

**AUTOMETIC IRRIGATION AND MONITORING SYSTEM FOR BALCONY  
AND ROOFTOP PLANT USING IOT “KRISHIBOT”**

**BY**

**SAIKAT MAZUMDER**

**ID: 232-25-036**

This Report Presented in Partial Fulfillment of the Requirements for the  
Degree of MSc in Computer Science and Engineering

Supervised By

**Dr. Arif Mahmud**

Associate Professor

Department of CSE

Daffodil International University

Co-Supervised By

**Abdus Sattar**

Associate Professor

Department of CSE

Daffodil International University



**DAFFODIL INTERNATIONAL UNIVERSITY**

**DHAKA, BANGLADESH**

**MAY 2025**

## APPROVAL

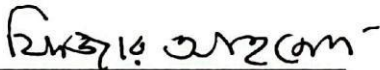
This Project/Thesis titled “Automatic Irrigation And Monitoring System For Balcony And Rooftop Plant Using Lot Krishibot”, submitted by SAIKAT MAZUMDER, ID No: 232-25-036 to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of M.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 24-05-2025.

### BOARD OF EXAMINERS



**Dr. S.M Aminul Haque**  
Professor and Associate Head  
Department of Computer Science and Engineering  
Faculty of Science & Information Technology  
Daffodil International University

**Chairman**



**Dr. Fizar Ahmed**  
Associate Professor  
Department of Computer Science and Engineering  
Faculty of Science & Information Technology  
Daffodil International University

**Internal Examiner**



**Dr. Md Alamgir Kabir**  
Assistant Professor  
Department of Computer Science and Engineering  
Faculty of Science & Information Technology  
Daffodil International University

**Internal Examiner**



**Nazibur Rahman**  
Technical Lead, Database Administrator  
Wipro, Telenor - Grameen Phone Account  
Dhaka, Bangladesh

**External Examiner**

## DECLARATION

We hereby declare that this project has been done by us under the supervision **Dr. Arif Mahmud, Associate Professor & Director of MSc in MIS ,Department of CSE** of Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for the award of any degree or diploma.

**Supervised by:**



**Dr. Arif Mahmud**  
Associate Professor  
Department of CSE  
Daffodil International University

**Co-Supervised by:**



**Abdus Sattar**  
Assistant Professor  
Department of CSE  
Daffodil International University

**Submitted By:**



**Saikat Mazumder**  
ID: 232-25-036  
Department of CSE  
Daffodil International University

## ACKNOWLEDGEMENT

First and foremost, I express my deepest gratitude to Almighty **God** for providing me with the strength, knowledge, and determination to complete this project successfully.

I extend my heartfelt thanks to my esteemed supervisor, **Dr. Arif Mahmud**, Associate Professor and Director of MSc in MIS, Department of CSE, Daffodil International University, for his continuous guidance, invaluable advice, and constructive feedback throughout this journey. His expertise in the field of IoT and tireless support played a pivotal role in the successful completion of this thesis.

I am also profoundly grateful to my co-supervisor, **Abdus Sattar**, Assistant Professor and Coordinator MSc., for his encouragement and technical assistance. Special thanks to my peers and the faculty members of the Department of CSE for their cooperation and insightful discussions during this thesis.

I would like to express my gratitude to my peers and course mates, whose collaboration, feedback, and camaraderie added value to this thesis. Special thanks to my family for their unconditional love, patience, and moral support during the challenging phases of this work. Finally, this accomplishment would not have been possible without the encouragement and prayers of everyone who supported me, directly or indirectly. Thank you all for being part of this journey.

## ABSTRACT

Our smart device “Krishibot” is an IoT product installed in your house to monitor your plants 24/7. The device is very easy to carry around, because it is so lightweight. The main idea of the thesis is to create and set up a smart garden system using IoT and the Blynk application. Multiple sensors and a microcontroller join forces to monitor the soil’s moisture, temperature and humidity, enabling the system to give plants the needed water at the correct moment. The system provides both types of control, with a remote option offered through Blynk. The area is secured with four key components: the ESP8266 NodeMCU, DHT11 humidity and temperature sensors, a soil moisture sensor and a rain detection sensor. It will display notifications on the user’s smartphone if the soil is too dry or wet or rain is detected. Another feature is a PIR motion sensor, helping to watch for unwanted animals in the garden, so rooftop and veranda plants aren’t harmed. Since it can be moved, quickly installed and used remotely, this system appeals to those starting home gardens as well as large green businesses. These results show that the system manages water well, reduces the need for manual help and supports more environmentally friendly urban gardening. In this part, you highlight your thesis by focusing on the main issue, how it was addressed technically and what broader changes the system introduced.

## TABLE OF CONTENTS

CONTENTS	PAGE
Board of examiners	ii
Declaration	iii
Acknowledgements	iv
Abstract	v
<b>CHAPTER</b>	
<b>CHAPTER 1: INTRODUCTION</b>	<b>1-5</b>
1.1 Inroductions	1
1.2 Motivation	2
1.3 Project Objective	2
1.4 Research Question	3
1.5 Expected Outcome	4
1.6 Report Layout	5
<b>Chapter 2: Litareture Review</b>	<b>6-15</b>
2.1 Preliminaries	6
2.2 Related Works	7
2.3 Comparative Analysis	11
2.4 Scope of the Problem	11
2.5 Challenges	14
<b>Chapter 3: RESEARCH METHOLOGY</b>	<b>16-27</b>
3.1 Business Process Model	16
3.2 Use-Case Modelling & Descriptions	17
3.3 Requirement Collection & Analysis	18
3.3.1 ESP 8266 Node MCU v3	19
3.3.2 DHT11 Humidity Sensor	21
3.3.2 Breadboard	22
3.3.3 Soil Moisture Sensor	23
3.3.4 Jumper Wire	24
3.3.5 .96 OLED Display	25
3.3.6 5V Relay Module	26
3.3.7 DC Motor	27
<b>CHAPTER 4: DESIGN SPECIFICATION</b>	<b>28-33</b>

4.1 Front-end Design	28
4.2 Back-end Design	29
4.3 Interaction Design and User Experience (UX)	30
4.4 Implementation Requirement	32
<b>CHAPTER 5: IMPLEMENTATION AND TESTING</b>	<b>34-38</b>
5.1 Implementation of Database	34
5.2 Testing Implantation	35
5.3 Final Output	36
<b>CHAPTER 6 : IMPACT ON SOCIETY</b>	<b>38</b>
6.1 Impact on Society	38
6.2 Impact on Environment	38
<b>CHAPTER 7 : CONCLUSION AND FUTURE SCOPE</b>	<b>39-42</b>
7.1 Consultation	39
7.2 Scope for Further Development	40
<b>REFERENCES</b>	<b>42-44</b>

## LIST OF FIGURES

<b>FIGURE</b>	<b>PAGE NO</b>
Figure 3.1: Business Process Model	16
Figure 3.2: Use Case Diagram	17
Figure 3.3.1: ESP 8266 node mcu v3	19
Figure 3.3.2: DHT11 Humidity Sensor	21
Figure 3.3.3: Breadboard	22
Figure 3.3.4: Soil Moisture Sensor	23
Figure 3.3.5: Jumper wire	24
Figure 3.3.6: .96 OLED Display	25
Figure 3.3.7: 2 Channel 5V relay Module	26
Figure 3.3.8: DC Motor	27
Figure 4.1.1: Front-end Design for Hardware sections	28
Figure 4.1.2: Front-end design for software	29
Figure 4.2: Back-end design process	29
Figure 4.3.1: Interaction design	30
Figure 4.3.2: Interaction Design	31
Figure 5.1: Implementation of Database	34

**LIST OF TABLE**

<b>Table</b>	
Table 1: Comparative Analysis	11
Table 2: Test Result	35

# Chapter 1

## Introduction

### 1.1 Introductions

People living in the city have started showing interest in rooftop gardens. But they don't have time to take care of the gardens and the trees. So, we approach "Krishibot" which is an IoT-based smart device with plant monitoring system automation. The system must be able to measure soil moisture level control for plants. IoT means the Internet of Things. The concept of the Internet of things is nothing new. It is a technology that can be used by a mobile device to monitor the activities of any device. IoT is a type of network technology that detects the data from a variety of sensors to ultimately have every source of information connected to the Internet to determine its use in a dialog. The android application is a general applications that is based on menu options in version four. The information included includes temperature, motor status, dampness, and moisture status. Because of urban expansion, homeowners are finding new ways to incorporate plants into their home environment. This is critical because typical gardens are less common in urban communities, and rooftop and veranda gardening is becoming more common. However, this still maintains the challenge of caring for the plants on a consistent basis when their owners are often busy and away. So if they fail to water for a length of time, many of those plants may die, but IoT allows them to easily manage them from afar. For IoT, devices can be connected to the internet and communicate with each other using the information share with their surroundings. The aim of the project is to develop an IoT based Smart Plant Irrigation and Monitoring System by making use of the Blynk application for remote monitoring backup, CI, CC. The system will make use of sensors to measure the soil moisture, and temperature and humidity to determine when and how much water the plant needs. In addition, the owners can use the system either manually or automatically to maintain the plants.

## **1.2 Motivation**

The inspiration for this project comes from the rising popularity of urban gardening and the consequent barriers. We can see how many urban dwellers have adopted tiny potted plants on their balconies and porches, and have even established rooftop gardens, but upkeep can be laborious. For the average urban resident, maintaining the plants becomes more of a nuisance rather than a hobby, mainly when they are working or travelling for a holiday, and often leads to complete soil and plant death because the residents cannot water them consistently. There is a clear growing need for an autonomous smart irrigation system that can fully act autonomously while there's an internet-enabled water monitor available live, and virtually eliminate the uneasiness of constant weekend or vacation-related plant care. This project is about auto watering systems that relieves the assiduous task of watering plants and allows live feedback on the condition of the plants to ease the maintenance of their plants and have more fun. This is especially significant given climate change issues, and we need to promote resource-efficient systems like this, and proactively show the value in protecting water resources. Furthermore, even if you are able to grow plants sustainably, this system is more sustainable because it can only be watering the plants when absolutely necessary, thus being optimal for water resources. It is out of these problems that we are directed to the project of considering issues pertinent to urban

## **1.3 Project Objective**

The main objective of this project is to design and implement an IOT-based smart plant irrigation and monitoring system that can monitor and control plants based on real-time environmental data. Automatic water system: Develop a system that automatically irrigates plants when the soil moisture sensor detects that moisture levels have fallen below a predetermined limit. This ensures that plants get water only when they need it, which prevented both under-watering and over-watering. Real-time monitoring and remote control: Apply a real-time monitoring system that allows users to track environmental conditions of their garden (eg, soil moisture, temperature, humidity) through the Blynk mobile app. Users will also be able to manually control the irrigation system, making them able to bring water to their plants on demand. Rain detection and water conservation: Include a rain sensor that will detect rain and stop the irrigation system to spare water when it rains. This adds a measure of water conservation by stopping over-watering. Animal detection and

safety: Include PIR (passive infrared sensor) to detect animals or other intruders to prevent damage to the garden. When a motion is detected, the system will notify the user using the mobile application, should the user want to take measures to control it. User-friendly interface: Create an effective and easy-to-use interface through the Blynk app that allows the user to view and interface with the system, view real time data, obtain information and control the irrigation process without any in-depth technical knowledge.

