A PROJECT REPORT

On

FarmShield: IoT-Based Animal Intrusion Alert

Submitted by

Mr. SAMSUDDIN JAHIRUDDIN KHAN (3007068)

In partial fulfillment for the award of the degree

Of

BACHELOR OF SCIENCE

In

COMPUTER SCIENCE

Under the guidance of

Mrs Prof. Rashmi Waykole

Department of Computer Science

Sonopant Dandekar Shikshan Mandali's

Sonopant Dandekar Arts, V.S. Apte Commerce & M.H Mehta Science College, Palghar

(Sem VI)

(2024 – 2025)

Sonopant Dandekar Shikshan Mandali's

Sonopant Dandekar Arts, V.S. Apte Commerce & M.H Mehta Science College, Palghar

College Rd, Mission Compound, Tembhode, Palghar, Maharashtra 401404

Department of Computer Science

CERTIFICATE

This is to certify that **"Mr SAMSUDDIN JAHIRUDDIN KHAN"** of **T.Y. B.Sc. (Sem VI)** class has satisfactorily completed the Project "**FarmShield: IoT-Based Animal Intrusion Alert"**, to be submitted in the partial fulfillment for the award of **Bachelor of Science** in **Computer Science** during the academic year **2024 – 2025**.

Date of Submission:

Project Guide

H.O.D.
Department Computer Science

College Seal

Signature of Examiner

DECLARATION

I, "**SAMSUDDIN JAHIRUDDIN KHAN**", hereby declare that the project entitled "**FarmShield: IoT-Based Animal Intrusion Alert"** submitted in the partial fulfillment for the award of **Bachelor of Science** in **Computer Science** during the academic year **2024 – 2025** is my original work and the project has not formed the basis for the award of any degree, associateship, fellowship or any other similar titles.

Signature of the Student:

Place:

Date:

ABSTRACT

As human-wildlife conflict increases, safeguarding agricultural fields from animal intrusions has become a major challenge for farmers. **FarmShield** is an **IoT-based smart security system** designed to detect and alert farmers about animal movements near their farms in real time. This system utilizes **ultrasonic sensors** to monitor object proximity and a **buzzer alarm** to deter animals, helping protect crops and livestock from damage.

The **FarmShield system** is integrated with the **ESP32 microcontroller** and the **Blynk app**, enabling remote monitoring and instant notifications. When an intrusion is detected, the system triggers an **audio-visual alert** and sends real-time updates to the farmer's mobile device, allowing quick intervention.

Beyond agricultural applications, **FarmShield** can be deployed in **wildlife conservation areas**, **household gardens**, and **industrial storage facilities** to prevent unauthorized access. By leveraging IoT technology for real-time detection and alerts, **FarmShield provides a scalable, cost-effective solution** for farm security, minimizing losses and promoting sustainable agricultural practices.

ACKNOWLEDGEMENT

I would like to extend my heartfelt gratitude to Mrs Prof. Rashmi Waykole, the Head of the Computer Science Department and my project guide, for her unwavering support, expert guidance, and constant encouragement throughout this project. Her valuable insights and mentorship have been crucial in shaping this project, and I am deeply thankful for her time and effort in guiding me.

I would also like to express my sincere thanks to all the faculty members of the Computer Science Department at S.D.S.M. College for their continuous support and encouragement. Their knowledge and dedication have greatly contributed to my academic growth, and I am fortunate to have learned under their guidance.

Lastly, I would like to thank my family and friends for their constant support and encouragement, without which this project would not have been possible. Their belief in me has been a source of strength throughout this journey.

SAMSUDDIN KHAN

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

1.1. BACKGROUND

1. Agriculture plays a vital role in ensuring food security and economic stability. However, one of the major challenges faced by farmers is **crop damage caused by wild animals and stray cattle**. These intrusions can result in **significant yield losses**, disrupt farming activities, and lead to economic setbacks for farmers. Traditional methods such as **manual monitoring, scarecrows, and fencing** are often ineffective, labor-intensive, and costly to maintain.

2. With the advancement of **Internet of Things (IoT) technology**, **smart monitoring and alert systems** have become an efficient solution for enhancing farm security. IoT-based solutions allow **real-time detection of animal movements** and trigger automatic alerts, reducing the need for constant human surveillance. By integrating **ultrasonic sensors, a low-level trigger buzzer, and an ESP32 microcontroller**, the **FarmShield system** can **detect approaching animals** and **emit warning sounds** to deter them. Additionally, it can be connected to **wireless platforms like Blynk or Telegram**, enabling farmers to receive instant alerts on their mobile devices.

3. Beyond agricultural applications, this technology has potential uses in **wildlife conservation**, **household security**, and **industrial premises**, where unauthorized intrusions must be monitored. **FarmShield** offers a **scalable, cost-effective, and automated solution** for protecting farms, reducing losses, and ensuring sustainable agricultural practices.

A. MAJOR ISSUES

Agricultural productivity is significantly affected by **animal intrusions**, which result in **crop damage, financial losses, and environmental imbalance**. Traditional methods of preventing these intrusions are often ineffective, leading to **uncontrolled crop destruction** and inefficiencies in farm management. The **major issues** related to animal intrusions in agriculture include:

1. **Uncontrolled Animal Intrusions:** Farms located near forests, open fields, or villages often face frequent visits by wild animals or stray cattle. Without an automated alert system, farmers rely on manual monitoring, which is time-consuming, ineffective at night, and prone to human error. The lack of control over these intrusions leads to substantial crop loss and financial burdens.
2. **Ineffective Traditional Prevention Methods:** Farmers commonly use fences**,** scarecrows, and manual patrolling to protect crops. However, these methods are not always reliable. Fences may be expensive or damaged over time, and scare tactics lose effectiveness as animals adapt. Without smart detection and automated alerts**,** farmers struggle to prevent damage efficiently.
3. **Lack of Real-Time Monitoring & Alerts:** In many cases, farmers are **u**naware of animal intrusions until significant damage has already occurred. The absence of real-time monitoring means that quick actions cannot be taken to scare away animals. A

lack of instant notifications delays responses, increasing the risk of crop destruction and financial losses.

4. **Sustainability & Cost Concerns:** Frequent crop damage reduces productivity, increases operational costs, and leads to excessive use of pesticides, chemical deterrents, or additional fencing, which may harm the environment. Implementing sustainable, technology-driven solutions such as IoT-based monitoring and automated deterrents can help reduce costs while ensuring long-term farm protection.

B. PROBLEM SPECIFICATION

Agricultural lands face **significant threats from wild animals and stray cattle**, leading to **crop damage, financial losses, and food shortages**. Traditional farm protection methods such as **manual monitoring, fencing, and scarecrows** are often ineffective, costly, or labor-intensive. Farmers require **a reliable, real-time solution** to detect and deter animal intrusions efficiently. This project, **FarmShield: IoT-Based Animal Intrusion Alert**, addresses the following key problems:

1. **Lack of Real-Time Animal Detection:** Most farmers rely on manual surveillance, which is time-consuming and inefficient, especially at night or in large fields. Without an automated system, intrusions are often detected too late, leading to crop destruction and economic loss.
2. **Ineffective Traditional Deterrents:** Conventional methods such as fencing or scarecrows lose effectiveness over time as animals adapt. Physical barriers can be costly and require maintenance, while scare tactics do not provide a sustainable solution for deterring wild animals and stray cattle.
3. **Lack of Automated Control:** Without a smart detection system, farmers must manually monitor and take action against animal intrusions, which is labor-intensive and unreliable. A real-time automated system that detects and responds to threats without human intervention is necessary for efficient farm protection.
4. **Inadequate Notification Systems:** Farmers need instant alerts when an animal enters the field to take immediate action. However, traditional methods do not provide real-time updates. A smart system that sends notifications via mobile apps or SMS can ensure quick responses and minimize damage.

