wires to each other. The system will be connected to the device and will be coded in such a way that it can display data from sensors on the click of a button and give out notifications to the registered user.**

# CHAPTER 4: System Design

**4.1 Basic Modules**

Sensing Module: This includes a variety of sensors to monitor conditions like soil moisture, temperature, humidity, light intensity, and pH level. These sensors continuously gather relevant data from the environment.

Communication Module: This module enables data transmission between the sensors and the cloud-based platform. It typically adopts wireless communication technologies such as Wi-Fi

Automation/Actuation Module: This includes actuators that can interact with physical farming systems (i.e., irrigation systems. They perform actions based on the decisions made in the system.

**4.2 Algorithm Modules**

1. Data Collection Algorithm:

a. Initialize sensor parameters (like soil moisture, temperature, humidity, etc.)

b. Collect the data from the sensors at regular intervals.

c. Send the collected data to the cloud for processing and analysis.

2. Data Processing Algorithm:

a. Extract data from cloud storage.

b. Clean the extracted data. Handle missing values, outliers, etc.

c. Analyse the data to obtain valuable insights. (e.g., Predictive analytics to predict future outcomes)

3. Decision Making Algorithm:

a. Based on insights obtained from data processing, make decisions. (e.g., if soil moisture level is below a certain threshold, irrigation might be needed).

4. Automation/Actuation Algorithm:

a. Execute commands based on decisions made. (e.g., turn on the irrigation system).

**4.3 UML Diagrams**

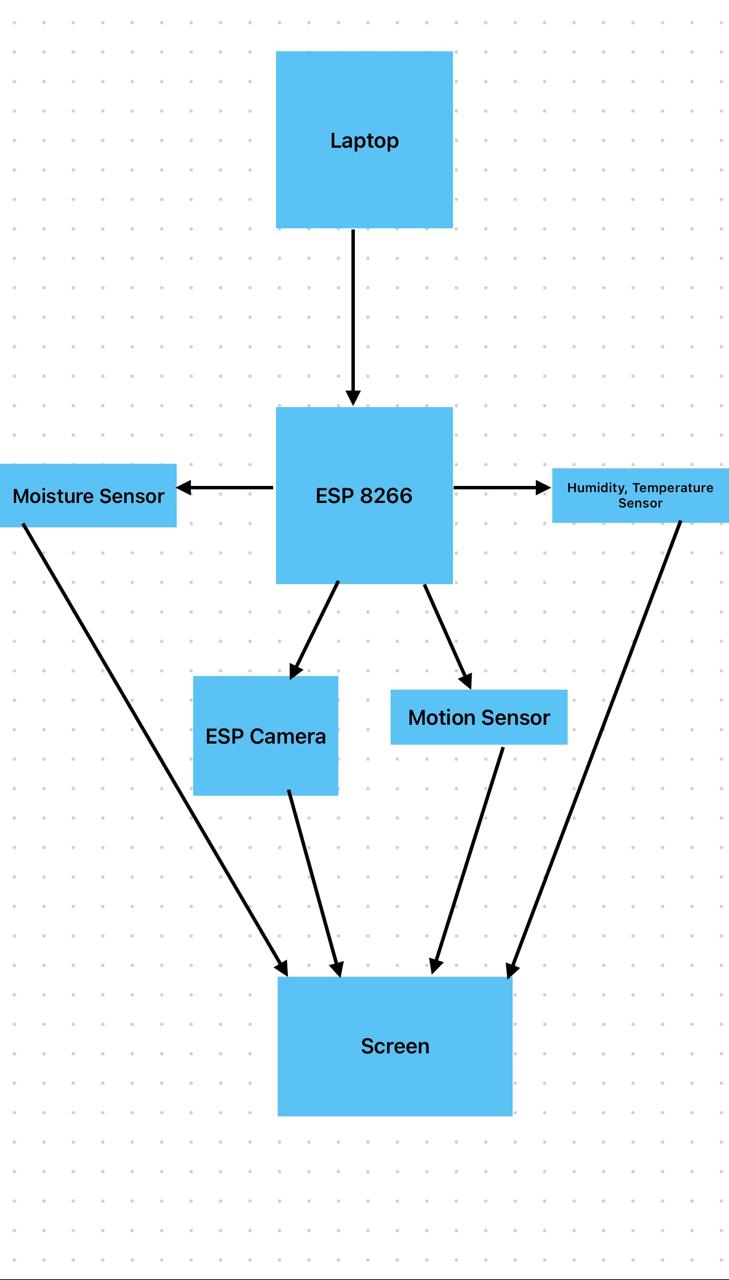
1. What is UML?