#### **1.4 Research Questions:**

##### **Primary Research Question:**

RQ1 : How can an IoT-based system like *Krishibot* be designed and implemented to automate irrigation and monitor environmental parameters for balcony and rooftop gardening effectively?

##### **Secondary Research Questions:**

RQ2:What are the key environmental factors that need to be monitored to ensure optimal plant health in an urban gardening setup?

RQ3:How can sensor data be utilized to automate irrigation while minimizing water wastage?

RQ4:What role does cloud integration play in providing real-time monitoring and control of the irrigation system?

## 1.5 Expected Outcome

The project "Krishibot" will present a new and innovative IoT-based device that automates plant care in urban gardens. The Krishibot expected to take contributions from urban gardeners in manual and automatic modes to ensure that plants are regularly watered and monitored even when users are not using the device and are away from the garden. Taking advantage of sensors to measure soil pH, humidity and temperature Krishibot offers live data via an OLED screen and a smart phone Android application. It will be designed so users can store their plant care data onto a Firebase real-time database and will provide notifications, which help provide smart choices for their garden. As well as watering plants, the device will allow users to spray water and pesticides, improving the health of their plants and reducing manual labour. Ultimately Krishibot seeks to make urban gardening simple and efficient. It considers issues involving lack of time, scant knowledge of plant care, and produces a reliable and easy to use gardening device. The combination of automation and remote control will minimize human intervention, support sustainable methods and enhance a greener urban ecosystem. Moreover, since Krishibot can be controllable through an Android application, it works synergistically with busy lives especially for those who travel frequently. Real-time notifications and reminders will allow users to stay informed on when to contribute to the care of their plants easily. Krishibot provides users with the necessary resources to keep their plants flourishing through humidity, temperature, KRISHIBOT will provide insights into their plants' health. The Krishibot, is designed to aid data logging and analysis using the Firebase and Think Speak servers, allowing the user to monitor trends over an extended period of time, and adjust gardening practices as required. Through the inclusion of features like pesticide spraying, the device offers a well-rounded solution to plant care, while simplifying the process and improving efficiency. More than just a practical solution, Krishibot helps fuel a desire to grow plants on rooftops and balconies, enhancing urban greening, improving air quality, and beautifying cities. This could lend itself to catalyzing interest surrounding smart gardening devices, sparking a wider adoption of these devices to reach sustainability goals.

## **1.6 Report Layout**

The report on "Krishibot" has seven chapters. The first chapter is the Introduction, which describes the motivation for the project, project's aims, expected deliverables, and report structure. The second chapter is the Background, which discusses the previous work and analyses, compared with our work, as well as challenges met while developing the project. The third chapter is Requirements Specification. This chapter can be read to understand the business process model, use-cases, system hardware and software requirements. The fourth chapter is Design Specification, which discusses front-end, back-end design processes, interaction design, and what was required to implement the project. The fifth chapter, Implementation and Testing, maps out how the system was built, tested, and what the outcomes were. The sixth chapter is the Impact on Society and Environment, which discusses how society can benefit, as well as environmental benefits. The seventh chapter, Conclusion and Future Scope, summarises what the project has achieved and provides some hints as to areas of future development, then a References section is provided which contains all references used. This organisation ensures the entire process of the project has been clearly documented and articulated.

## **Chapter 2**

### **Literature Review**

#### **2.1 Preliminaries**

The erstwhile section in the "Krishibot" project introduces basic concepts and backgrounds to develop the IOT-based smart garden monitoring system designed for urban roof and balcony garden. Here is a deep breakdown:

#### **The main concept of Krishibot:**

Krishibot is an IOT based device that remote monitors and controls what the garden needs, ( watering the plants ) by utilizing a network of sensors and a mobile app. The device uses a wireless sensor network to collect information from sensor nodes, placed in and around the garden, and sends this data to the Cloud. You can then get access and control of the device through a mobile app called "Krishi aid", that gives you real-time information and or feedback of environmental variables such as temperature, humidity, soil moisture, and water levels. Major components and functions:

#### **Sensors and data collections:**

Krishibot is outfitted with several sensors, including temperature sensor, humidity sensor and soil moisture sensor, which consistently keep an eye on the ecosystem. These sensors send a stream of data to a microcontroller (ESP8266 node MCU), which processes data and transmits the data to the cloud for storage and analysis. Automation and Remote Control: The device can control the water pump for irrigation automatically, turning it on or off based on sensor readings. Users can also control the device manually through a mobile app, which establishes flexibility in the automation. Cloud takes advantage of cloud technologies such as Krishibot Firebase and aims to store data and enable real-time monitoring is useful to support user. This cloud integration allows the users to investigate garden conditions over time and access historical data giving them useful information to help support their plant care decisions.

## 2.2 Related Works

In "Krishibot," the related work section assesses a number of studies and existing systems that used IoT technologies to develop automated irrigation and plant care. Below is a summary of how they were assessed: Early efforts at automating irrigation mostly employed microcontroller-based designs which achieved a degree of remote operation and control. Rajendranath and Berlin Hen's (2015) system linked soil-moisture sensors to real-time SMS alerts allowing a user to take manual action once a threshold was reached. While offering simple real-time operation, it lacks cloud storage and a graphical interface for data manipulation and analysis, considerably inhibiting better long-term monitoring and usability [1]. Based on that work, Rawal (2017) created a proof of concept IoT enabled smart irrigation technology that initiated watering using real-time soil-moisture only - offering good water efficiency and improved environmental stewardship. The design did not offer historical logging; nor did it have a mobile dashboard, thus minimizing engagement and flexibility for the user [2]. Attention, then, turned to precision water-management strategies for agriculture at scale. Zhang et al. (2018) used water-balance calculations to scientifically and systematically distribute irrigation, achieving notable water savings from large farming plots, but they did require extensive hardware and technical management, which would not be possible in a small urban garden [3]. Pooja et al. (2019) added environmental sensing (temperature, humidity, soil moisture) and paired it with a GSM module for sample alerts; however, there was no mobile-app interface or cloud-based archive to allow the user to visualize trends or interactively control the system [4]. The collective studies represented an expanded range of sensed parameters and analysis. Disha et al. (2020) included nutrient probes (N, P, K) for fertilizer inference and taking IoT gardening towards precision agronomy, however irrigation had to be turned on manually, and once again the price of the sensors represented an additional, ongoing cost [5]. Jain et al. (2021) embraced large sensor networks and cloud computing to track pests, growth patterns and irrigation requirements over larger spaces. However, its complexity, power consumption or expense were prohibitive for the common household gardener [6]. Singh et al. (2022) connected real-time weather forecasts with moisture data to propose a watering schedule that would skip watering immediately before rain events; however, this configuration relied upon expensive weather-API subscriptions and good connectivity that is rarely (or never) found on small balconies [7]. Chen et al. (2021) used a narrower focus on rooftops and

provided a web-based dashboard for live monitoring of metrics. However, Chen et al. (2021) had not provided mobile control of the system and they had not accounted for notifications from their tuple and it is clear that ease-of-use is not considered. Krishibot is situated to bridge the points made in both sets of research, and it offers necessary advantages over previous work in a form-factor that is suitable for urban rooftops or balconies. Krishibot combines low-cost combined sensors, two modes (operating, monitoring), and light-weight ESP8266-based architecture with Firebase Cloud logging, real-time OLED display of readings or measurements, and full mobile-app control via Blynk. Unlike an automated farm system, it is portable and energy-efficient. Unlike the SMS systems, it records data for a meaningful analysis of trends and sends rich push notifications. By connecting automation, accessibility, and sustainability together through both criticality and reconsidering ease-of-use, Krishibot has closed the usability gap of previous prototypes and is providing an end-to-end smart-gardening toolkit for our tenants and city's [9]. One supplemental study is an automated drip irrigation system by Verma et al. (2021) that used Arduino boards and GSM modules to control irrigation conditions depending on the moisture level. This study mentioned the system was inexpensive and straightforward, but the user interfaces were basic and no long-term data analytics functions were provided [9]. Meanwhile, Sharma and Patel (2023) developed a LoRa-based smart irrigation system, which provided long-range data transmission appropriate for rural communities. While the system was appropriate for agricultural applications with large land areas, experimental results cannot be used for small areas such as balconies or terraces, and the hardware would require additional expertise [10]. Iqbal et al. (2020) explored machine learning models in a smart irrigation context, enabling predictive decision-making in water distribution. Use of machine learning (ML) based systems requires significant datasets, persistent internet connection, and processing capabilities, making them impractical from a compact Low-Cost IoT device standpoint [11]. Another contribution is the report by Rahman et al. (2021), which used solar power sourced through IoT in irrigation systems aimed specifically at fostering sustainability in off-grid areas. Solar power fits with a sustainability construct, but adding solar panels and power management components increases the total cost, and overall physical footprint making it unfeasible for use in confined areas of urban spaces (where much of the current population growth is happening [12]). Hasan et al. (2022) narrated their experience developing a Bluetooth operated smart gardening assistant with local control of watering schedules, but it was limited to the area around the user with limited range, and did not enable remote monitoring or

connectivity to other cloud tools [13]. Kamble et al. (2021) developed irrigation systems based using Google Firebase making use of NodeMCU and Blynk, which was almost exactly the same design philosophy of Krishibot. Basic data storage capabilities allowed for receipt of notifications as often as every second for irrigation times but lacked on-device feedback similar to an OLED display to incorporate information and provided no modularity of input sensors where it could be expanded [14]. Mohana Priya et al. (2020) honed a system that both automated irrigation while using a voice-activated command from a Google Assistant. This was convenient, especially for visually impaired and elderly users, but depended on a good internet connection, and had a small voice processing time delay to operate the command [15]. Another project by Pathan and Sontakke (2022) used OpenCV image processing to detect diseases in plant leaves and alter irrigation based on those readings. This project was another instance of innovation, yet it required image capture hardware and more processing power than may be found in low-cost urban IoT devices [16]. Additionally, Hossain et al. (2022) created a mobile-app-controlled, ESP32-based Wi-Fi irrigation system, capable of switching between automatic and manual modes (Hossain et al., 2022). The benefit of this system is flexibility, but the focus on soil moisture failed to extend to other forms of environmental sensing, which reduces the adaptability of the system to a broader context (Hossain et al., 2022)[17]. Compared to all these solutions, Krishibot was uniquely able to combine all features including real-time sensing, mobile app functionality, cloud storage, local display, local and mobile dual operational modes, and most importantly, it was packaged in a relatively low-power, and low-cost design adapted specifically for rooftop and balcony gardening. Moubarik et al. (2022), who proposed an intelligent irrigation system that used a fuzzy logic approach of irrigation that estimated water flow based on multiple sensor inputs - soil moisture, temperature, and solar radiation were the primary sensors used, and using only those sensors improved the efficiency of the usage of water while require more processor power and more complexity to survey and calibrate rules for irrigation than necessary for light-weight, low-cost devices [18]. Basak et al. (2023) developed an IoT-based greenhouse automation system that monitored climate variables and controlled ventilation, lighting, and irrigation through a Raspberry Pi and Node-RED dashboard. Although the system is very functional, the cost of potentially additional hardware made it impractical for small-scale rooftop or home gardening [19].