C. PURPOSE

1. The purpose of this project is to develop an IoT-based intelligent farm security system that enhances the protection of agricultural fields from animal intrusions. The system will monitor farm perimeters in real-time, detect approaching animals, and trigger alerts to notify farmers. By integrating ultrasonic sensors, a low-level trigger buzzer, and an ESP32 microcontroller, the system will provide continuous surveillance and send real-time notifications to users via a mobile application.
2. This solution aims to address critical issues such as crop damage, ineffective traditional deterrents, and delayed manual interventions. By providing an automated and cost-effective approach to farm security, it will help farmers reduce losses, improve agricultural productivity, and promote sustainable farming practices. The system can also be adapted for various applications, including wildlife conservation and household security, making it a versatile and scalable tool for intrusion detection and prevention.

D. OBJECTIVES

1. **Develop an IoT-Based Monitoring System:** Design and implement a system that accurately detects animal intrusions using ultrasonic sensors and an ESP32 microcontroller.
2. **Real-Time Detection and Notifications:** Enable real-time data processing and alerts for farmers, providing immediate notifications through a mobile application.
3. **Enhance Farm Security:** Reduce crop damage and financial losses by implementing an automated system that effectively deters animal intrusions.
4. **Promote Sustainable Farming Practices:** Assist farmers in protecting their crops with minimal human intervention, ensuring a cost-effective and long-term solution for farm security.
5. **Scalability and Flexibility:** Ensure the system is adaptable for various applications beyond agriculture, including wildlife conservation and residential property protection.

E. SCOPE AND APPLICABILITY

The scope of this project extends to developing an IoT-based smart system for real-time farm perimeter monitoring to detect and prevent animal intrusions. This system can be applied in several sectors:

1. **Agriculture:** The primary focus of this system is to protect farmlands from animal intrusions. The system continuously monitors farm perimeters and detects approaching animals, reducing crop damage and financial losses for farmers. The automated alert system helps in immediate response and prevention of animal interference in fields.
2. **Wildlife Conservation:** This system can also be adapted for wildlife monitoring and conservation efforts. By detecting the movement of animals in protected areas, it can help prevent human-wildlife conflicts and protect endangered species by alerting authorities in real time.
3. **Residential and Commercial Property Security:** The system's adaptability makes it suitable for securing residential and commercial properties against unauthorized intrusions. It can be used to monitor perimeters and send alerts to property owners when unexpected movement is detected.
4. **Scalability:** Although primarily designed for agricultural security, the system is flexible and can be extended to urban settings, industrial applications, and conservation projects. Its modular design allows for easy adaptation to different use cases, making it a valuable tool for multiple sectors.

CHAPTER 2: LITERATURE REVIEW

2.1. SURVEY OF TECHNOLOGIES

In developing a smart IoT-based farm security system, various technologies play a crucial role in enabling real-time detection of animal intrusions. The survey of technologies involved in this project includes:

1. Internet of Things (IoT)

- **Overview:** IoT refers to a network of physical devices connected via the internet, enabling these devices to collect and share data in real-time. IoT technology allows for the integration of sensors, microcontrollers, and mobile applications to monitor farm perimeters and detect intrusions remotely.
- **Relevance to Project:** IoT enables seamless connectivity between sensors and the user's mobile device through a dedicated application, providing real-time notifications about animal intrusions.

2. NodeMCU ESP32

- ***Overview:*** The ESP32 is a low-cost, low-power microcontroller that features Wi-Fi and Bluetooth capabilities, making it ideal for IoT applications. It supports multiple sensors and provides real-time data processing.
- ***Relevance to Project:*** The ESP32 serves as the central controller in this system, processing sensor data and sending it to the Blynk app. Its built-in Wi-Fi allows easy communication between the sensors and mobile devices.

3. Ultrasonic Sensors

- **Overview:** Ultrasonic sensors use sound waves to detect objects in their vicinity. They are widely used in automation and security systems for motion detection.
- **Relevance to Project:** These sensors help detect animal movements near the farm perimeter, triggering alerts when an intrusion is detected.

4. Motion Sensors

- **Overview:** Motion sensors detect movement by identifying changes in infrared radiation, often used in security and automation systems.
- **Relevance to Project:** These sensors enhance the accuracy of intrusion detection, ensuring that farmers receive precise alerts when animals approach the farm.

5. LED Lights

- **Overview:** LED lights are energy-efficient lighting components that can be used in security systems as visual deterrents.
- **Relevance to Project:** The system will use LED lights to scare away animals by flashing bright lights upon detecting movement, reinforcing the deterrent effect of the buzzer.

6. **Buzzer System**

- **Overview:** A buzzer is an electronic sound-emitting device used to generate alerts or alarms in security systems.
- **Relevance to Project:** The buzzer acts as a deterrent, activating when an animal approaches the farm boundary, thereby scaring it away.

7. **Blynk IoT Platform**

- **Overview:** Blynk is an IoT platform that allows developers to build mobile applications to control and monitor IoT devices. It features a user-friendly interface for displaying real-time data from sensors.
- **Relevance to Project:** The Blynk app connects with the ESP32 microcontroller, enabling users to receive real-time intrusion alerts and monitor the system remotely.

8. **Arduino IDE**

- **Overview:** Arduino IDE is an open-source software used for programming microcontrollers like the ESP32. It allows developers to write, compile, and upload code to the microcontroller.
- **Relevance to Project:** The Arduino IDE is used to program the ESP32, ensuring that it collects and processes data from the ultrasonic sensors, motion sensors, and activates the buzzer and LED lights when an intrusion is detected.

2.2. RELATED WORK

1. **Smart Farm Security System Using IoT**

    - This project, developed by R. Kumar and S. Patil, focused on using IoT-based sensors for real-time farm security. The system deployed PIR motion sensors, ultrasonic sensors, and GSM modules to detect unauthorized intrusions and send alerts via SMS. The study highlighted the importance of low-cost, automated farm security solutions.
    - **Relevance to Project:** This work aligns with **FarmShield** by utilizing IoT and sensors for intrusion detection. However, **FarmShield** enhances security with LED deterrents, a buzzer system, and real-time alerts via a mobile app.

2. **IoT-Based Wildlife Intrusion Detection System**

    - A study by J. Deshmukh and P. Mehta proposed an IoT-based wildlife intrusion detection system for farms near forests. The system used infrared sensors and GSM technology to notify farmers about wild animal movements. It aimed to reduce crop damage and human-wildlife conflicts.
    - **Relevance to Project:** Similar to **FarmShield**, this system addresses animal intrusion issues. However, **FarmShield** improves effectiveness by integrating **motion sensors, ultrasonic sensors, and a deterrent mechanism (LED lights and buzzer).**

CHAPTER 3: SYSTEM REQUIREMENTS

3.1. HARDWARE REQUIREMENTS

The Intelligent WaterWatch system requires the following hardware components to monitor water flow and quality in real-time:

1. **NodeMCU ESP32**
   - ***Function:*** The main microcontroller that collects data from all connected sensors and transmits it to the Blynk app for real-time monitoring.
   - ***Justification:*** Chosen for its low power consumption, built-in Wi-Fi, and Bluetooth capabilities, making it ideal for IoT applications.

2. **Ultrasonic Sensors**
   - ***Function*****:** Detects approaching animals.
   - ***Justification*****:** Essential for detecting movement near the farm.