UML stands for Unified Modelling Language.   
• It is a general-purpose modelling language.   
• It is popular for its diagrammatic notations.   
• It is for visualizing, specifying, constructing and documenting the components of software and non-software systems.   
• The model is useless, unless its purpose is depicted properly and clearly.   
• Hence learning notations should be emphasized from the very beginning.   
• Different notations are available for things and relationships.   
• UML diagrams are made using the notations of things and relationships.   
• Extensibility is an important feature which makes UML more powerful and flexible.

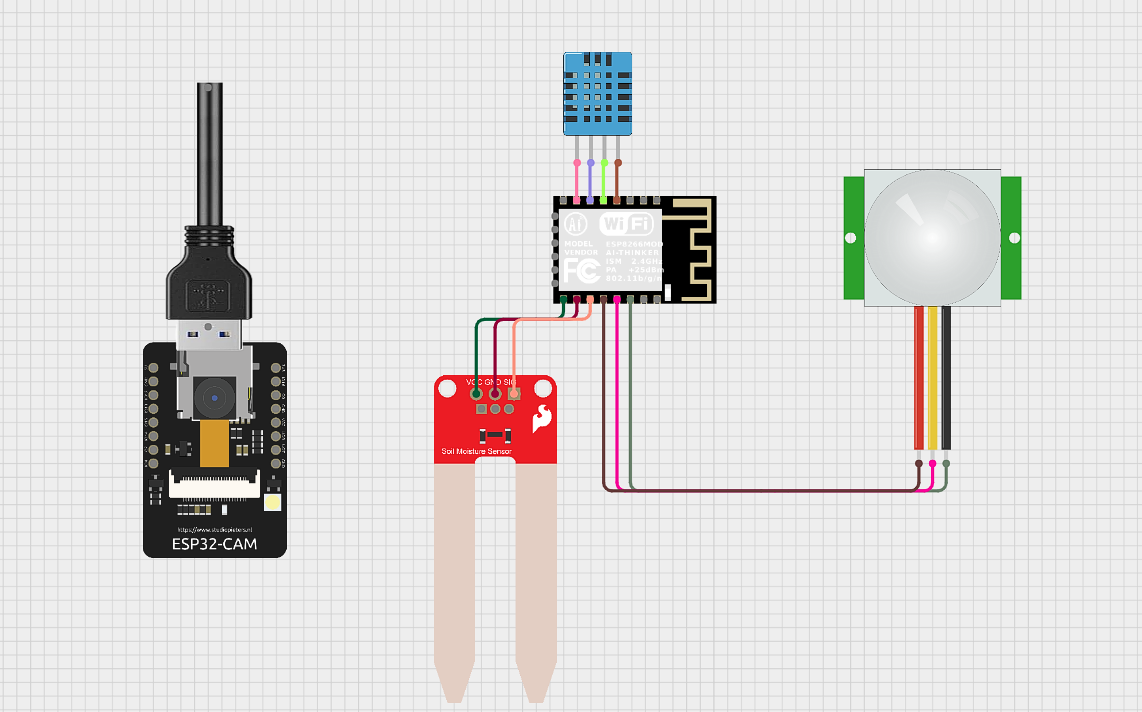
2. Why do we use UML?   
  
• It can be used as a way to visualize a project before it takes place or as documentation for a project afterward.   
• Can provide new ideas for how the individual needs to collaborate to achieve the goal.   
  
3. What are the types of UML diagrams used in this project?   
  
**Block Diagram**: Is a visual representation that depicts the major components or functional blocks of a system and their interactions.

**Circuit Diagram**: A circuit diagram is a visual representation of an electrical circuit. It shows the components, such as resistors, capacitors, and inductors, and how they are connected by wires. Circuit diagrams are used to design, analyze, and troubleshoot electrical circuits. They can be drawn using a variety of tools, including pencil and paper, computer software, or specialized circuit design programs.

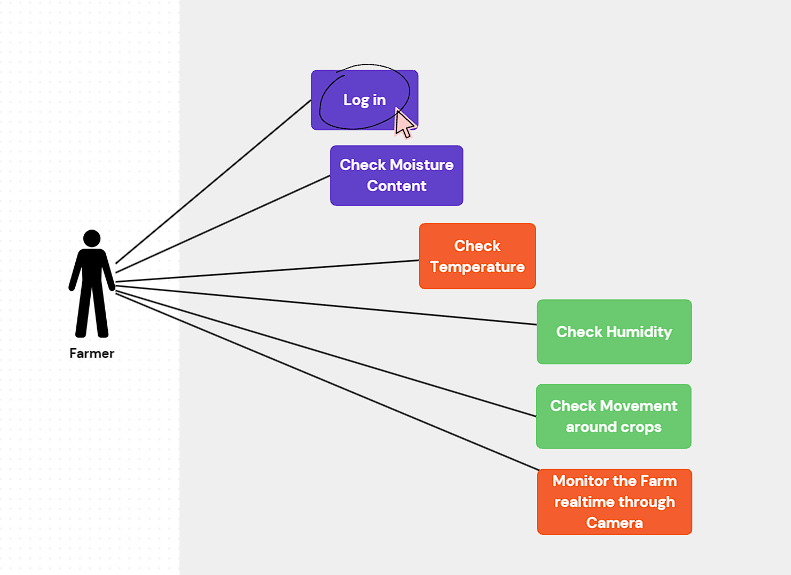
**Use Case Diagram**: A use case diagram is a graphical representation of the interactions between a system and its users. It depicts the actions that users can take on the system and the responses that the system provides. The diagram helps in understanding the functionality of the system and its user interface.