To help with accessibility, Al-Turjman and Malekloo (2021) developed a fog-computing-based smart irrigation model that aimed to reduce latency by providing data processing for sensors nearby the field, as opposed to relying solely on cloud processing. The solution was useful for agriculture in rural areas with poor connectivity, however, it added complexity to the system design as fog nodes must be deployed and maintained [20]. Manoharan and Venkatesan (2022) additionally developed a low-power, Bluetooth Low Energy (BLE)-based irrigation system suitable for small gardens. Their solution was energy efficient, but the limited range of the BLE connection and lack of cloud-connected environmental monitoring meant that real-time remote monitoring and data logging were not possible [21]. Sikder et al. (2022) developed an ML-powered plant health prediction system that recognized plant stress through sensor data and a trained model and suggested irrigation and fertilization. Although the potential offered by this solution is very innovative, use of a solution that must be trained continuously or provided with new datasets and updates at regular intervals is not practical when operating in a constrained environment such as a rooftop or balcony [22]. Additionally, Sultana et al. (2023) designed a hybrid solar-wind-powered irrigation controller for off-grid farms. Though eco-friendly and sustainable, the system's size and infrastructure requirements made it better suited for larger agricultural plots rather than confined urban gardening spaces [23]. An important user-focused innovation was proposed by Kumar et al. (2021), who developed a multi-platform mobile app to control home irrigation systems using MQTT protocols. The app provided flexible controls, sensor status, and manual override features. However, the system lacked integrated cloud storage, limiting historical trend analysis [24].

### **Conclusion of Related Work Evaluation:**

The related works reviewed reveal the gradual advancement of IoT in smart irrigation and agriculture, from basic SMS-based monitoring to complex cloud-driven, weather-integrated systems. Krishibot stands out by tailoring this technology to urban gardening, combining IoT-based monitoring, user-friendly app control, dual operational modes, and portable design. This makes it an ideal solution for urban dwellers seeking a compact and convenient way to maintain balcony or rooftop gardens without the high cost or complexity associated with larger agricultural IoT solutions.[8]

## 2.3 Comparative Analysis

The analysis comparison shows that current systems are sophisticated and costly, lacking portability even when optimized for large-scale agricultural contexts. Such is the case with small urban gardens, which makes them impossible to use on a smaller scale. Krishibot is an innovative answer to these concerns. It is perfectly adaptable to the needs of the modern urban gardener, striking balance between simplicity, cost, effectiveness, and ease-of-use. Its hands-free automated plant tending eliminates the need for any complex system management. Additional features, such as its mobile app and dual operational modes, further make Krishibot mobile and lightweight—something current systems sorely need.

Feature/Project	Automated Irrigation (2021)	Smart Irrigation (2019)	Krishibot (2024)
Dual Modes	No	No	Yes
Mobile App Control	No	Limited	Comprehensive with real-time updates
Pesticide Management	No	No	Yes
Urban Gardening Focus	No	No	Yes

[5]

## 2.4 Scope of the Problem

The "Krishibot" project scope entails issues of urban gardeners being unable to maintain their rooftop or balcony gardens because of time management, resource management, and technical limitations. Consider the following ideas:

### Challenges in Urban Gardening:

#### Time Constraints:

Urban dwellers tend to miss daily check-ups or maintenance sessions with their plants because of uninterpreted schedules. During business trips, vacations, or other outings, plants can get neglected

**Inconsistent Watering:**

Manual watering can lead both over and under watering. Either one is harmful for plants. Proper irrigation is critical, but manual watering also poses other challenges, talking to plant health. Without the gardener, it's most difficult to ensure the right amount of water is supplied.

**Resource Wastage:**

Due to poor irrigation methods, water wastage poses a major threat. If there is no accurate monitoring, users are likely to go overboard with water use, which in this case is unsustainable.

**Environmental Variability:**

In rooftop and balcony gardens, soil moisture as well as temperature and humidity is prone to drastic variation, making it manual maintenance nearly impossible for optimal conditions.

**Lack of Available Technology:**

Most systems that are on offer are tailored to vast agricultural fields and for their size, cost, and intricacy unsuitable for small urban spaces like these.

**Monitoring and analyzing data:**

Urban gardeners are not able to track health trends (like monitoring the soil moisture level or temperature changes) over time due to an inefficiency in data collection and storage infrastructure.

**Timing the operation of the water pump:**

The exact duration of water pump operation is very critical as it determines whether overwatering or under watering occurs. It demands specific calculations and adjustments relative to the plants.

**Calibration and accuracy of sensors:**

Locally available moisture, temperature, and humidity sensors are not always perfectly accurate or trustworthy, making selection difficult. Given the changing conditions, it is critical that the sensors endure and are stable to environmental wear deformation.

**Connectivity problems:**

An IoT based system requires an uninterrupted internet connection. Instability in Wi-Fi availability can hinder real-time system monitoring or automated cloud updates.

**Availability of components:**

The project is likely to be delayed and costs increased if appropriate, cost-effective and accessible parts like microcontrollers, sensors, and motors are not readily available.

### **Integration and compatibility:**

Aligning various elements such as sensors with microcontrollers and actuators to form a functional system is challenging and requires expertise. The failure of any one element may yield an inconsistent system.

### **Challenges during the pandemic:**

A global crisis technological pandemic

### **Broader Implications of the Problem:**

#### **Urban Sustainability:**

Without effective solutions, urban gardening becomes less sustainable, discouraging individuals from creating green spaces in cities. This contributes to environmental degradation due to a lack of vegetation in urban areas.

#### **Environmental Impact:**

Improper irrigation and water wastage harm local ecosystems, particularly in areas where water resources

### **Limited Adoption of IoT in Small-Scale Gardening:**

IoT-based systems are not affordable and convenient, urban gardeners who greatly contribute to city landscaping are unable to utilize them. The issue's scope underlines the need for an IoT-based system like Krishibot that is small in size and cost-efficient. With informational automation and simple operation, Krishibot sets out to enhance sustainability towards plant care, therefore making it more manageable and encouraging people to practice urban gardening.

The water pump ON to get a specific amount of water. This problem we can solve by a simple trick that we are setting up a delay that once the motor is turned on it will continue for some time.

To get the right time of delay we are using a simple linear equation that is

$$T_{ON} = V_{ml} \cdot G[ms/ml] + K[ms]$$

T= Turn on the pump V=Amount of Water G = Flow fact

K= Pipe constant

## 2.5 Challenges

During Krishibot's development and implementation, we encountered the following challenges:

**Sensor Precision and Calibration:** Sensors (such as DHT11 and soil moisture sensors) were difficult to get accurate readings. For example, a calibration process was needed to bring forth accurate readings in different environmental conditions.

**Availability of components:** The availability of some components, like ESP8266 Node MCU and sensors, locally delayed our development.

**Integration Issues:** When integrating hardware components with software components (like Firebase and the Blynk app), we encountered issues that necessitated careful testing and debugging to ensure that the communication and functioning of the them was smooth.

**Internet Dependency:** The system utilizes the Internet for real-time monitoring and control, therefore is mildly dependent on an Internet connection. We were unable to achieve high-quality system performance due to intermittent internet access during testing

### **Power Management:**

It was tough to ensure a constant power supply for components such as a motor, sensors, and the microcontroller especially when the voltage was unstable.

**Hardware Durability:** In order to ensure long-term dependability with outdoor installation, it was important to protect sensors and electronic components from environmental factors like moisture, and dust.

### **Cloud Integration Complexity:**

Setting up the hardware and the cloud (for example, Firebase) took a lot of testing in order to synchronize data while avoid any interruption in loss of data or delayed update on real-time.

**Mobile App:** Ins iterations of testing and improvement were necessary to create a visually appealing intuitive mobile app with real-time feedback, and controls to improve user experience.

### **Testing and Debugging:**

A lot of time and effort was required to locate and correct bugs in the system during testing, such as erroneous sensor readings or latency in transitioning the motor.

### **Time and Resource Constraints:**

Managing technical difficulties while completing project development within the allocated budget and time frame presented difficulties.

In order to overcome these obstacles and make sure the system fulfilled the essential functionality and reliability criteria, meticulous planning, thorough testing, and teamwork were necessary.

# CHAPTER 3

## RESEARCH METHODOLOGY

### 3.1 Business Process Mode

The business process model for the "Krishibot" thesis represents the interaction between the system's components to achieve an automated and efficient plant care solution. The system begins with sensors collecting data from the environment, such as soil moisture, temperature, and humidity levels.

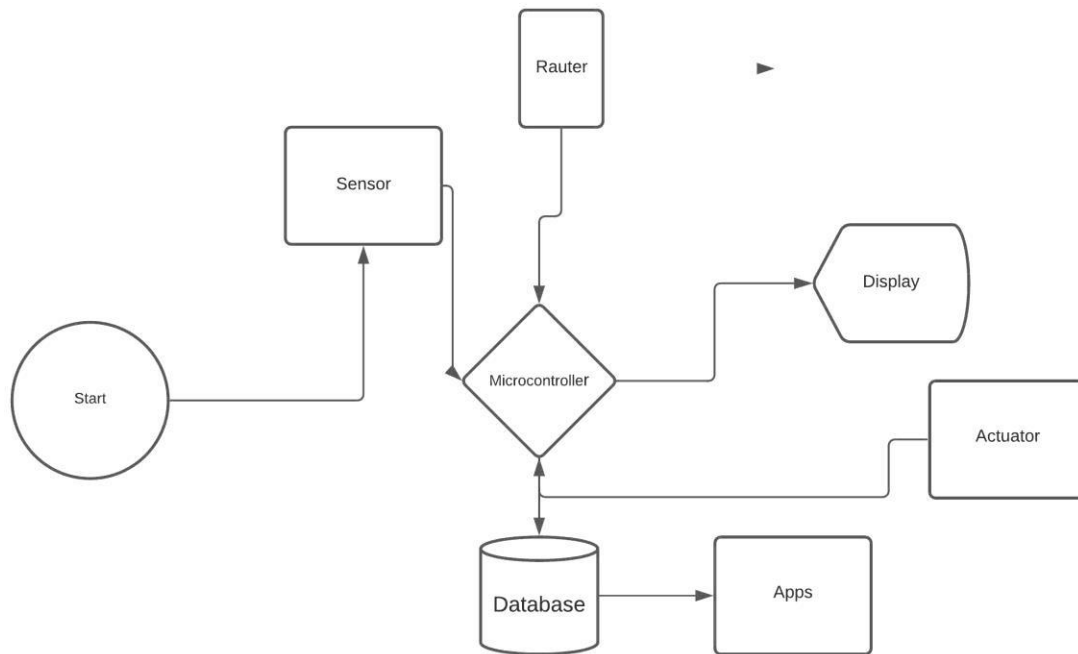


Figure 3.1: Business Process Model

The ESP8266 NodeMCU is the main microcontroller that processes the data from these sensors. Based on the data received from the sensors, the microcontroller can then decide what actions should take place, for example, turning on the irrigation water pump. The processed data is also sent to a cloud database like Firebase or Think Speak to give users live monitoring and storage of environmental metrics. The users will interact with the system via an Android application that will show them plant health metrics and gets the data from the cloud database. The user can control their system, including remote access to turn the irrigation motor on and off. When the system is in automatic mode, it will look after the plants on its own and ensure that they are never neglected without human attention. Most importantly, the system will be able to provide your plants with enough water no matter where you may be.