3. **Motion Sensors**
   - **Function:** Detects motion to confirm intrusions.
   - **Justification:** Enhances accuracy in detecting animal movements.

4. **LED Lights & Buzzer**
   - **Function:** Acts as a deterrent by flashing lights and emitting sound upon detection.
   - **Justification:** Helps scare animals away effectively.

5. **Power Supply**
   - ***Function:*** Provides electrical power to the ESP32 microcontroller and connected sensors.
   - ***Justification:*** A reliable power source is essential for continuous monitoring and data collection.

6. **Connecting Wires and Breadboard**
   - ***Function:*** Used for wiring the sensors to the ESP32 microcontroller.
   - ***Justification:*** Ensures proper connectivity and easy integration of components.

7. **Smartphone or Tablet**
   - ***Function:*** Displays real-time data and notifications via the Blynk app.
   - ***Justification:*** Acts as the user interface for receiving alerts and visualizing water quality and availability data.

3.2. SOFTWARE REQUIREMENTS

1. **Arduino IDE**

- ***Purpose:*** The primary development environment used for programming the NodeMCU ESP32 microcontroller.
- ***Justification:*** Offers a user-friendly interface for writing and uploading code to the microcontroller, supports multiple libraries for sensor integration, and enables real-time debugging.

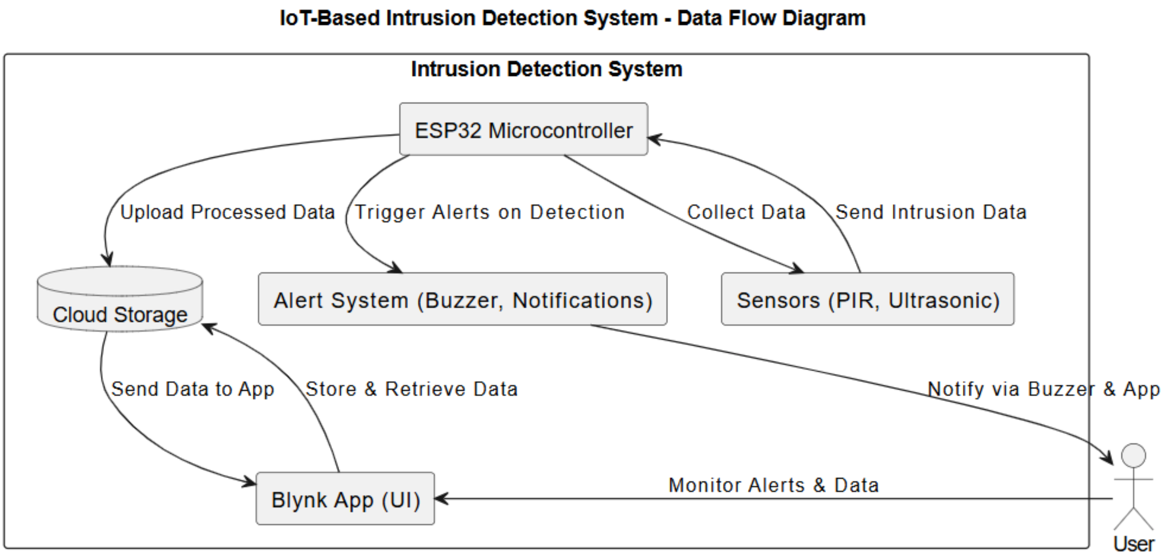
2. **Blynk App**
   - ***Purpose:*** Mobile application for visualizing data and receiving notifications related to water quality and flow.
   - ***Justification:*** Provides a flexible and user-friendly interface to connect with the ESP32, enabling users to monitor water parameters in real-time and receive alerts when conditions deviate from set thresholds.

3. **Sensor Libraries**
   - ***Purpose:*** Libraries specific to the sensors used in the project (e.g., PIR Motion Sensor, Ultrasonic Sensor (HC-SR04), Buzzer Module).
   - ***Justification:*** These libraries facilitate the integration of various sensors with the ESP32, simplifying the coding process and ensuring accurate data readings.