Block Diagram



Circuit Diagram



Use Case Diagram

**4.4 Security Issues**

**Data Privacy**: Since sensitive data such as crop yield, usage of resources, farming practices, and more are continually collected, the potential for data breaches is a significant concern. Unauthorized access could lead to misuse of this information

**Data Integrity:** The risk of intentional alteration or corruption of the data, either while in storage or during transmission, poses a threat to the decisions derived from the data, potentially leading to harmful outcomes.

**Device Tampering**: Physical IoT devices placed in the field could be vulnerable to tampering or vandalism, leading to skewed data or total loss of data.

**Unauthorized System Access**: Without proper access controls and authentication protocols, unauthorized individuals could gain access to the system and manipulate the controls wrongly. For instance, they could trigger the irrigation system to flood the fields or disable it completely.

**Network Security**: The wireless network used for communication between devices can be vulnerable to hacking attempts like man-in-the-middle attacks or denial of services (DOS) attacks.

# CHAPTER 5: Implementation and Testing

**5.1 Implementation Approaches**

A planned approach for a successful project is of utmost importance as the saying goes like "Plans are of little importance, but planning is essential." The planning approach used in this project is ITERATIVE MODEL.

Iterative Model:

• Iterative development is a technique for developing software that reduces risk by breaking a large body of work into smaller, more manageable units of work.

• Iterative development tends to be an organic process, subject to constant change and evolution throughout the duration of the project lifecycle.

• It is not possible to know everything about a software system in the early stages of the development project.

• Change can be imposed by many factors; under these circumstances embracing changes is essential because satisfying the requirements is always the fundamental goal of the project.

• Iterative software development approach is structured around the activities of distinct phases of development efforts.

• These phases basically include the Inception, Elaboration, Construction and Transition phases of the software development project. These phases occur consecutively and describe the activities that are appropriate to the development efforts at the point in the project.

• Iterative development slices the deliverable business value (System Functionality) into iterations.

• In each iteration a slice of functionality is delivered to cross-discipline work, starting from the model/requirement through to the testing/development.

**The Stages of the Iterative Model:**

1. Planning: The primary objectives of the planning phase are to identify the scope of the new system, ensure that the project is feasible, develop a schedule and allocate resources and the budget.

2. Requirement Analysis & Development: All possible requirements of the system to be developed and captured in this phase. Requirements are set of functionalities and constraints that the end – user expects from the system. The requirements are gathered from the end – user by consultation, these requirements are analyzed for their validity and the possibility of incorporating the requirements in the system to be developed is also studied. Finally, a Requirement Specification document is created which serves the purpose of guideline for the next phase of the model.

3. System & Software Design: Before starting the actual coding, it is very important to understand what we are going to create and what it should look like. The requirement specifications from first phase are studied in this phase and system design is prepared. System design helps in specifying hardware and system requirements and also helps in defining overall system architecture. The system design specifications serve as input for the next phase of the model.

4. Implementation and Unit Testing: On receiving system design document, the work is divided in modules/ units and actual coding is started. The system is first developed in small programs called units, which are integrated in the next phase. Each unit developed and tested for its functionality, referred to as “Unit Testing”. Unit testing mainly verifies if the units meet their specifications.

5. Integration and System Testing: As specified above, the system is divided in units which are developed and tested for their functionalities. These units are integrated into a complete system during Integration Phase and tested to check whether all modules/units co-ordinate between each other and the system behaves as per the specifications. After successfully testing the software it is delivered to the customer.

6. Operations & Maintenance: This phase is the longest phase of this model. It is never ending phase in which problems with the system developed (which were not found during the development life cycle) come up after its practical use starts. So the issues related to the system are solved after development of the system. These problems arise time to time and needs to be solved. Hence this process is referred as Maintenance.

**5.2 Code Details and Code Efficiency**

Code Efficiency

Optimization is a program transformation technique, which tries to improve the code by making it consume less resource (i.e. CPU, Memory) and deliver high speed. In optimization, high-level general programming constructs are replaced by very efficient low-level programming codes. A code optimizing process must follow the rules given below:

• The output code must not, in any way, change the meaning of the program.