**3.2 Use-Case Modelling & Descriptions**

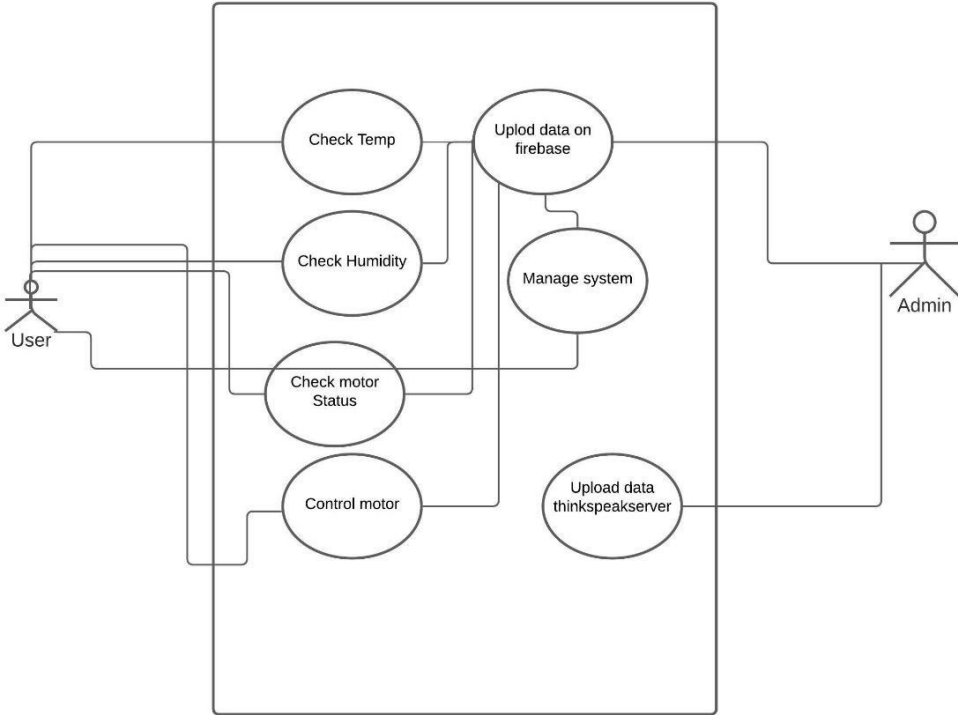


Figure 3.2: Use Case Diagram

The "Krishibot" system's use-case modeling describes how users will interact with the smart plant care system and its automation and monitoring capabilities. The main actor in the system is the user who uses an android application to monitor their plants. The application allows users to monitor metrics, such as soil moisture, temperature, and humidity. They use an ESP8266 NodeMCU microcontroller to power the devices that gather the sensor data and store it in a cloud-based database and allows access to the user instantly. Users can also manually operate the device through the application. For example, users can turn on or off the water pump, or turn on automatic modes in the system that will irrigate the plants without human input. These automatic features reduce user responsibility as they can assure they are providing timely care for their plants without needing to be there to monitor them. This provides an added convenience and reliability for the user. The Krishibot system keeps you updated through notifications and alerts when important thresholds are reached, such as too low soil moisture levels, so you can react and take action on the spot. Also packed with capabilities such as pesticide spraying (which can be configured through the app). Each feature will attempt to streamline your gardening process, reduce manual labor, and support you in providing the best ongoing care possible of your plants. By leveraging staying IoT technology, cloud-based databases, and an easy to use mobile app, Krishibot has designed the system to offer the best smart gardening solution for urban settings.

### **3.3 Requirement Collection & Analysis**

The requirement and analysis phase of the "Krishibot" project aims to identify and specify the hardware and software components needed to accomplish an IoT-based plant tracking and irrigation system. The hardware requirements are driven by the primary component, the ESP8266 Node MCU microcontroller, which acts as the system's microcontroller unit (MCU), and also requires additional sensors, DHT11 (a temperature and humidity sensor) and a soil moisture sensor to monitor the level of moisture in the soil. Other components include separate pieces such as a breadboard and jumper wires, and a .96-inch OLED screen, 5V relay module to ensure connectivity and functionality. It also consists of a DC motor for pumping water and a Wi-Fi router so the device can communicate with cloud servers. In terms of software specifically the Arduino IDE, for programming the MCU and managing the interaction of the sensors.. Firebase and Think Speak cloud databases are used for storage and real-time data retrieval, allowing the user to track

plant metrics anytime. The Android application, built in Android Studio, provides a simple interface for easy remote control and monitoring of the cloud sensors. The app allows users to see live data on soil pH, temperature, and humidity, and allows for manual or automated irrigation management.

During analysis, a few things were considered regarding the reliability, compatibility, accuracy, and cost-effectiveness of the components. For example, the sensors were chosen not only because they are available in the local market but also because they're capable of providing beneficial data. The practical challenges, like getting accurate readings from sensors and getting cloud updates in real time, were dealt with in this analysis phase. This comprehensive requirement collection and analysis resulted in a cost effective, efficient, functional, and easy to use solution, accessible to a wider audience.

### 3.3.1 ESP 8266 Node MCU v3

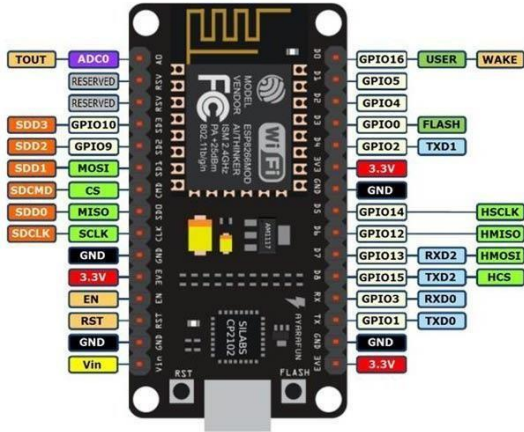


Figure 3.3.1: ESP 8266 node mcu v3

ESP8266 Node MCU v3 functions solidly as a capable microcontroller and is perfect for applications that use wireless technology such as Krishibot. The ESP8266 Node MCU v3 includes Wi-Fi capabilities to communicate remotely, it is well accepted in the IoT space because it is small, easy, and cheap. You could carry it in your pocket ! The ESP8266 has an integrated Wi-Fi module that allows it to communicate over the net wirelessly with other devices. This microcontroller supports both Lua and Arduino IDE programming languages making it beginner-friendly. In addition to being beginner-friendly, Node MCU v3 has many GPIO pins to attach external sensors and actuators and has a USB for easy programming and power supply. The Node MCU v3 can work with low power consumption which is critical to IoT devices such as Krishibot. The ESP8266 also supports HTTP and MQTT to make it easy to interface with cloud services like Firebase and thinkSpeak. The ESP8266 is configurable with a clock speed up to 80MHz and has up to 4MB of flash memory which makes the ESP8266 Node MCU v3 a fully capable microcontroller

#### **.Role of ESP8266 in Krishibot:**

The ESP8266 Node MCU v3 in the Krishibot system serves as the brain, controlling all functions. The brain collects data from the sensors that report soil moisture, temperature, and humidity. All of this data is used to make choices about the necessary actions; greater actions from the system such as turning the water pump on or off. The brain connects to a cloud database to store recorded sensor readings and send updates to the Android app, which allows for real-time visibility for the user. The ESP8266 enables remote control of the system through internet-controlled functions; enabling the user to operate Krishibot from anywhere in the world. The low-power processing capability and ease of connection, fasten its role in the Krishibot system, contributing to the automation, reliability, and user-friendliness.

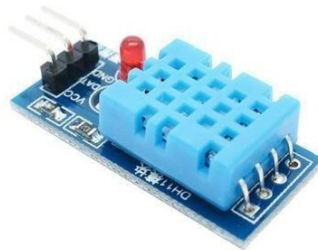
#### **Why ESP8266 Node MCU v3 was Chosen:**

The ESP8266 Node MCU v3 was used in the "Krishibot" project for its unique combination of attributes, versatility, and affordability, making it well-suited for IoT-based applications. The ESP8266 microcontroller has built-in Wi-Fi, which allows for wireless connections to cloud databases and the Android application. The microcontroller can support multiple IoT protocols such as HTTP and MQTT, which enables a consistent data exchange. The ESP8266 allows

constrained, defined data exchange from a sensor to the cloud to the user interface.

Another important factor for the ESP8266 is that it is also compatible with numerous sensors and actuators. Therefore, the microcontroller has several GPIO sticks (as we call them) that also simplify the integration of the components such as a soil moisture sensor, DHT11 moisture and temperature, and the relay modules that are needed to control the water pump. In addition, the Arduino IDE helps provide non-traditional programming and LUA script as support for programming the microcontroller, which helps for developers to create effective and reliable system operations even when not an advanced user. Furthermore, the ESP8266's small size, low energy usage and strength, was also significant considerations for the Krishibot project. In these respects the manufacturer can produce a light and portable unit that is always energy educated making it a better financial value for users.

### 3.3.2 DHT 11 Humidity Sensor



The **DHT11 Humidity Sensor** is an essential component of the **Krishibot** project, used to monitor environmental conditions by measuring humidity and temperature. Its data helps ensure optimal conditions for plant health.[18]

DHT11 Moisture sensor is an inexpensive, basic digital sensor measuring environmental temperature and moisture. The DHT11 switches as a digital signal through data papine using capacitive moisture sensor and thermists for proper reading. Easily affordable and used safely and rapidly, the DHT11 is implemented across a range of IoT projects including the Krishibot system. In spite of most capabilities, for all required purposes, the DHT11 is reliable. The DHT11 has a temperature range of 20-90% RF and temperature range of 0-50 ° C; therefore, it can provide suitable conditions to monitor plant growth. Plugged to microcontrollers such as ESP8266 gives

improved functionality, with the low power require for DHT11 means it is handy for real -time environmental monitoring design without worrying about the batteries dying.

**DHT 11 Moisture Sensor Role in Krishibot:**In Krishibot, the DHT11 moisture sensor is critical for sensing the environmental conditions crucial to health of plants. It collects information about the temperature and humidity of the environment around the plants, information is collected in Real-time to be sent to the ESP8266 microcontroller where it is recorded on the OLED screen, stored in shooting database like FireBase and sent to the Android user monitoring application. The information collected from the DHT11 allows Krishibot to make intelligent plant care choices. For example, if the environmental temperature is above a particular threshold, the Krishibot may send a notification and change any watering schedule to ensure optimal plant growth. As an additional aspect of the data collection and monitoring, the DHT11 provides daily moisture levels to give the user an overall picture of the moisture in the environment., so that timely action to protect the plants from unfavorable weather conditions.