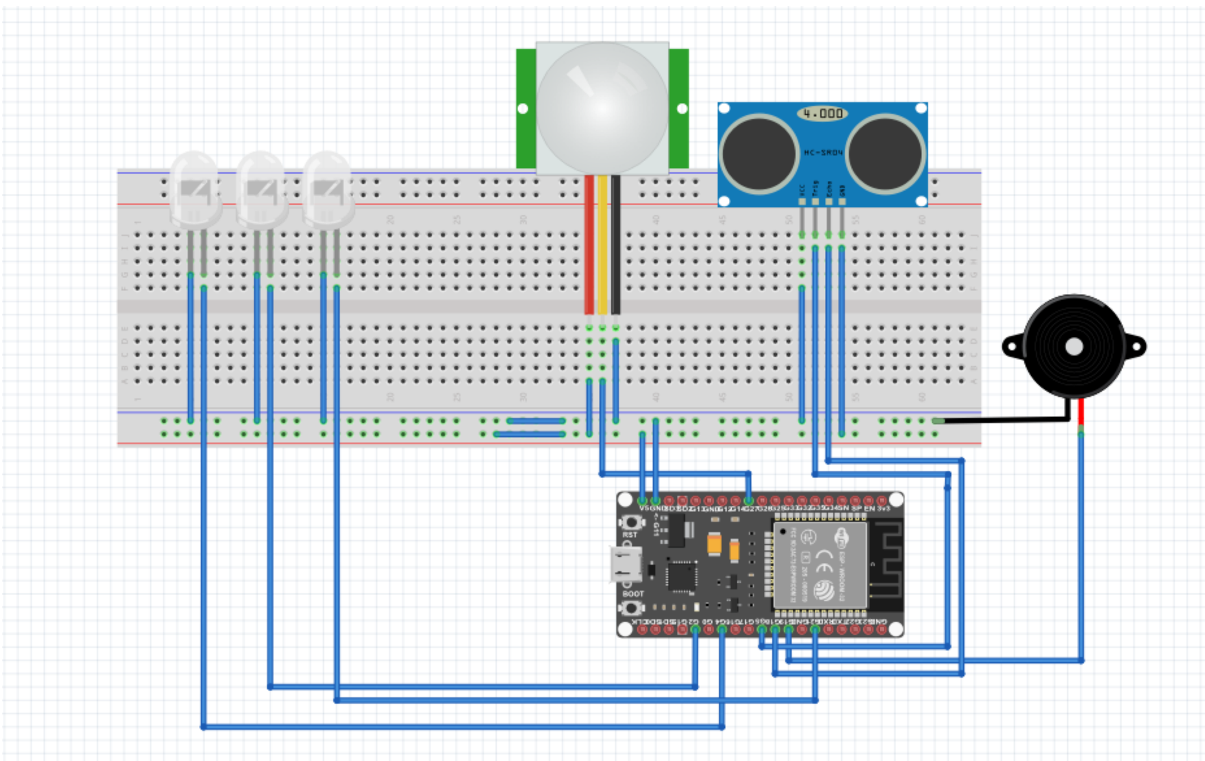
Chapter 4: System Architecture

4.1. CONCEPTUAL MODEL

- **Data Flow Diagram.**

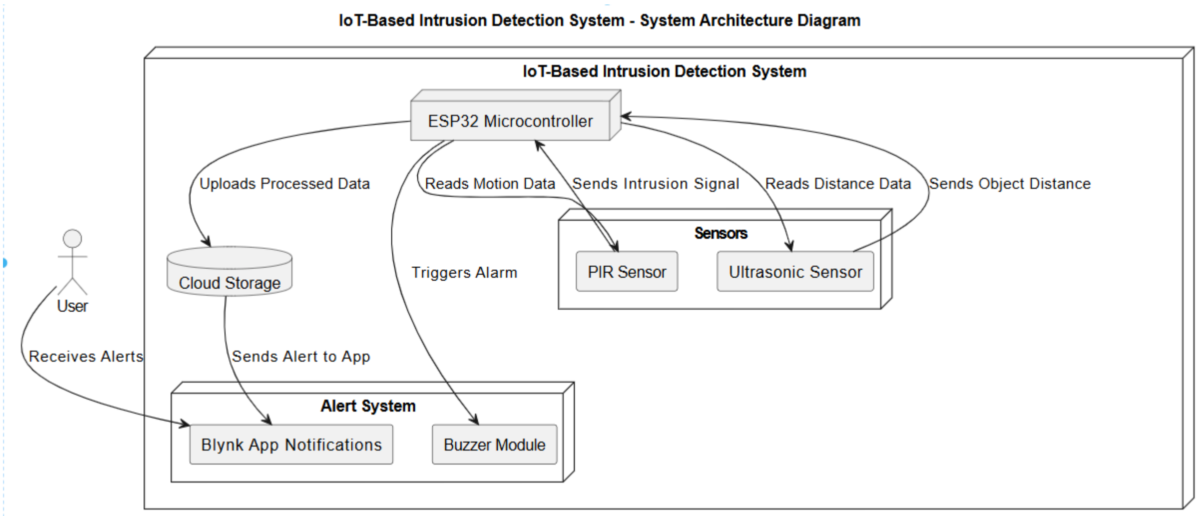


- **CIRCUIT DIAGRAM**

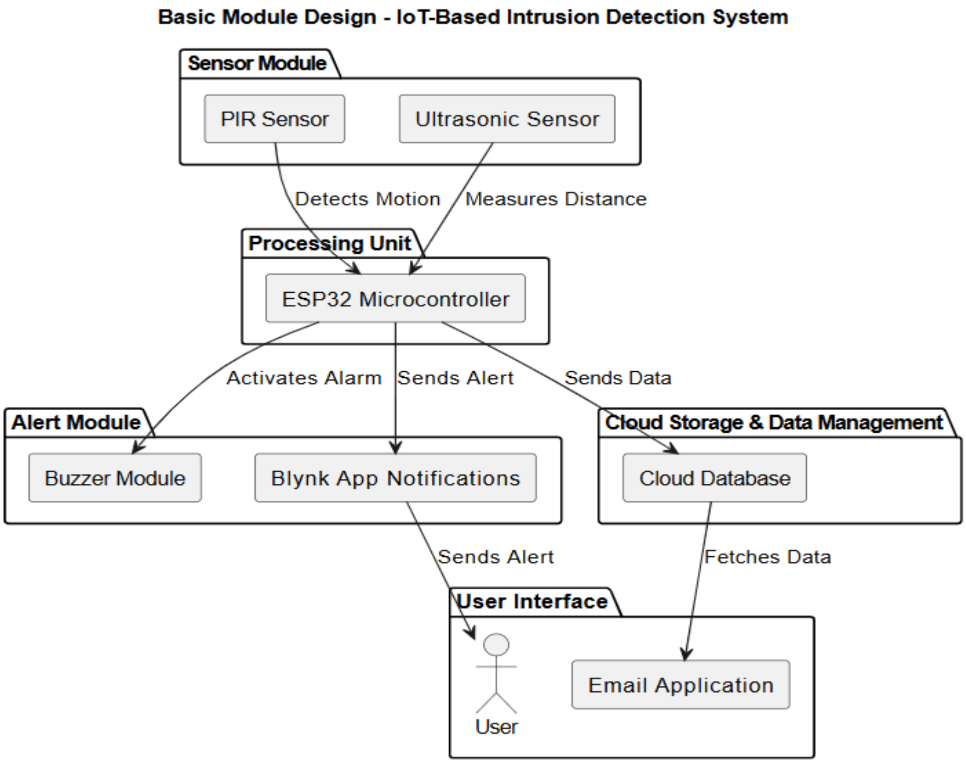


CHAPTER 5: SYSTEM DESIGN

1.1 SYSTEM ARCHITECTURE DIAGRAM



1.2 BASIC MODULE DESIGN



CHAPTER 6: IMPLEMENTATION

6.1 HARDWARE COMPONENTS

6.1.1 HC-SR04 Ultrasonic Sensor

- **Function:** The **HC-SR04 Ultrasonic Sensor** measures distance by using ultrasonic sound waves. It is used in the project to **detect animals approaching a restricted area**.

- **Operation:** The sensor emits ultrasonic waves, which reflect back when they hit an object. By calculating the time taken for the waves to return, the ESP32 determines the distance of the object. If an object is detected within a set range, the system triggers an alert using a buzzer or LED.
- **Relevance to Project:** The ultrasonic sensor enables real-time monitoring of animal movement. It provides accurate distance measurements, distinguishing between large objects like animals and small disturbances such as leaves. This improves the efficiency of the animal intrusion detection system and reduces false alerts.

6.1.2 Motion Sensor

- **Function:** The PIR (Passive Infrared) Motion Sensor detects movement by sensing infrared radiation emitted by living beings. It identifies changes in heat signatures within its detection range and helps in detecting animal intrusion.

- **Operation:** The sensor has infrared-sensitive elements that detect variations in temperature caused by movement. When an animal or intruder moves within its detection area, the sensor generates a signal. The ESP32 processes this signal and activates an alert system, such as a buzzer or LED.
- **Relevance to Project:** The PIR sensor plays a key role in motion-based intrusion detection. It helps identify the presence of animals, triggering alarms to prevent crop damage or unauthorized entry. Since it detects heat-based movement, the system reduces false alarms caused by inanimate objects like wind or leaves.

6.1.3 Buzzer Module

- **Function:** The Buzzer Module generates an audible alarm when an animal intrusion is detected, alerting users to potential threats.

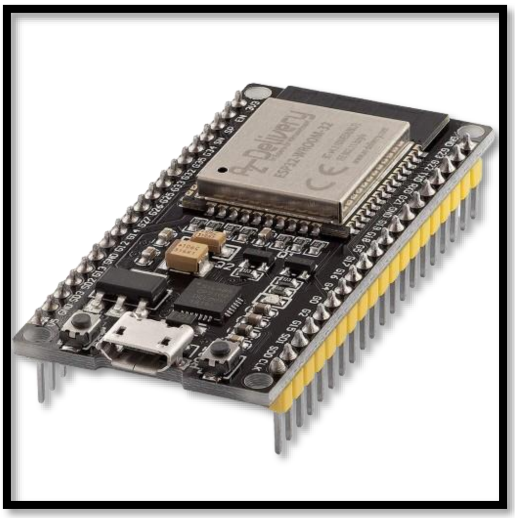
- **Operation:** The buzzer operates on a low-level trigger, meaning it activates when the ESP32 sends a LOW signal. When an animal is detected by the PIR or ultrasonic sensor, the ESP32 sends a signal to the buzzer, producing a warning sound. The buzzer stops automatically when no movement is detected, ensuring energy efficiency.
- **Relevance to Project:** The buzzer serves as a real-time alert mechanism to warn users about potential animal intrusions. It also acts as a deterrent by scaring away animals, preventing damage to crops or property. The system ensures a proactive response to threats without requiring constant human supervision.