• Optimization should increase the speed of the program and if possible, the program should demand less number of resources.

• Optimization should itself be fast and should not delay the overall compiling process.

• Efforts for an optimized code can be made at various levels of compiling the process.

• At the beginning, users can change/rearrange the code or use better algorithms to write the code.

• After generating intermediate code, the compiler can modify the intermediate code by address calculations and improving loops.

• While producing the target machine code, the compiler can make use of memory hierarchy and CPU registers.

• Optimization can be categorized broadly into two types: machine independent and machine dependent.

Machine-independent Optimization

• In this optimization, the compiler takes in the intermediate code and transforms a part of the code that does not involve any CPU registers and/or absolute memory locations.

Machine-dependent optimization

• This type of optimization is done after the target code has been generated and when the code is transformed according to the target machine architecture. It involves CPU registers and may have absolute memory references rather than relative references. Machine-dependent optimizers put efforts to take maximum advantage of memory hierarchy.

Basic block identification

We may use the following algorithm to find the basic blocks in a program:

• Search header statements of all the basic blocks from where a basic block starts: First statement of a program.

• Statements that are target of any branch (conditional/unconditional).

• Statements that follow any branch statement.

• Header statements and the statements following them form a basic block.

• A basic block does not include any header statement of any other basic block.

Dead-code Elimination

Dead code is one or more than one code statements, which are:

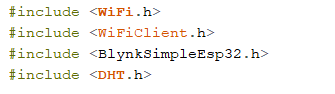
• Either never executed or unreachable

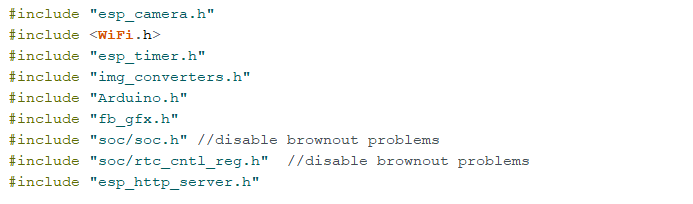
• Or if executed, their output is never used.

• Thus, dead code plays no role in any program operation and therefore it can simply be eliminated

**CODE**

Packages used





**5.3 Testing Approach**

**5.3.1 Unit Testing**

What is Unit Testing?

• Unit testing, a testing technique using which individual modules are tested to determine if there are any issues by the developer himself. It is concerned with functional correctness of the standalone modules.

• The main aim is to isolate each unit of the system to identify, analyse and fix the defects.

• There are different methods that can be used for testing. This chapter briefly describes the methods available.

Black-Box Testing

• The technique of testing without having any knowledge of the interior workings of the application is called black-box testing. The tester is oblivious to the system architecture and does not have access to the source code. Typically, while performing a black-box test, a tester will interact with the system by providing inputs and examining outputs without knowing how and where the inputs are worked upon.

White-Box Testing (Adopted Approach)

• White-box testing is the detailed investigation of internal logic and structure of the code. Whitebox testing is also called glass testing or open-box testing. In order to perform whitebox testing on an component, a tester needs to know the internal workings of the code.

• The tester needs to have a look inside the source code and find out which unit/chunk of the code is behaving inappropriately.

Grey-Box Testing

• Grey-box testing is a technique to test the developed software with having a limited knowledge of the internal workings of an application. Mastering the domain of a system always gives the tester an edge over someone with limited domain knowledge. Unlike black-box testing, where the tester only tests the systems working; in grey-box testing, the tester has access to architecture documents. Having this knowledge, a tester can prepare better test data and test scenarios while making a test plan.

**5.3.2 Integrated Testing**

What is Integration Testing?

• Upon completion of unit testing, the units or modules are to be integrated which gives raise to integration testing. The purpose of integration testing is to verify the functional, performance, and reliability between the modules that are integrated.

• Integration Strategies:

• Big-Bang Integration

• Top Down Integration

• Bottom Up Integration

• Hybrid Integration

**5.3.3 Beta Testing**

What is Beta Testing?

• Beta testing also known as user testing takes place at the end users site by the end users to validate the usability, functionality, compatibility, and reliability testing.

• Beta testing adds value to the software development life cycle as it allows the "real" customer an opportunity to provide inputs into the design, functionality, and usability of a product. These inputs are not only critical to the success of the product but also an investment into future products when the gathered data is managed effectively.

• There are number of factors that depends on the success of beta testing:

o Test Cost

o Number of Test Participants

o Shipping

o Duration of Test

o Demographic coverage

**5.4 Modification and Improvements**

• This system has modified the traditional and most used clock system to a new way of display and also a which will be apt for the aesthetics and interiors for your beautiful homes.

• Till now either you would get a clock or Google Home separately, i.e. you need to pay for it individually but now this isn’t needed.