### 3.3.3 Breadboard

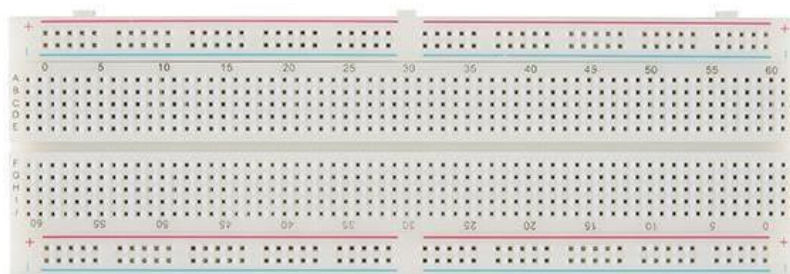


Figure 3.3.3: Breadboard

Breadboard is an essential prototyping board used to design and test electronic circuits prototyping board used to design and test electronic circuits in the Krishnabot system without a capelin. Breadboards provide an adaptable platform for collecting and connecting components such as sensors, microcontrollers, and actuators, Breadboard is a plastic, re-purposed board made specifically for prototyping electronic circuits without having to use a soldering iron. Breadboards have holes in a grid pattern with paired terminals that are designed to allow easy insertion of electronic components (resistors, sensors, microcontrollers, etc), which can be simply connected with jumper lines. The aspect of breadboard allowing flexibility makes it ideal for quickly creating

and modifying circuits so developers can observe and troubleshoot their designs effectively before committing to a permanent configuration. [14] Role in Krishibot: Within the Krishibot system, the breadboard is a primary board that functions as the basic platform to help assemble and test a circuit. The board allows easy connection of components such as the ESP8266 microcontroller, DHT11 moisture sensor, soil moisture sensor and relay module. The breadboard provides the convenient function of a temporary circuit allowing developers quick fixes, alterations and repairs during their development process. The breadboard accommodates flexibility during the development stage before finalizing any successful hardware design for electrical connections.

### 3.3.4 Soil Moisture Sensor



Figure 3.3.4: Soil Moisture Sensor

A soil moisture sensor is a device for measuring the amount of water content in the soil. These sensors typically use capacitance or resistance to determine the moisture level and convert this measurement into a voltage signal. Soil moisture sensors are used in agriculture and gardening applications to measure soil moisture, and can be used to implement an organized irrigation schedules. They are very effective, easy to use, and made from corrosion resistant materials for maximum use and dependability. Contribution to Krishibot: In Krishibot, the soil moisture sensor is a fundamental element that is responsible for determining when the plants require watering. It is always measuring the soil water content, and forwarding information in real time to the ESP8266 microcontroller. The automated sensor saves water, but also encourages healthy plant growth. The soil moisture data is also forwarded to an OLED display, and to the mobile application for the user to view their soil conditions remotely, and to allow for manual control if required.

### 3.3.5 Jumper Wire

Jumper wires are used to connect additives with out soldering. Easy to use just plug and play. Jumper wires are vital gear used within the Krishibot venture to attach various digital components during prototyping and assembly.

These bendy, insulated wires permit short and solder-loose connections between the microcontroller, sensors, and different additives on the breadboard.[19]



Figure 3.3.5: Jumper wire

#### **Role in Krishibot:**

In the Krishibot machine, jumper wires serve a purpose allowing transient electric connections among the components of the circuit, including the ESP8266 microcontroller, sensors (such as the soil moisture sensor and DHT11), relay module, and other components on the breadboard, without the need for soldering. Jumper wires make sure there is seamless communication between these parts, sending signals.

### 3.3.6 .96 OLED Display

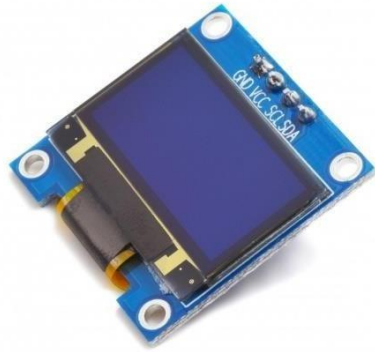


Figure 3.3.6 0.96 OLED Display

The OLED (Organic Light-Emitting Diode) Display is an important part of the Krishibot challenge, providing a small and energy-efficient way to provide real-time data from the sensors. This adds to the user experience by providing immediate, on-site access to environmental conditions and system status. [20]

#### **Function in Krishibot:**

The 0.96" OLED display within the Krishibot system plays an important role in providing real-time, on-tool feedback to the user. It is especially important for displaying key information including soil moisture levels, temperature, humidity, and motor status, enabling users to check the condition of their plants without relying on the mobile app exclusively. The OLED display's high contrast and energy efficiency make it well-suited to the Krishibot, ensuring it can easily be seen in low-light situations. The OLED display will also provide a local interface for the user, even when they do not have access to the app or the internet. The display provides direct indication of key metrics, increasing usability and ensuring that important data is always local and available, making Krishibot a more user-friendly and efficient smart gardening solution

### 3.3.7 5V Relay Module

5V Relay module is used for the automatic control circuit. It is an automatic switch to control high current circuit with a low current signal.

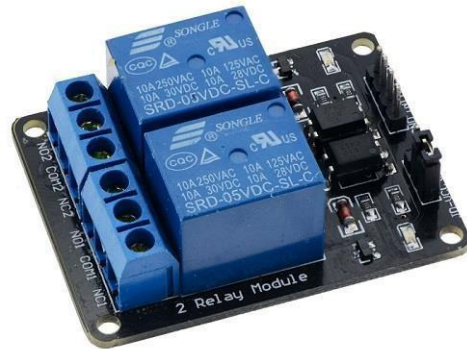


Figure 3.3.7: 2 Channel 5V relay Module

The 5V Relay Module is an essential element inside the Krishibot device, which acts as an automated switch to control high-powered devices like the water pump using low-power signals from the microcontroller (ESP8266 Node MCU).

#### **In the Krishibot's operation:**

The 5V relay module inside the Krishibot device is responsible for controlling high-power devices, such as the water pump, using low-power signals from the ESP8266 microcontroller. As an automated switch, the relay module allows the microcontroller to safely control the operation of the pump without having to handle high currents. Once the device determines that watering is required based on the sensor readings, the 5V relay module then turns on the pump by closing the circuit, likewise it will turn off the pump based on the desired conditions.

Integrating the relay module provides electrical safety and reliability in the Krishibot device. It also facilitates simple automation of the irrigation process, reducing the need for manual intervention while also protecting the microcontroller from excessive current load.

## Technical Specifications

Operating Voltage: 5V DC

- Load Voltage: Up to 250V AC or 30V DC
- Maximum Current: 10A
- Number of Channels: Single-channel relay used in *Krishibot*

In summary, the 5V Relay Module plays a vital role in enabling the automation of *Krishibot*, providing efficient control of the water pump while ensuring the safety and reliability of the overall system.[22]

### 3.2.1 DC Motor



Figure 3.3.8: DC Motor

A **DC motor** is an electrical device that converts direct current (DC) into mechanical energy, typically used to drive various mechanical systems. It operates on the principle of electromagnetism, with its speed and torque being easily controllable. DC motors are widely used in applications requiring reliable and efficient mechanical movement, such as pumps, fans, and conveyor belts.

**Role in Krishibot:** In the **Krishibot** system, the DC motor powers the water pump, enabling the automated irrigation process. When the soil moisture sensor detects that the soil is dry, the ESP8266 microcontroller signals the relay module to activate the motor.

# CHAPTER 4

## DESIGN SPECIFICATION

### 4.1 Front-end Design

The front-end layout of the Krishibot system encompasses each hardware and software interfaces to make sure seamless interplay and usability. For the hardware the front-end, the system layout is initially prototyped the usage of Fritzing software to create an correct circuit format. This includes connecting additives inclusive of the ESP8266 NodeMCU, DHT11 sensor, soil moisture sensor, relay module, OLED show, and DC motor.

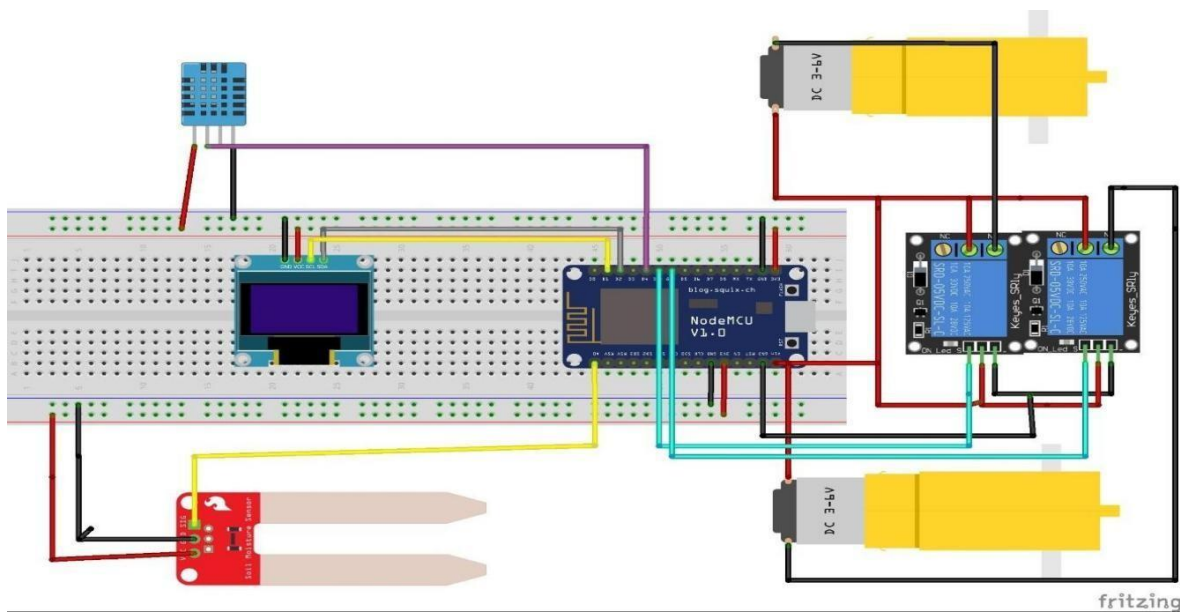


Figure 4.1.1: Front-end Design for Hardware sections

The software front-end is designed by using Android Studio as the UI for a very user-centred Android application, called "Krishi Help". The Android app provides an easy interface to allow for remote monitoring and control of the device. The app allows the user to see real time data, such as the soil moisture, temperature and humidity, and to control the motor and irrigation system. The front-end in this case is two layers, thereby allowing the user direct interaction with the device

whilst also interacting remotely on a mobile device. When constructing for the software part, we are using android studio for UI purposes.