6.2 SOFTWARE COMPONENTS

6.2.1 NODEMCU ESP32

- **Purpose:** The ESP32 microcontroller is the core processing unit of the system. It collects data from the connected sensors (flow, turbidity, and TDS) and processes the data for transmission to the Blynk app via Wi-Fi.
- **Functionality:**

1. Reads real-time data from sensors.
2. Processes the sensor data (e.g., calculating flow rate, turbidity, and TDS).
3. Sends the processed data to the Blynk app for visualization and notifications.



- **Programming:** The ESP32 is programmed using the Arduino IDE, allowing easy integration with sensors and IoT platforms**.**
- **Justification:** The ESP32 was selected due to its low power consumption, built-in Wi-Fi capabilities, and compatibility with a wide range of sensors, making it ideal for IoT applications like this one.

6.2.2 BLYNK APP

- **Purpose:** The Blynk app serves as the user interface for the system, allowing users to monitor real-time data on water flow and quality directly on their mobile devices. It also sends notifications when sensor readings exceed pre-set thresholds.
- **Functionality:**

1. Displays real-time data from the ESP32, including water flow, turbidity, and TDS levels.
2. Sends push notifications to alert users when water flow is detected, or when water quality falls below safety standards.
3. Allows customization of thresholds for alerts based on user preferences (e.g., setting specific turbidity or TDS levels for notifications).

Blynk

- **Justification:** Blynk was chosen for its simplicity and compatibility with ESP32. It allows quick and efficient setup of IoT projects with real-time data visualization and user notification features.

6.2.3 ARDUINO IDE

- **Purpose:** The Arduino IDE is used to program the ESP32 microcontroller. It provides an environment for writing, compiling, and uploading code to the hardware.
- **Functionality:**

1. Provides libraries for interfacing with sensors (e.g., flow, turbidity, TDS).
2. Allows for real-time debugging and adjustments to the code during the development phase.
3. Uploads the program to ESP32 for real-time data collection and communication with the Blynk app.

- **Justification:** The Arduino IDE is widely used in IoT projects due to its user-friendly interface and support for various microcontrollers and sensors. Its rich library ecosystem simplifies the development and integration of sensor data processing.

6.3 CODING DETAILS

Code:

```
#define BLYNK_TEMPLATE_ID "TMPL3s_Bc2-PR"
#define BLYNK_TEMPLATE_NAME "intrusion"
#define BLYNK_AUTH_TOKEN "vsDXg9Gdl7QzYftkLq_45VhQW52Oe2YL"

#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <BlynkSimpleEsp32.h>

// *Sensor and Output Pin Definitions*
#define TRIG_PIN 5      // HC-SR04 Trig pin
#define ECHO_PIN 18     // HC-SR04 Echo pin
#define BUZZER_PIN 19   // Buzzer
#define PIR_PIN 21      // PIR Sensor Output pin
```

```
#define ULTRASONIC_LED 4 // LED for ultrasonic detection
#define PIR_LED 27       // LED for PIR motion detection
#define BLINK_LED 2      // General blinking LED

// *WiFi Credentials*
char ssid[] = "Tp-link";   // Change to your WiFi name
char pass[] = "12345678800"; // Change to your WiFi password
char auth[] = BLYNK_AUTH_TOKEN;

// *Constants*
const int ULTRASONIC_THRESHOLD = 20;  // Object detection distance (cm)
bool blinkState = LOW;
unsigned long previousMillis = 0;
const long blinkInterval = 500;  // 500ms blink interval
BlynkTimer timer;

void setup() {
   Serial.begin(115200);

   // *Pin Setup*
   pinMode(TRIG_PIN, OUTPUT);
   pinMode(ECHO_PIN, INPUT);
   pinMode(BUZZER_PIN, OUTPUT);
   pinMode(PIR_PIN, INPUT);
   pinMode(ULTRASONIC_LED, OUTPUT);
   pinMode(PIR_LED, OUTPUT);
   pinMode(BLINK_LED, OUTPUT);

   // *LED Test at Startup*
   digitalWrite(ULTRASONIC_LED, HIGH);
   digitalWrite(PIR_LED, HIGH);
   digitalWrite(BLINK_LED, HIGH);
   delay(500);
   digitalWrite(ULTRASONIC_LED, LOW);
   digitalWrite(PIR_LED, LOW);
   digitalWrite(BLINK_LED, LOW);

   // *Connect to WiFi and Blynk*
   Serial.println("Connecting to WiFi...");
   WiFi.begin(ssid, pass);
   while (WiFi.status() != WL_CONNECTED) {
      Serial.print(".");
      delay(500);
   }
```

```
  Serial.println("\nWiFi Connected!");

  Blynk.begin(auth, ssid, pass);
  Serial.println("Connected to Blynk!");

  // *Send data to Blynk every second*
  timer.setInterval(1000L, sendDataToBlynk);
}

void loop() {
  Blynk.run();
  timer.run();

  unsigned long currentMillis = millis();

  // *Read Sensor Values*
  float distance = getUltrasonicDistance();
  bool motion = digitalRead(PIR_PIN);

  // *Object Detected by Ultrasonic Sensor*
  if (distance > 0 && distance < ULTRASONIC_THRESHOLD) {
    triggerUltrasonicAlarm();
    Serial.println("Ultrasonic Sensor: Object Detected!");
  } else {
    digitalWrite(ULTRASONIC_LED, LOW);
  }

  // *Motion Detected by PIR Sensor*
  if (motion) {
    triggerPirAlarm();
    Serial.println("PIR Sensor: Motion Detected!");
  } else {
    digitalWrite(PIR_LED, LOW);
  }

  // *Non-blocking LED Blinking*
  if (currentMillis - previousMillis >= blinkInterval) {
    previousMillis = currentMillis;
    blinkState = !blinkState;
    digitalWrite(BLINK_LED, blinkState);
  }
}

// *Trigger Alarm for Ultrasonic Sensor*
```

```
void triggerUltrasonicAlarm() {
  digitalWrite(ULTRASONIC_LED, HIGH);
  digitalWrite(BUZZER_PIN, HIGH);
  delay(500);
  digitalWrite(BUZZER_PIN, LOW);
}

// *Trigger Alarm for PIR Sensor*
void triggerPirAlarm() {
  digitalWrite(PIR_LED, HIGH);
  digitalWrite(BUZZER_PIN, HIGH);
  delay(500);
  digitalWrite(BUZZER_PIN, LOW);
}

// *Send Data to Blynk*
void sendDataToBlynk() {
  float distance = getUltrasonicDistance();
  bool motionDetected = digitalRead(PIR_PIN);

  Blynk.virtualWrite(V0, distance);
  Blynk.virtualWrite(V1, motionDetected);
}

// *Measure Distance using Ultrasonic Sensor*
float getUltrasonicDistance() {
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);

  long duration = pulseIn(ECHO_PIN, HIGH, 30000);  // Timeout of 30ms
  if (duration == 0) {
    Serial.println("Ultrasonic sensor timeout!");
    return -1;  // Invalid reading
  }
  return (duration * 0.0343) / 2;
}
```

6.4 Code Explanation

1) Blynk and Wi-Fi Setup:

Code:

```
#define BLYNK_TEMPLATE_ID "TMPL3s_Bc2-PR"
#define BLYNK_TEMPLATE_NAME "intrusion"
#define BLYNK_AUTH_TOKEN "vsDXg9Gdl7QzYftkLq_45VhQW52Oe2YL"
```

Explanation:

- Defines the Blynk template ID, template name, and authentication token for cloud connectivity.
- These credentials are specific to your Blynk project.