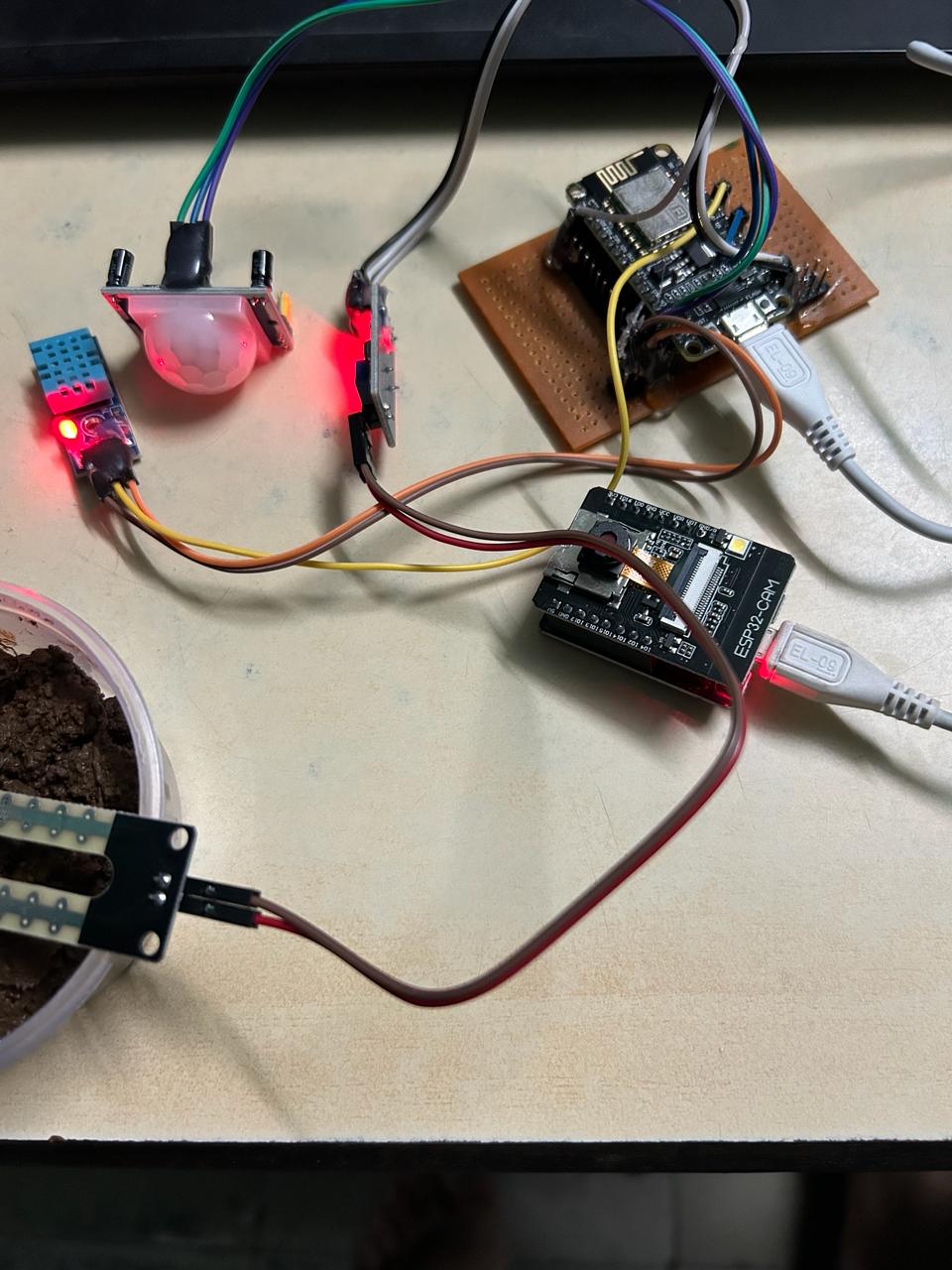
• We can buy a word clock rather than a traditional clock which can be customised as per your convenience and also comes with it the facility of Google Home which enables u set alarms and do various other things easily.