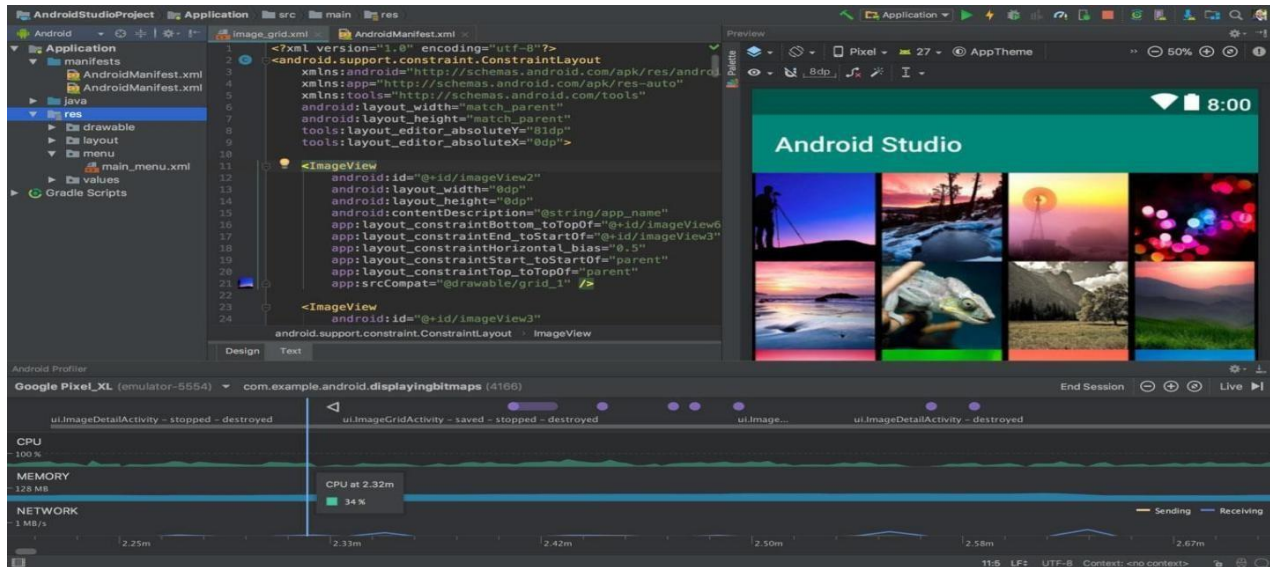


Figure 4.1.2: Front-end design for software

## 4.2 Back-end Design

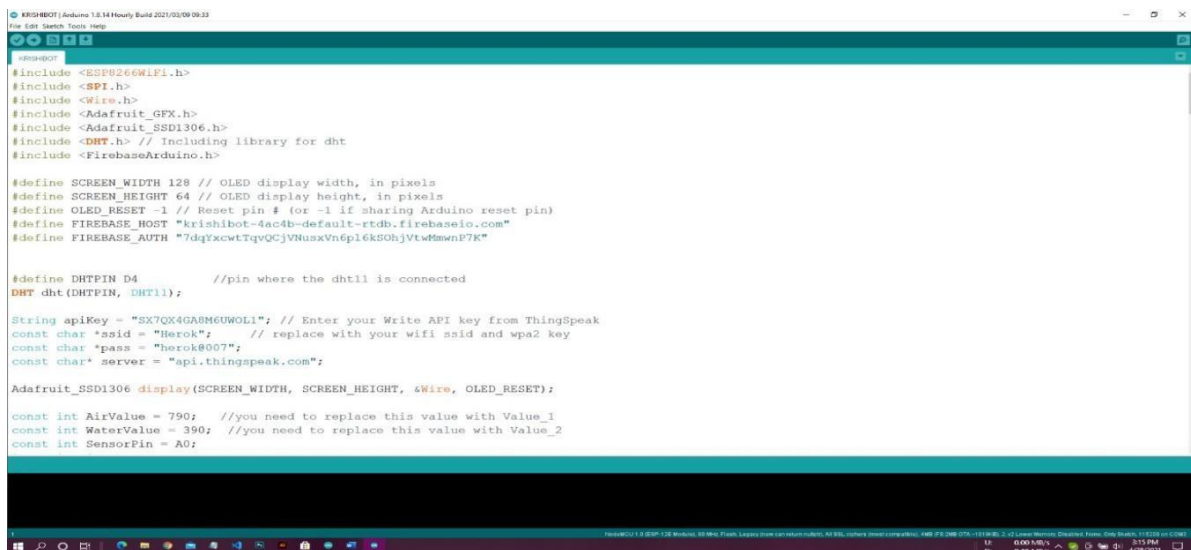


Figure 4.2: Back-end design process

The back-end design of the Krishibot gadget integrates hardware additives, cloud services, and communication protocols to make sure seamless operation and records management. At its center, the ESP8266 Node MCU microcontroller tactics records from sensors consisting of the soil moisture sensor, DHT11 for temperature and humidity, rain sensor, and PIR movement sensor. The processed statistics is transmitted to Firebase in real-time, allowing cloud-primarily based data garage and synchronization with the Blynk cellular app. This lets in customers to display their garden’s situations remotely and control the irrigation gadget manually or automatically. The microcontroller communicates with the 5V relay module to control the water pump based totally on sensor readings, making sure unique irrigation. Real-time comments is also supplied locally through an OLED display, enhancing usability. The again-quit architecture is designed to help dependable, low-latency verbal exchange among components and the user interface, ensuring a strong, efficient, and consumer-pleasant revel in.

### **.Interaction Design and User Experience (UX)**

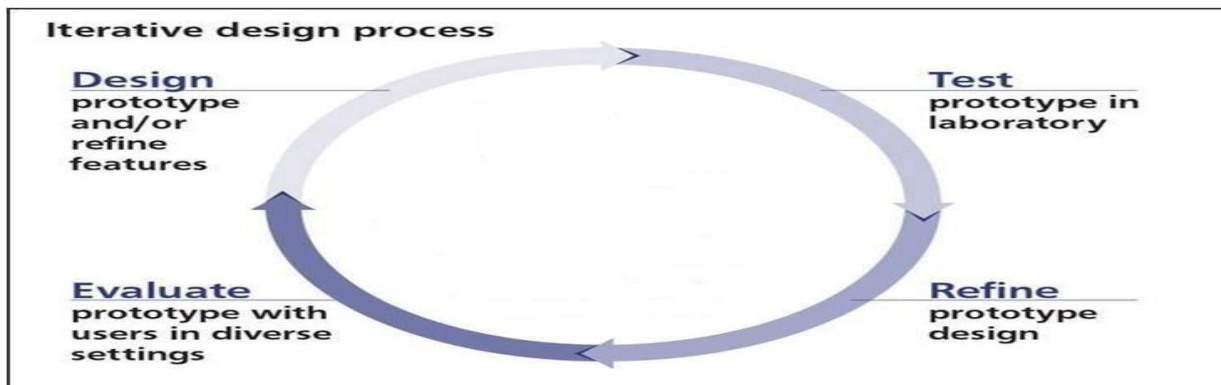


Figure 4.3.1: Interaction design

The interplay layout and consumer revel in (UX) of the Krishibot machine prioritize simplicity, efficiency, and accessibility to make sure a continuing and user-friendly revel in for urban gardeners. The system’s interplay model revolves across the Blynk cell app, which acts as the primary interface for tracking and controlling the garden remotely. The app features a easy dashboard displaying real-time sensor readings for soil moisture, temperature, and humidity, at the side of gadget reputation updates. Users can switch among computerized and guide irrigation modes with only a tap,

providing flexibility and manage. Notifications are designed to be instant and specific, alerting users about essential activities like low soil moisture, rain detection, or movement near the vegetation.

### User Experience (UX) Design Principles

The Krishibot device's User Experience (UX) design is guided with the aid of concepts that prioritize simplicity, efficiency, and accessibility to make sure a unbroken interaction for users of all technical backgrounds. The device functions an intuitive mobile app powered with the aid of Blynk, presenting real-time tracking and control of key environmental parameters together with soil moisture, temperature, and humidity. The app's easy and minimalistic layout makes use of a dashboard layout to provide critical statistics simply, enabling customers to navigate and have interaction with the machine effortlessly. Features like guide and automated irrigation manipulate are designed to be just a faucet away, whilst notifications for important events consisting of low soil moisture, rain detection, or movement intrusion make certain timely consumer engagement. By retaining consistency in iconography, button placement, and visible hierarchy, the interface reduces cognitive load, making it easier for users to cognizance on dealing with their flora efficaciously.

### Benefits of Effective UX Design

The *Krishibot* system's interaction design and UX emphasize simplicity, accessibility, and efficiency, ensuring a smooth user experience for urban gardeners. The mobile app, integrated with

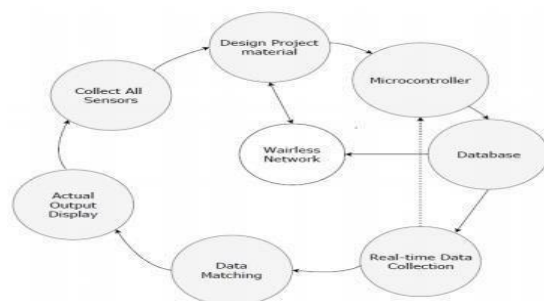


Figure 4.3.2: Interaction Design

real-time feedback and intuitive controls, empowers users to care for their plants effortlessly. And send data to the cloud by using a microcontroller. Based on the data actuator are take action.[8]

## 4.1 Implementation Requirement

For the Krishibot system to operate properly and robustly, it is important to integrate the hardware, software and environmental setups that constitute the system, prudently. The hardware components include a central microcontroller, sensors, actuators and ancillary components to enable the collection of data, the processing of the data, and to initiate actions. The ESP8266 Node MCU acts as the hub of the system by enabling data collection from the sensors, which include the soil moisture sensor as well as the DHT11 for temperature and humidity, rain sensor and PIR motion sensor. The actuators based on the processed facts from sensors with the 5V relay module and the DC water pump are controlled by the ESP8266 Node MCU to ensure precise irrigation activities. Other components such as the OLED display unit provide real-time feedback locally while power supply components (5V supplies for microcontroller and sensors, 12V for the pump) ensure stable power supply. The software program requirements revolve across the programming and integration of the hardware with cloud services. The Arduino IDE is used to jot down and upload code to the ESP8266, defining the logic for sensor data processing, decision-making, and communicate. Firebase Realtime Database is employed for cloud storage and synchronization, allowing actual-time information updates and historic monitoring. The Blynk mobile app presents the number one interface for customers, permitting remote tracking and control of the irrigation device. Tools like Fritzing are used for circuit layout and prototyping, ensuring the hardware setup is properly established and documented at some stage in development. Functional requirements encompass the gadget's capability to mechanically activate or deactivate the pump primarily based on soil moisture ranges, locate rain to pause irrigation, and reveal temperature and humidity. It also helps manual manipulate through the app and provides notifications for crucial events like low soil moisture or animal intrusion. The machine have to be responsive, updating sensor information in actual-time and ensuring synchronization among the microcontroller, cloud database, and cell app. These functionalities are important to deliver the preferred automation and consumer level in.

Environmental necessities encompass strong Wi-Fi connectivity to make certain uninterrupted verbal exchange between the microcontroller, Firebase, and the cellular app. Hardware components ought to be housed in protect. Lastly, the implementation is constrained by factors

such as cost, component availability, and potential environmental challenges. The components should be cost-effective without compromising performance, and readily available to avoid delays. The system's reliance on Wi-Fi connectivity necessitates addressing potential disruptions by designing fallback mechanisms for offline operation. By meeting these requirements, *Krishibot* ensures a reliable, efficient, and user-friendly solution for automating plant care in urban grade.

# CHAPTER 5

## IMPLEMENTATION AND TESTING

### 5.1 Implementation of Database

The execution order is to place a DBMS server on the ideal machine and update a file Database to run better on this hardware and organize programming and make database and store information. The basic information could either be that the unused information was obtained clearly or that the current data was imported from any DBMS. Manufacturers also create database security in this type and give the various clients that architects envision access to reasonable for their needs.