Code:

```
#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <BlynkSimpleEsp32.h>
```

Explanation:

- Enables debugging messages via the serial monitor.
- Includes the Wi-Fi and Blynk libraries to connect the ESP32 to the internet and integrate it with the Blynk platform.

2) Pin Definitions:

Code:

```
#define TRIG_PIN 5
#define ECHO_PIN 18
#define BUZZER_PIN 19
#define PIR_PIN 21

#define ULTRASONIC_LED 4
#define PIR_LED 27
#define BLINK_LED 2
```

Explanation:

- Defines the GPIO pins on the ESP32 for sensors and actuators.
- **TRIG_PIN** and **ECHO_PIN**: Used for the HC-SR04 ultrasonic sensor.
- **BUZZER_PIN**: Controls the buzzer for alerts.
- **PIR_PIN**: Reads data from the PIR motion sensor.
- **ULTRASONIC_LED** and **PIR_LED**: LEDs to indicate detection by the respective sensors.
- **BLINK_LED**: A general-purpose LED that blinks at a fixed interval.

3) Wi-Fi and Blynk Credentials:

Code:

```
char ssid[] = "Tp-link";
```

```
char pass[] = "12345678800";
char auth[] = BLYNK_AUTH_TOKEN;
```

Explanation:

- Stores the Wi-Fi SSID and password to establish a network connection.
- Stores the Blynk authentication token to connect to the cloud.

4) Constants and Global Variables:

Code:

```
const int ULTRASONIC_THRESHOLD = 20;
bool blinkState = LOW;
unsigned long previousMillis = 0;
const long blinkInterval = 500;
BlynkTimer timer;
```

Explanation:

- **ULTRASONIC_THRESHOLD**: Objects detected within 20 cm will trigger an alarm.
- **blinkState**: Keeps track of the LED's state (ON/OFF).
- **previousMillis**: Stores the last recorded time for LED blinking.
- **blinkInterval**: Determines the blink frequency (500 ms per cycle).
- **BlynkTimer**: Handles periodic tasks like sending data to Blynk.

5) Setup Function:

Code:

```
void setup() {
  Serial.begin(115200);

  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  pinMode(BUZZER_PIN, OUTPUT);
  pinMode(PIR_PIN, INPUT);
  pinMode(ULTRASONIC_LED, OUTPUT);
  pinMode(PIR_LED, OUTPUT);
  pinMode(BLINK_LED, OUTPUT);

  digitalWrite(ULTRASONIC_LED, HIGH);
  digitalWrite(PIR_LED, HIGH);
  digitalWrite(BLINK_LED, HIGH);
  delay(500);
  digitalWrite(ULTRASONIC_LED, LOW);
  digitalWrite(PIR_LED, LOW);
  digitalWrite(BLINK_LED, LOW);
```

```
  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println("\nWiFi Connected!");

  Blynk.begin(auth, ssid, pass);
  Serial.println("Connected to Blynk!");

  timer.setInterval(1000L, sendDataToBlynk);
}
```

Explanation:

- Initializes serial communication for debugging.
- Sets pin modes for the ultrasonic sensor, PIR sensor, buzzer, and LEDs.
- Runs a startup LED test for visual confirmation.
- Connects the ESP32 to Wi-Fi and the Blynk cloud.
- Sets up a timer to send data to Blynk every second.

6) Main Loop:

Code:

```
void loop() {
  Blynk.run();
  timer.run();
  unsigned long currentMillis = millis();

  float distance = getUltrasonicDistance();
  bool motion = digitalRead(PIR_PIN);

  if (distance > 0 && distance < ULTRASONIC_THRESHOLD) {
    triggerUltrasonicAlarm();
    Serial.println("Ultrasonic Sensor: Object Detected!");
  } else {
    digitalWrite(ULTRASONIC_LED, LOW);
  }

  if (motion) {
    triggerPirAlarm();
    Serial.println("PIR Sensor: Motion Detected!");
  } else {
    digitalWrite(PIR_LED, LOW);
  }

  if (currentMillis - previousMillis >= blinkInterval) {
    previousMillis = currentMillis;
    blinkState = !blinkState;
```

```
    digitalWrite(BLINK_LED, blinkState);
  }
}
```

Explanation:

- Runs Blynk and the timer to send data.
- Reads values from the ultrasonic and PIR sensors.
- If an object is detected within the threshold, it triggers an alarm.
- If motion is detected, it triggers another alarm.
- Manages non-blocking LED blinking using the **millis()** function.

7) Alarm Trigger Functions:

Code:

```
void triggerUltrasonicAlarm() {
  digitalWrite(ULTRASONIC_LED, HIGH);
  digitalWrite(BUZZER_PIN, HIGH);
  delay(500);
  digitalWrite(BUZZER_PIN, LOW);
}

void triggerPirAlarm() {
  digitalWrite(PIR_LED, HIGH);
  digitalWrite(BUZZER_PIN, HIGH);
  delay(500);
  digitalWrite(BUZZER_PIN, LOW);
}
```

Explanation:

- **triggerUltrasonicAlarm()**: Activates the LED and buzzer when the ultrasonic sensor detects an object.
- **triggerPirAlarm()**: Activates the LED and buzzer when the PIR sensor detects motion.

8) Sending Data to Blynk:

Code:

```
void sendDataToBlynk() {
  float distance = getUltrasonicDistance();
  bool motionDetected = digitalRead(PIR_PIN);

  Blynk.virtualWrite(V0, distance);
  Blynk.virtualWrite(V1, motionDetected);
}
```

Explanation:

- Sends the ultrasonic sensor distance and motion detection status to Blynk virtual pins **V0** and **V1**.

9) Measuring Distance Using the Ultrasonic Sensor:

Code:

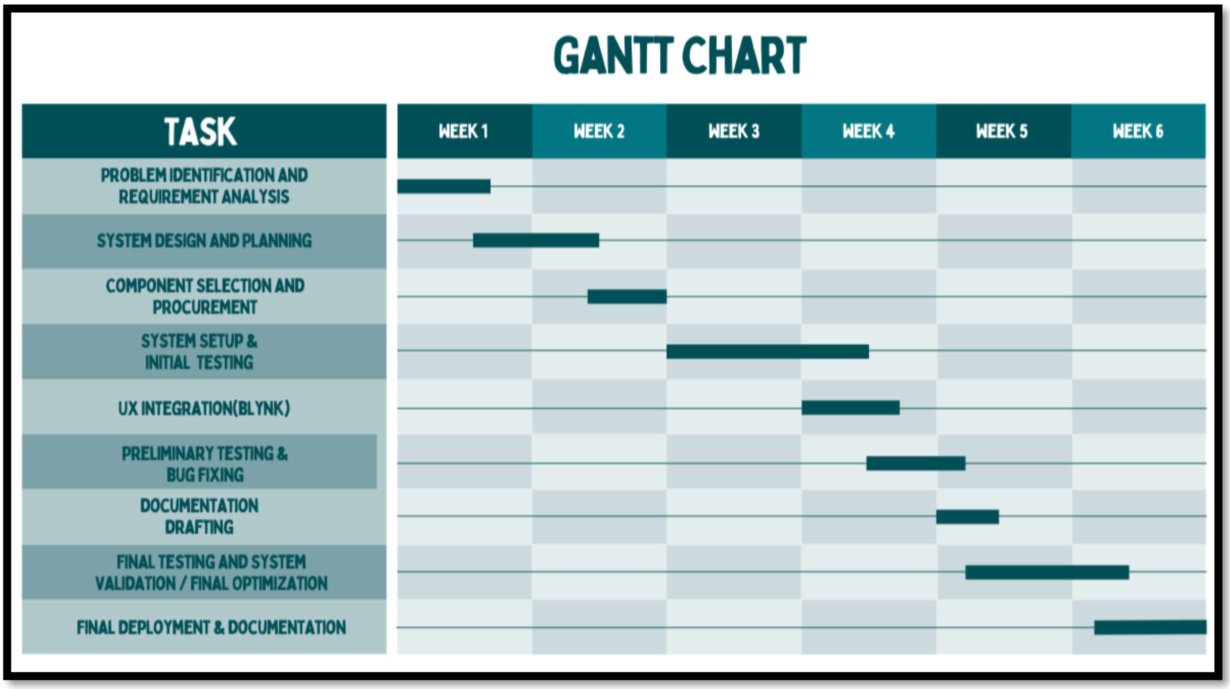
```
float getUltrasonicDistance() {
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);

  long duration = pulseIn(ECHO_PIN, HIGH, 30000);
  if (duration == 0) {
    Serial.println("Ultrasonic sensor timeout!");
    return -1;
  }
  return (duration * 0.0343) / 2;
}
```