**5.5 Test Cases**

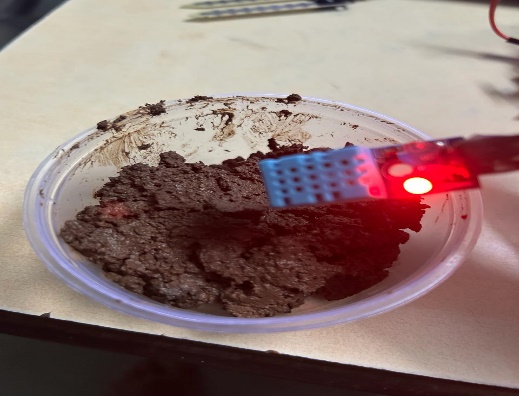
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test No | Description | Expected Result | Actual Result | Remark |
| TC 1 | Connection check for ESP 8266 | Connection working | Connection working | Pass |
| TC 2 | Connection check for Moisture Sensor | Connection working | Connection working | Pass |
| TC 3 | Connection check for Temperature, Humidity Sensor | Connection working | Connection working | Pass |
| TC 4 | Connection check for Motion Sensor | Connection working | Connection working | Pass |
| TC 5 | Connection check for Camera | Connection working | Connection working | Pass |
| TC 6 | Connection to Blynk IOT Platform | Connection working | Connection working | Pass |
| TC 7 | Check Output from Moisture Sensor | Giving output | Giving output | Pass |
| TC 8 | Check Output from Temperature, Humidity Sensor | Giving output | Giving output | Pass |
| TC 9 | Check Output from Motion Sensor | Giving output | Giving output | Pass |
| TC 10 | Check Output from Camera | Giving output | Giving output | Pass |

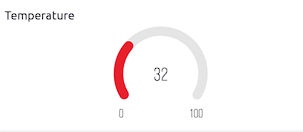
# CHAPTER 6: User Documentation

**Step 1: Connect all the connections to the device. Make sure all of them are working by checking the led light on them.**