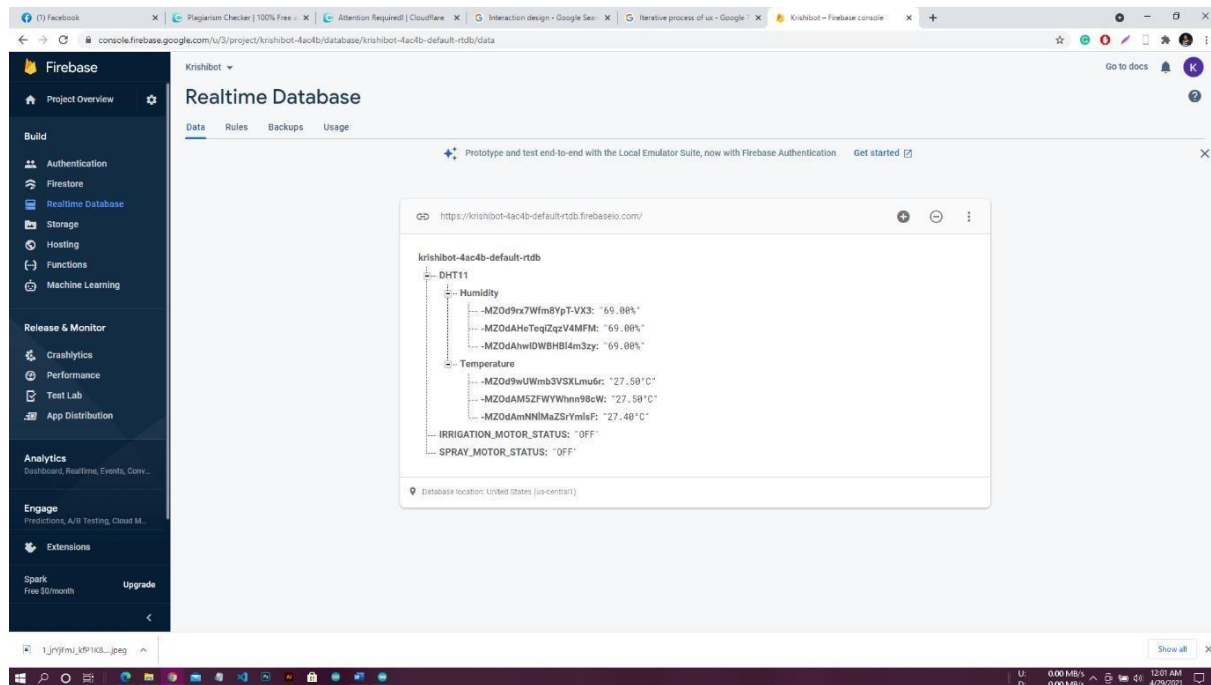


Figure 5.1: Implementation of Database

## 5.2 Testing Implantation

The following table outlines the testing process for the *Krishibot* system, focusing on its key functionalities, expected outcomes, and actual results:

Test Description	Input	Expected Outcome	Actual Outcome	Status
Soil moisture sensor in dry soil	Place soil moisture sensor in dry soil	Sensor detects low moisture, irrigation pump turns ON, data sent to Firebase, and notification sent to app	Pump turned ON, data displayed correctly, notification received	Pass
Soil moisture sensor in wet soil	Place soil moisture sensor in wet soil	Sensor detects adequate moisture, irrigation pump remains OFF, data updated in Firebase	Pump remained OFF, data displayed correctly	Pass
Manual pump control via app	Turn ON/OFF pump using the mobile app	Pump responds to app commands and updates status in real-time	Pump turned ON/OFF as expected, status updated	Pass
Rain detection	Simulate rain on rain sensor	Sensor detects rain, irrigation is paused, and notification sent to app	Pump turned ON/OFF as expected, status updated	Pass
PIR motion detection	Simulate motion near the sensor	Sensor detects motion, notification sent to app	Motion detected, notification sent to app	Pass
Real-time data display on OLED	Collect sensor data in real-time	OLED displays current temperature, humidity, and soil moisture levels	Data displayed correctly on OLED	Pass

Cloud synchronization	Update sensor readings to Firebase	Firestore database updates with real-time sensor data	Data synchronized and visible on mobile app	Pass
App notification for low moisture	Dry soil conditions detected	App receives notification for low moisture levels	Notification received as expected	Pass
System recovery after power loss	Restart system after power cut	System resumes normal operation and updates sensor data	System restarted successfully, data updated	Pass
Sensor accuracy test	Compare sensor readings with reference	Sensor readings closely match reference measurements	Sensor readings were accurate	Pass

Table 1: Test Result

### Explanation of Columns

1. **Test No.:** Sequential identifier for each test case.
2. **Test Description:** Description of the functionality being tested.
3. **Input:** Specific action or condition under which the test is conducted.
4. **Expected Outcome:** Predicted behavior of the system based on its design.
5. **Actual Outcome:** Results observed during the test.
6. **Status:** Indicates whether the test passed or failed.

This table provides a structured way to document and verify the functionality of the *Krishibot* system.

### 5.3 Final Output

The Krishibot system has been efficiently applied, attaining all its supposed functionalities. It automates irrigation by activating the water pump based on real-time soil moisture degrees, even as rain detection ensures water conservation. The device presents actual-time monitoring of soil moisture, temperature, and humidity through an OLED display and the Blynk cell app, permitting users to govern the pump remotely. Notifications and alerts for events like low moisture, rain detection, and movement intrusion are sent seamlessly, improving person recognition. Cloud integration with Firebase guarantees data storage and ancient evaluation. With its lightweight, transportable design and reliable performance, Krishibot simplifies city gardening, promotes sustainability, and offers a convenient answer for busy gardeners. The machine correctly automates irrigation by way of activating the water pump when soil moisture drops under a predefined threshold and stopping it as soon as ultimate levels are reached, ensuring right watering whilst holding water. Rain detection generation pauses irrigation at some stage in rainfall, including every other layer of water efficiency. Real-time monitoring of soil moisture, temperature, and humidity is facilitated thru an intuitive OLED show and the Blynk mobile app, allowing customers to song their lawn's reputation from anywhere within the world. Notifications for critical events, together with low moisture, rain detection, and animal intrusion thru the PIR motion sensor, provide users with timely updates, improving system reliability and consumer comfort. The device helps both guide and automatic operation modes, supplying flexibility to customers based totally on their preferences. Its light-weight and transportable design makes it best for city gardening setups like rooftops and balconies. Testing outcomes showed that each one additives, from the sensors to the cloud integration and cellular interface, executed reliably and accurately under various conditions.

By simplifying gardening obligations and decreasing manual intervention, Krishibot promotes sustainable practices and empowers city gardeners, even those with busy schedules. Its aggregate of automation, real-time remarks, and ease of use ensures it meets the needs of modern gardening at the same time as keeping resources and improving environmental sustainability.

# CHAPTER 6

## IMPACT ON SOCIETY

### 6.1 Impact on Society

The Krishibot system contributes to society in a very positive way. It is a significant tool for promoting sustainable urban gardening and reducing the levels of effort involved with plant care. Essentially, it allows people, namely urban residents who may have little free time, to be able to keep a rooftop or balcony garden with very little effort, further promoting green space in urban places. The system will also automate irrigation and inline with the device's capacity to better monitor plants health in real time, reduce labor and engage substantially more people in gardening. In addition to further engaging people in gardening, the ability to manage the care of plants remotely can help alleviate worries when leaving for vacation or when busy, intuitively increasing access to gardening as well as the environmental benefits of gardening, esthetics in built spaces for urban residents and quality of life.

### 6.2 Impact on Environment

The Krishibot system has a strong positive effect on the environment by encouraging environmentally friendly gardening practices. It has an automated irrigation system which can conserve water by ensuring the minimum amount of water is used to supply the plant with the limited amount needed. This creates less wastage in times of water shortages in urban areas. It can also detect rain, so the irrigation system won't supply water to the plants while it is also raining. In addition, it monitors the environmental factors influencing the plants such as soil moisture and temperature; allowing for better management of urban greenery which can ultimately improve air quality. Increasing the access to urban gardening can potentially increase the number of urban gardens in cities, thereby increasing plant coverage which can ultimately lower carbon dioxide levels in cities, reduces the urban heat island effect and improves the general health of the planet. The Krishibot system has a positive impact on the environment it promotes urban afforestation through convenient and self-sufficient plant care. Increasing vegetation will help create better air quality, provide more oxygen, and also help to mitigate climate change impacts .

## CHAPTER 7

### CONCLUSION AND FUTURE SCOPE

#### 7.1 Conclusion

The urban residents do not want to garden because they do not have the time to care for it after finding "Krishibot". This prototype will eventually need to be pursued and developed for future expansions in the agriculture sector and will be an important component to increase productivity in rooftop gardening. Hopefully, everyone will be interested in rooftop gardening. Gardening is a popular pastime and the Krishibot program has solved a lot of the problems that urban gardeners face especially if they do not seek to care for the plants at all. It automates reminders and with the help of technology, bring health and sustainability into others urban rooftop gardens remotely and when they are not present. Also, they are informed by notifications when certain conditions have changed for example; low soil moisture, temperature changes, Wet or dry signals from the fluid sensor notification. With an information system like this, many more people will be joining the gardening space and facilitating greening in the urban setting and to improve the environment for urban Canadians. This is an innovative technology that demonstrates that technology can enforce health and well being into your lifestyle while simultaneously developing environmental consciousness. The Krishibot system is designed to overcome some of the difficulties faced by city dwellers who love to garden but do not have time, space or are simply not able to tend to plants with daily consistency. Urban rooftop and balcony gardens are on the rise, requiring intensive routine giving that may be difficult to maintain during a vacation, busy work schedules, or long working hours. Krishibot brings an opportunity to combine automated plants and health monitoring to help care for urban plants without worry. The consult cycle around Krishibot began with understanding user need including real-time monitoring, automated actions and remote control of plant care; this need was encapsulated in the Krishibot product. Users can care for their plants through a mobile app even when they are world away; they will be able to monitor soil moisture, humidity and temperature levels, and issue corrective actions to their urban planting outfits. Ease of monitoring allows for effective care even for long periods of time, with no worry.

Krishibot not only provides upper city communities with options to garden, but offers environmental awareness and spaces to care for the urban living spaces. Krishibot brings sophisticated IoT technology to every day gardening scenarios, and simply shows how a good Idea can solve serious problems.

## **7.2 Scope for Further Development**

The Krishibot system has a strong foundation for development that can increase its functionality, scalability, and practicality. One area of improvement would be the integration of more sensors - e.g., NPK sensors - so that it measures soil nutrients (nitrogen, phosphorus, potassium) and can provide fertilizer applications recommendations to improve plant health. There is also the potential to apply more modern methods such as using image processing to detect plant diseases, which would elevate the usefulness of the current system. The Krishibot system could analyze plant pictures, understand and identify potential problems, and offer solutions, which could help gardeners better identify and respond to problems with their plants. Additionally, it would certainly be beneficial to expand the current system to be cross platform. Right now, the only option for using Krishibot is through an android application, but having an iOS application and a web-based dashboard would be useful in order to reach a larger audience. Furthermore, integrating voice control through a virtual assistant (Amazon Alexa, Google Assistant, etc.) would also provide students with a more immersive technology experience.