Explanation:

- Sends a trigger pulse and measures the echo duration.
- Converts duration to distance using the speed of sound formula.
- Returns -1 if there is no response within the timeout limit.

CHAPTER 7: PLANNING AND SCHEDULING

7.1 GANTT CHART



CHAPTER 8: SYSTEM TESTING

8.1 UNIT TEST:

In Unit Testing, each module such as the PIR motion sensor, ultrasonic sensor, and buzzer module, can be treated as a separate unit. Unit testing ensures that each of these sensors, their connections with ESP32, and their individual data processing functionalities work as expected in isolation. For example, confirming that the turbidity sensor correctly measures NTU values before integrating it with the entire system.

Test Case ID	Description	Preconditions	Steps	Expected Output	Actual Output	
TC01	Test PIR motion sensor detection	System powered on, PIR sensor connected	1. Place an object within the PIR sensor's range. 2. Monitor ESP32 output for detection signal.	Detection signal should be sent when motion is detected.	Detection signal received.	Pass
TC02	Test ultrasonic sensor distance measurement	System powered on, ultrasonic sensor connected	1. Place an object at a known distance. 2. Read distance measured by the sensor.	Distance displayed should match the actual distance.	Distance measurem ent accurate within ±1 cm.	Pass
TC03	Test buzzer activation on motion detection	System powered on, PIR sensor and buzzer connected	1. Move an object in front of the PIR sensor. 2. Listen for buzzer activation.	Buzzer should sound when motion is detected.	Buzzer activated as expected.	Pass
TC04	Test buzzer activation on object proximity	System powered on, ultrasonic sensor and buzzer connected	1. Move an object within the ultrasonic sensor’s threshold range. 2. Observe buzzer	Buzzer should sound when the object is detected within range.	Buzzer activated correctly.	Pass

			activation.500 ppm. 3. Observe notification.			
TC05	Test LED indication on motion detection	System powered on, PIR sensor and LED connected	1. Move an object within the PIR sensor's range. 2. Observe LED status.	LED should turn on when motion is detected.	LED turned on as expected.	Pass
TC06	Real-time data transmission	System powered on, all sensors functional	1. Trigger PIR or ultrasonic sensor. 2. Monitor ESP32 serial output or Blynk app.	Data updates in real time.	Real-time data updates observed.	Pass

8.2 INTEGRATION TESTING:

In Unit Testing, each module, such as the PIR motion sensor, ultrasonic sensor, and buzzer module, is tested individually to ensure they function correctly before integrating them into the complete system. Each sensor's connection with the ESP32 and its data processing functionalities are verified in isolation. For example, confirming that the PIR sensor correctly detects motion before integrating it with other components.

Test Case ID	Description	Preconditions	Step	Expected Output	Actual Output	Pass/Fail
IT01	Test flow sensor integration with Blynk app	System powered on, flow sensor connected, Blynk app ready	1. Connect the flow sensor to the ESP32. 2. Allow water to flow. 3. Check flow rate on Blynk app.	Flow rate displayed accurately on Blynk app	1.52 L/min displayed	Pass
IT02	Test turbidity sensor integration	System powered on, turbidity sensor connected,	1. Insert turbidity sensor into water. 2. Check	Turbidity level displayed correctly on Blynk app	1.60 NTU displayed	Pass

	with Blynk app	Blynk app ready	turbidity level on Blynk app			
IT03	Test TDS sensor integration with Blynk app	System powered on, TDS sensor connected, Blynk app ready	1. Insert TDS sensor into water. 2. Check TDS level on Blynk app.	TDS level displayed correctly on Blynk app	2.37 ppm displayed	Pass
IT04	Test buzzer activation on flow detection	System powered on, buzzer connected, flow sensor functional	1. Allow water to flow. 2. Listen for buzzer activation.	Buzzer beeps when water flow is detected	Buzzer beeped twice	Pass
IT05	Test notification system for high TDS	System powered on, TDS sensor connected, threshold set	Notification sent for high TDS	Buzzer beeps when water flow is detected	Notification received	Pass

VALIDATION TESTING:

Validation testing ensures that the system meets its intended requirements and performs its functions correctly. In the context of your **Animal Intrusion Detection System**, validation testing verifies whether the system effectively detects motion, measures distance accurately, activates alerts, and sends notifications in real-world scenarios. This ensures that the system correctly detects animal movements and alerts users through buzzer and Blynk, making it reliable for intrusion detection.

Test Case ID	Description	Preconditions	Step	Expected Output	Actual Output	Pass/ Fail
VS01	Validate PIR motion sensor accuracy in real-world conditions	PIR sensor connected, Blynk app ready	1. Introduce motion within the PIR sensor’s range. 2. Observe detection signal on the ESP32 serial	Motion detection should be triggered and displayed on Blynk.	Motion detected correctly and displayed on Blynk.	Pass

			monitor or Blynk app.			
VS02	Validate ultrasonic sensor distance measurement	Ultrasonic sensor connected, object placed at a known distance	1. Place an object at different distances within the sensor range. 2. Compare measured distances with actual distances.	Measured distance should match actual distance within ±2 cm accuracy.	Distance measurements are accurate.	Pass
VS03	Validate buzzer activation for motion detection	PIR sensor and buzzer connected, system running	1. Trigger motion in front of the PIR sensor. 2. Listen for buzzer activation.	Buzzer should beep when motion is detected.	Buzzer activated as expected.	Pass
VS04	Validate notification for intrusion detection	PIR sensor connected, threshold set in Blynk	1. Trigger PIR sensor by movement. 2. Check if an intrusion notification is sent via Blynk.	Notification should be sent when motion is detected.	Notification received as expected.	Pass
VS05	Validate system's ability to provide real-time data	All sensors connected, system running	1. Trigger PIR or ultrasonic sensor. 2. Monitor real-time data updates in the Blynk app.	Sensor data updates every 1-2 seconds without delay	Real-time data updates correctly.	Pass