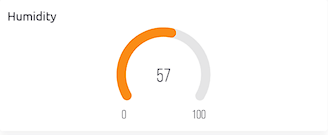


**Step 2: Try to check if the sensors are working by taking in input and out put in the form of readings.**



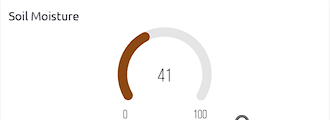
****

DHT11 Sensor working for Temperature

****

DHT11 Sensor working for Humidity



****

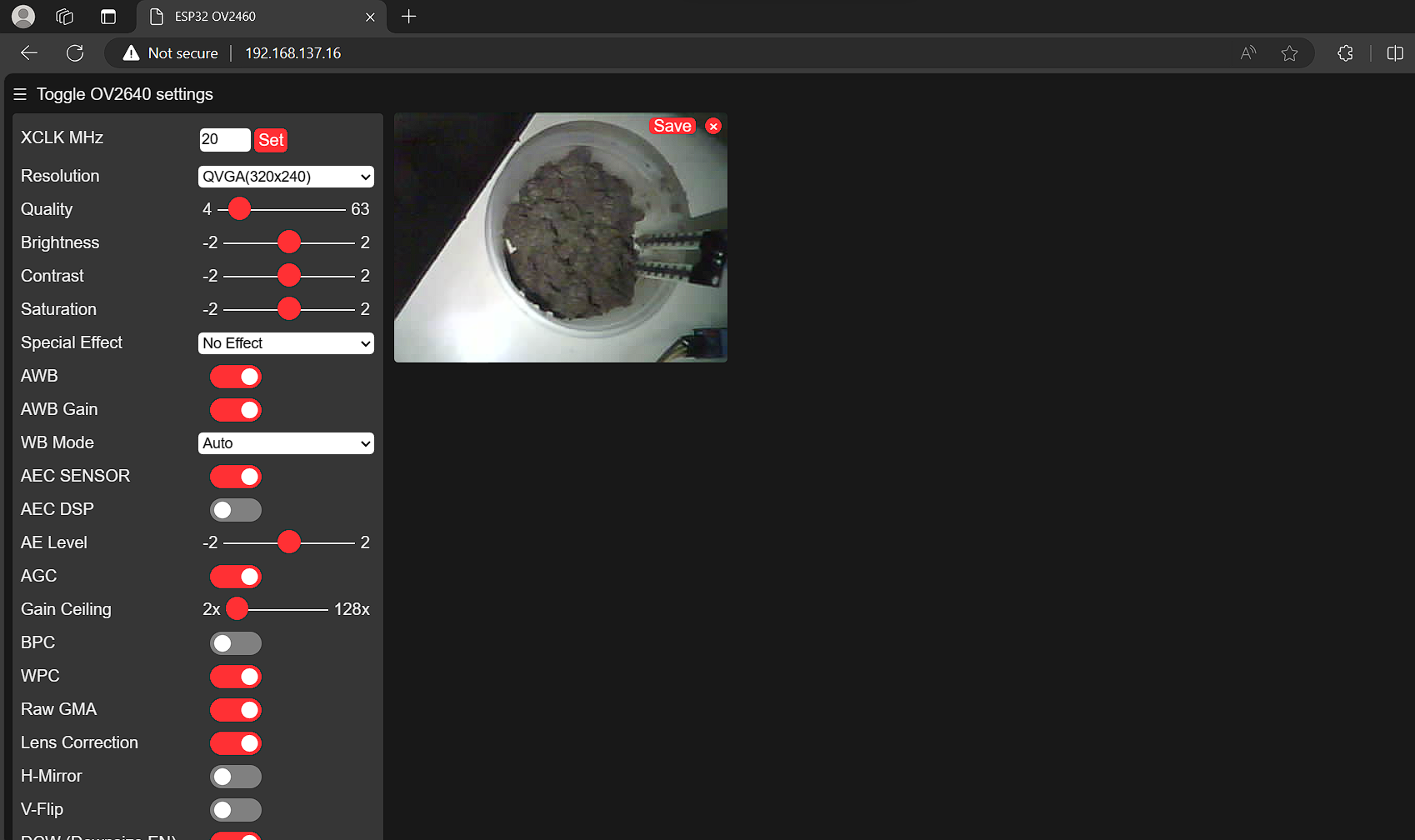
Soil Moisture Sensor working



****

PIR Motion Sensor working

**Step 3: Check if the camera is working live.**



# CHAPTER 7: Cost Benefit Analysis

The term “Cost Benefit Analysis” refers to how the software will prove to be beneficial than the existing system used in the organization implementing it. Cost Benefit Analysis involves the balance of project expenditure and returns from its implementations. It must be noted that any hardware implementation involves initial investments and benefits are usually long term. System implementation time may vary for different organizations and it may also take some time for the employees of the different organizations to adjust with the system. A thorough analysis of the system was undertaken by keeping in the cost benefit ratio, details of which are as follows:

**7.1 Cost Evaluation**

**• Hardware Cost**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr no** | **Components** | **Quantity** | **Price** | **Total Price** |
| **1** | **PCB Board** | **1** | **100** | **100** |
| **2** | **Jumper Wires** | **Multiple** | **20** | **20** |
| **3** | **ESP 8266** | **1** | **600** | **600** |
| **4** | **Moisture Sensor** | **1** | **100** | **100** |
| **5** | **Motion Sensor** | **1** | **200** | **200** |
| **6** | **ESP 32 Camera** | **1** | **500** | **500** |
| **7** | **Temperature, Humidity Sensor** | **1** | **150** | **150** |
| **8** | **Glue Gun** | **1** | **130** | **130** |

**Total: 1800**

**ANALYSIS, PROTOTYPING AND CODING COST:**

Analysis, Prototyping and Coding = No. of people involved in project \* Charges per day \* No. of days

No. of people involved in project = 1

Charges per day = 70

No. of days = 215

Analysis, Prototyping and Coding = 1 \* 70 \* 197

**= Rs. 15,050**

**TESTING AND MAINTENANCE COST:**

Testing and Maintenance Cost = No. of hours \* Charges per day \* No. of days

No. of hours = 3

Charges per day = 70

No. of days = 30

Testing and Maintenance Cost = 3 \* 70 \* 30 **= Rs. 6,300**

**Overall Cost = Rs. 23,150**

# CHAPTER 8: Conclusion

**8.1 Significance of the System**

1. **Precision Agriculture:** The integration of IoT sensors allows for precise monitoring and management of environmental parameters such as soil moisture, temperature, and humidity. This enables farmers to optimize resource utilization by providing real-time insights into the specific needs of crops, leading to improved yields and resource efficiency.
2. **Optimized Irrigation:** Moisture sensors provide accurate data on soil moisture levels, allowing farmers to implement precision irrigation techniques. By only irrigating when necessary and in the right amounts, water usage can be significantly reduced, mitigating water wastage and conserving valuable resources.
3. **Climate Monitoring:** Temperature and humidity sensors play a crucial role in monitoring microclimates within agricultural fields. This data is essential for assessing the impact of weather conditions on crop growth and identifying potential risks such as frost or heat stress, enabling proactive measures to protect crops and optimize growth conditions.
4. **Pest and Disease Management:** Motion sensors can detect the presence of pests or unauthorized intruders in the fields. Combined with camera feeds, they provide real-time surveillance, allowing farmers to identify pest infestations or unusual activities and take prompt action to prevent crop damage or theft.
5. **Data-Driven Decision Making:** The IoT system generates a wealth of data regarding environmental conditions, crop health, and field activities. Analyzing this data using advanced analytics techniques enables farmers to make informed decisions regarding crop management strategies, resource allocation, and risk mitigation, ultimately improving productivity and profitability.
6. **Remote Monitoring and Management:** The IoT-enabled system allows farmers to remotely monitor and manage their agricultural operations from anywhere, using web-based dashboards or mobile applications. This capability provides flexibility and convenience, allowing farmers to stay connected to their fields and respond promptly to emerging challenges or opportunities.
7. **Sustainability and Environmental Impact:** By optimizing resource usage, minimizing inputs, and adopting sustainable farming practices guided by IoT data insights, the system contributes to environmental conservation and reduces the ecological footprint of agriculture. This aligns with global sustainability goals and supports the development of eco-friendly farming practices.
8. **Scalability and Adaptability:** The modular nature of IoT systems allows for scalability and adaptability to different farm sizes, crop types, and geographical locations. Whether deployed on small-scale family farms or large commercial operations, the system can be customized to meet specific needs and requirements, making it accessible to a wide range of farmers