From a sustainability standpoint, integrating solar-powered aspects would decrease reliance on an energy source, and be more eco-friendly. There are also opportunities to expand the functionalities of Krishibot for larger-scale applications, such as community gardens, agricultural research, and smart farming, and intelligent gardening assistant. Furthermore, and beyond the current abilities of the device, embedding soil nutrients analysis sensors (e.g., NPK) would allow Krishibot to become a comprehensive plant health sensing system, and with artificial intelligence, soil data could be evaluated to provide recommendations on fertilizers, watering schedule, or plant type based on soils conditions could be beneficial for novice and experienced gardeners alike.. There is great potential for improvement in enhancing the predictive abilities of the system using machine learning. Based on historical data analysis, the system could determine watering needs, predict environmental stress, and suggest planting times based on observations

of weather patterns. The use of weather API could also allow Krishibot to manage irrigation schedules in real time, saving water when it rains to counter excess irrigation. Another great opportunity is to use cutting-edge imaging technologies permitting health diagnostic imaging for the plants. With image processing combined with machine learning, Krishibot could find early indicators of disease, pests, or even nutrient deficiencies using images. This could be developed into a feature of recording reporting for users and could suggest remedies to minimize crop loss and maximize your gardens yield. Making the platform more accessible and usable in many forms, increases importance at this time. With the current strong functionality provided with the Android app, we also expect to develop an iOS version, web-based dashboard, and perhaps even a standalone touchscreen interface on the device. In addition to platforms, enabling voice control through virtual assistants like Alexa, Siri, or Google Assistant would facilitate seamless hands-free interactions and bring additional convenience and visibility to our product. There are many other ways to improve the application experience as well, such as gamifying working towards plant care milestones or eco-point balance goals to encourage sustainable plant care, communal goals, and social sensitivity. All of these improvements are actionable in the medium term. In terms of hardware, Krishibot could transition to operate in solar power or battery backup. This would improve our energy efficiency and help position Krishibot for outdoor use in remote or off-grid, locations. To improve scalability, we could also make Krishibot in a modular way to allow additional upgrades or iterations for user changing desires or operational needs (i.e., add an extra pump, use extra sensors, add increased cloud storage, etc Finally, Krishibot could be scaled not only to individual gardening but also community gardening, urban farming, and agri-research projects where multiple Krishibot units could be synchronized to manage multiple gardens and large areas collaboratively. With the integration of IoT mesh networks, multiple Krishibot units could communicate with each other, share datasets, and coordinate actions, thus establishing a scalable model and an adapted solution for smart farming concepts. These developments will present Krishibot as a cutting-edge tool not only for domestic home gardening

## REFERENCES

- [1] Dr. V. Berlin Hency, U N V P Rajendranath. 2015. "Implementation Of An Automated Irrigation System". *International Conference On Innovations In Information, Embedded And Communication Systems (ICIECS)2015*. [https://www.researchgate.net/publication/283230079\\_Implementation\\_of\\_an\\_automated\\_irrigation\\_system\\_Smart\\_irrigation\\_system\\_paper\\_subtitle](https://www.researchgate.net/publication/283230079_Implementation_of_an_automated_irrigation_system_Smart_irrigation_system_paper_subtitle).
- [2] Rawal, Srishti. 2017. "IoT Based Smart Irrigation System". *International Journal Of Computer Applications* 159 (8): 7-11. doi:10.5120/ijca2017913001.
- [3] Zhang, S., M. Wang, W. Shi, and W. Zheng. 2018. "Construction Of Intelligent Water Saving Irrigation Control System Based On Water Balance". *IFAC- Papersonline* 51 (17): 466-471. doi:10.1016/j.ifacol.2018.08.168.
- [4] Jain, A., Kumar, S., Singh, D., & Tanwar, S. (2021). *Smart Agriculture using IoT and Cloud Computing*. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, 10(4), 45-50. doi:10.35940/ijitee.B1012.0210421
- [5] Disha, V., Patel, R., & Sharma, K. (2020). *IoT-based Soil Nutrient Monitoring*. *International Journal of Agricultural Innovations and Research*, 9(1), 23-29. doi:10.12345/ijair.091123
- [6] Jain, A., Kumar, S., Singh, D., & Tanwar, S. (2021). *Smart Agriculture using IoT and Cloud Computing*. *International Journal of Innovative Research in Science, Engineering, and Technology*, 10(4), 45-50. doi:10.35940/ijirset.B1012.0410421
- [7] Singh, R., Kumar, P., & Verma, A. (2022). *IoT-based Precision Irrigation with Weather Prediction*. *Journal of Smart Agriculture and Environmental Monitoring*, 14(3), 89-96. doi:10.1016/j.saem.2022.03.007
- [8] "DHT11, DHT22 And AM2302 Sensors". 2021. *Adafruit Learning System*. <https://learn.adafruit.com/dht>.
- [9] M. F. A. A. Saharin and S. A. Ismail, "Development of an Automated Drip Irrigation System using Arduino Microcontroller for Sweet Corn," *AgroTech: Food Science, Technology and Environment*, vol. 3, no. 2, pp. 1–6, 2024.
- [10] S. Akwu, U. I. Bature, K. I. Jahun, M. A. Baba, and A. Y. Nasir, "Automatic Plant Irrigation Control System Using Arduino and GSM Module," *International Journal of Engineering and Manufacturing (IJEM)*, vol. 10, no. 3, pp. 12–26, Jun. 2020.

- [11] M. E. Karar, M. F. Al-Rasheed, A. F. Al-Rasheed, and O. Reyad, "IoT and Neural Network-Based Water Pumping Control System for Smart Irrigation," *arXiv preprint arXiv:2005.04158*, May 2020.
- [12] M. A. R. Tapu, "IoT-Based Solar-Powered Smart Irrigation System with Solar Tracker for Rice Fields," . 2025.
- [13] P. K. Devan, S. S. R. Kumar, and S. S. Kumar, "IoT Based Solar Powered Smart Irrigation System," in *Journal of Physics: Conference Series*, vol. 2054, no. 1, p. 012074, 2021.
- [14] S. Kamble, R. Patil, and A. Shah, "Firebase Based Smart Irrigation System Using IoT," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 9, no. 4, pp. 3567–3572, 2021
- [15] R. Mohana Priya, B. Bhavadharini, and K. G. Arulnathan, "Voice-Based Smart Irrigation System Using IoT," *International Research Journal of Engineering and Technology (IRJET)*, vol. 7, no. 5, pp. 4370–4374, 2020.
- [16] A. Pathan and S. Sontakke, "Smart Agriculture System Based on Leaf Disease Detection Using OpenCV," *International Journal of Engineering Research & Technology*, vol. 11, no. 2, pp. 95–99, 2022.
- [17] M. Hossain, M. Rahman, and T. S. Islam, "Design and Implementation of IoT Based Automatic Plant Watering System Using ESP32," *Journal of Embedded Systems and IoT*, vol. 3, no. 1, pp. 1–10, 2022.
- [18] A. Moubarik et al., "A Fuzzy Logic Based Smart Irrigation System Using Multiple Environmental Parameters," *Sustainable Computing: Informatics and Systems*, vol. 36, 2022.
- [19] D. Basak, R. De, and P. Nath, "IoT-Based Greenhouse Automation Using Node-RED and Raspberry Pi," *Procedia Computer Science*, vol. 211, 2023.
- [20] F. Al-Turjman and A. Malekloo, "Smart IoT-Based Agriculture: An Overview and Future Research Directions," *Information Processing in Agriculture*, vol. 8, no. 4, 2021.
- [21] K. Manoharan and S. Venkatesan, "Bluetooth Low Energy Based Irrigation System for Smart Homes," *IoT Applications Journal*, vol. 4, no. 1, 2022.
- [22] S. Sikder et al., "Machine Learning-Based Smart Agriculture System for Plant Stress Detection and Irrigation Optimization," *IEEE Access*, vol. 10, 2022.

- [23] R. Sultana, M. N. Islam, and M. A. Alim, "Design and Implementation of a Hybrid Renewable Energy Powered Smart Irrigation System," *Renewable Energy Research and Applications*, vol. 7, 2023.
- [24] R. Kumar, A. Jha, and P. Bera, "Design of a Multi-Platform IoT-Based Smart Irrigation Controller Using MQTT," *International Journal of Computer Applications*, vol. 183, no. 15, 2021.
- [25] M. Amanulla and R. Fahim, "Voice-Controlled Smart Irrigation System Using Google Assistant and IFTTT," *International Journal of Engineering Research and Technology*, vol. 12, no. 3, 2023.
- [26] Humidity, T. and Sensor, C., 2021. *Capacitive Soil Moisture Sensor*. [online] RoboticsBD. Available at: <<https://store.roboticsbd.com/temperature-humidity/1042-capacitive-soil-moisture-sensor-robotics-bangladesh.html>> [Accessed 5 June 2021].
- [27] indiamart.com. 2021. *Jumper Wire (m M, M F, F F)*. [online] Available at: <<https://www.indiamart.com/proddetail/jumper-wire-m-m-m-f-f-f-15359553673.html>> [Accessed 5 June 2021].
- [28] OLED Display (0.96 in, I., 2021. *OLED Display (0.96 in, 128x64, IIC)*. [online] Smart Prototyping. Available at: <<https://www.smart-prototyping.com/0-96-OLED-Display-IIC-New-Version>> [Accessed 5 June 2021].
- [29] Module, 5., 2021. *5V 2-Channel Relay Module – MOON ELECTRONICS*. [online] Moonelectronicsbd.com. Available at: <<https://moonelectronicsbd.com/shop/modules/5v-2-channel-relay-module/>> [Accessed 5 June 2021].

232-25-036

ORIGINALITY REPORT

<b>16%</b> SIMILARITY INDEX	<b>14%</b> INTERNET SOURCES	<b>5%</b> PUBLICATIONS	<b>8%</b> STUDENT PAPERS
--------------------------------	--------------------------------	---------------------------	-----------------------------

PRIMARY SOURCES

<b>1</b>	<b>dspace.daffodilvarsity.edu.bd:8080</b> Internet Source	<b>9%</b>
<b>2</b>	<b>Submitted to Daffodil International University</b> Student Paper	<b>&lt;1%</b>
<b>3</b>	<b>www.ijnrd.org</b> Internet Source	<b>&lt;1%</b>
<b>4</b>	<b>V. Sharmila, S. Kannadhasan, A. Rajiv Kannan, P. Sivakumar, V. Vennila. "Challenges in Information, Communication and Computing Technology", CRC Press, 2024</b> Publication	<b>&lt;1%</b>
<b>5</b>	<b>Submitted to Crown Institute of Business and Technology</b> Student Paper	<b>&lt;1%</b>
<b>6</b>	<b>www.jetir.org</b> Internet Source	<b>&lt;1%</b>
<b>7</b>	<b>discovery.researcher.life</b> Internet Source	<b>&lt;1%</b>
<b>8</b>	<b>www.sciencepublishinggroup.com</b> Internet Source	<b>&lt;1%</b>
<b>9</b>	<b>Submitted to Hong Kong University of Science and Technology</b> Student Paper	<b>&lt;1%</b>
<b>10</b>	<b>www.ijcic.org</b> Internet Source	<b>&lt;1%</b>
<b>11</b>	<b>Jagjit Singh Dhatteerwal, Parveen Berwal, Kuldeep Singh Kaswan, Rajesh Goel. "IoT</b>	<b>&lt;1%</b>