**8.2 Limitations of the System**

1. **Reliability and Connectivity Issues:** The reliability of IoT sensors and connectivity infrastructure can be affected by factors such as network coverage, signal interference, and hardware malfunctions. In remote or rural areas with poor network connectivity, maintaining consistent data transmission and communication with the central monitoring system may be difficult.
2. **Sensor Calibration and Maintenance:** Sensors used in the system, such as moisture sensors, temperature sensors, and humidity sensors, require periodic calibration and maintenance to ensure accurate readings. Failure to calibrate sensors properly or address maintenance issues can lead to erroneous data and inaccurate decision-making.
3. **Data Security and Privacy Concerns:** Collecting and transmitting sensitive agricultural data over IoT networks raise concerns about data security and privacy. Unauthorized access, data breaches, or cyber-attacks targeting the IoT infrastructure could compromise farmers' confidential information, intellectual property, or operational security**.**

**8.3 Future Scope of the Project**

1. **Advanced Sensor Technology:** Future advancements in sensor technology, including improvements in accuracy, reliability, and energy efficiency, will enable the development of more sophisticated IoT devices for agriculture. Integrating cutting-edge sensors capable of detecting additional parameters such as soil nutrient levels, pest infestations, and crop health will provide farmers with more comprehensive insights into their agricultural operations.
2. **Data Analytics and Artificial Intelligence:** The integration of advanced data analytics techniques and artificial intelligence (AI) algorithms will enable more intelligent and predictive decision-making in agriculture. Machine learning models trained on large datasets collected from IoT sensors can help farmers optimize resource allocation, predict crop yields, identify anomalies, and automate farming tasks.
3. **Precision Agriculture:** The adoption of precision agriculture techniques facilitated by IoT technology will continue to increase. Precision agriculture involves the precise management of agricultural inputs, such as water, fertilizers, and pesticides, based on real-time data and spatial variability within fields. IoT-enabled precision agriculture solutions will lead to higher crop yields, improved resource efficiency, and reduced environmental impact.
4. **Integration with Remote Sensing and Satellite Imagery:** Integrating IoT data with remote sensing technologies and satellite imagery will provide farmers with a more comprehensive view of their agricultural landscapes. Combining ground-level sensor data with aerial and satellite-based observations will enable farmers to monitor larger areas, detect trends, and assess environmental conditions with greater accuracy.

# CHAPTER 9: References

**REFERENCES**

* <https://google.github.io/mediapipe/solutions/holistic.html>
* [https://www.youtube.com/watch?v=doDUihpj6ro&list=PLLElFqlosYJExYrIE7TBmr4fsrHLDWSm&index=23](https://www.youtube.com/watch?v=doDUihpj6ro&list=PLLElFqlosYJE-xYrIE7TBmr4fsrHLDWSm&index=23)
* <https://academic.oup.com/jdsde/article/11/1/94/410770>
* <https://www.irjet.net/archives/V6/i3/IRJET-V6I3639.pdf>
* <https://www.ijsdr.org/papers/IJSDR1910017.pdf>
* <https://www.ripublication.com/ijaerspl2018/ijaerv13n4spl_01.pdf>
* <https://ieeexplore.ieee.org/document/8191869>
* <http://cse.anits.edu.in/projects/projects1920C6.pdf>
* <https://ijirt.org/master/publishedpaper/IJIRT154130_PAPER.pdf>
* <https://www.guru99.com/uml-class-diagram.html>
* [https://www.geeksforgeeks.org/unified](https://www.geeksforgeeks.org/unified-modeling-language-uml-sequence-diagrams/)
* <https://www.guru99.com/uml-object-diagram.html>
* <https://www.guru99.com/state-transition-testing.html>
* [https://www.geeksforgeeks.org/software-engineering-rapid-application-developmentmodel-rad/](https://www.geeksforgeeks.org/software-engineering-rapid-application-development-model-rad/)
* [https://www.techopedia.com/definition/27151/codeefficiency#:~:text=Code%20efficiency%20is%20a%20broad,element%20in%20ensuring %20high%20performance.](https://www.techopedia.com/definition/27151/code-efficiency#:~:text=Code%20efficiency%20is%20a%20broad,element%20in%20ensuring%20high%20performance)
* <https://www.geeksforgeeks.org/unit-testing-software-testing/>
* <https://www.geeksforgeeks.org/software-engineering-integration-testing/>● <https://www.geeksforgeeks.org/beta-testing-software-testing/>