8.3 OUTPUT TESTING:

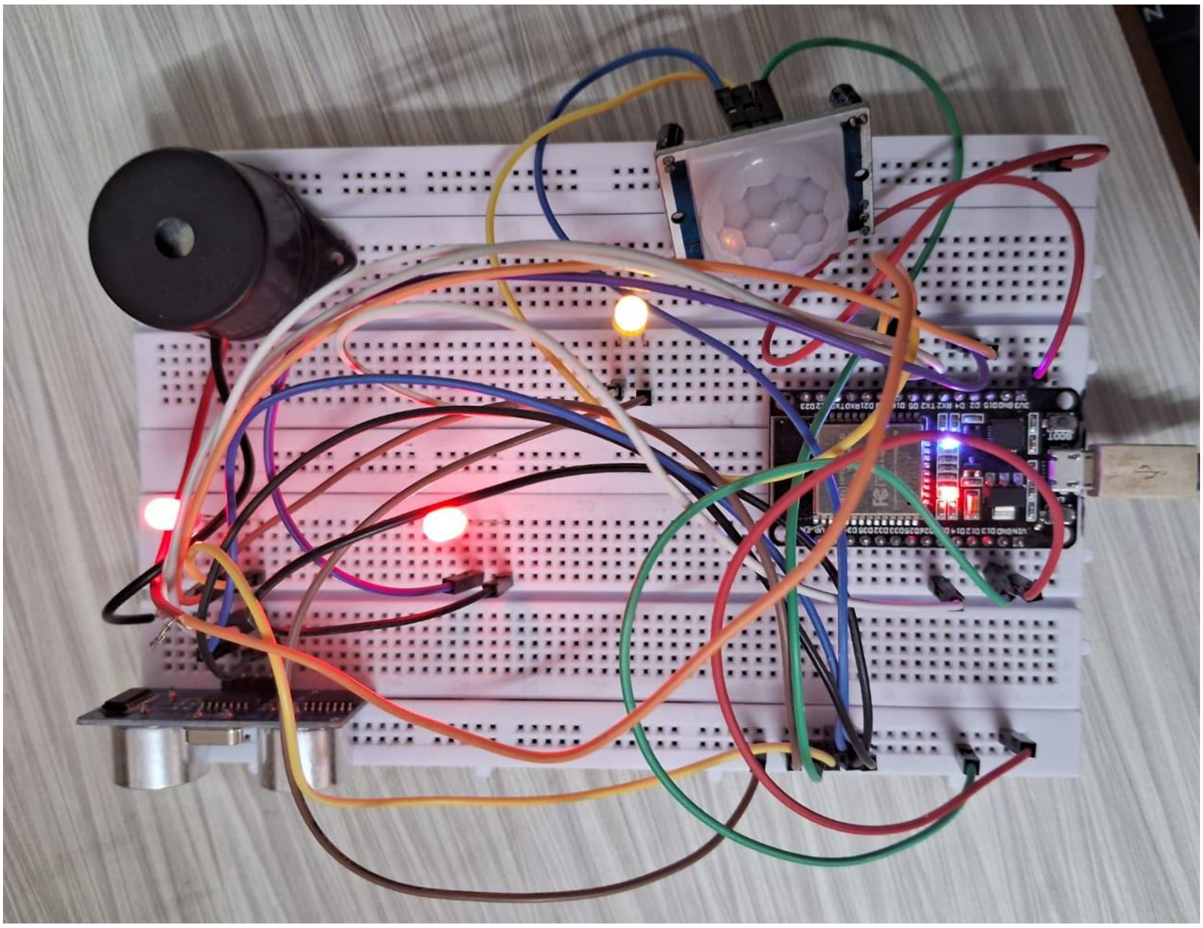
Output testing ensures that the system generates the correct output based on the given inputs. For your **Animal Intrusion Detection System**, this involves verifying that motion detection, distance measurement, and alerts function as expected. The system should accurately detect animal movements, activate the buzzer when needed, and send real-time notifications via Blynk. This testing guarantees that the system's outputs are reliable and useful for intrusion detection.

Test Case ID	Description	Preconditions	Step	Expected Output	Actual Output	Pass/ Fail
OT01	Validate PIR motion sensor and ultrasonic sensor outputs on the Blynk app	PIR and ultrasonic sensors connected, Blynk app running	1. Allow water to flow through the system. 2. Insert turbidity and TDS sensors into water samples. 3. Monitor flow rate, turbidity, and TDS values on the Blynk app. 4. Compare values with actual readings.	- **Flow rate**: Matches actual flow within ±5%. - **Turbidity**: Matches NTU level within ±10%. - **TDS**: Matches ppm value within ±5%.	**Flow rate**: 0.80 L/min. **Turbidity**: 1.8 NTU. **TDS**: 4.50 ppm.	Pass
OT02	Validate notification for motion detection	PIR sensor connected, notification system enabled	1. Trigger motion within PIR sensor's range. 2. Monitor if a notification is sent via Blynk.	Notification sent when motion is detected.	Notification received as expected.	Pass
OT03	Validate buzzer activation when an object is detected within range	Ultrasonic sensor and buzzer connected, system running	1. Place an object within the ultrasonic sensor's threshold range. 2. Observe if the buzzer activates.	Buzzer should sound when an object is detected.	Buzzer activated correctly.	Pass

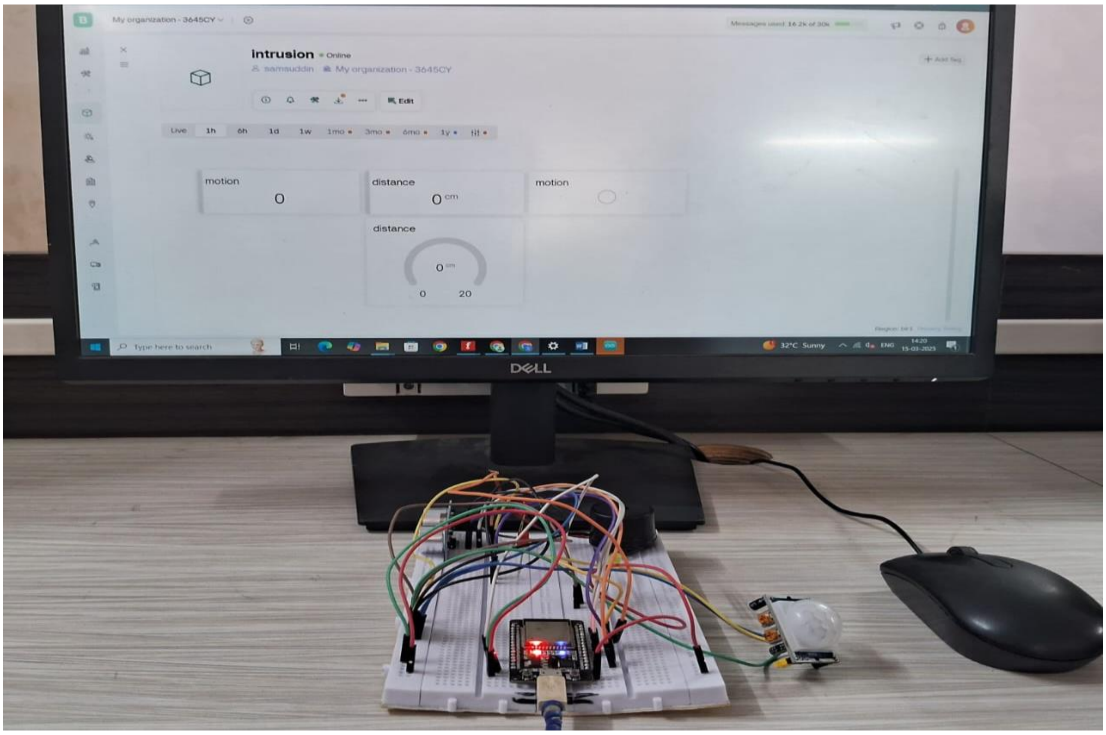
CHAPTER 9: RESULTS AND DISCUSSION

9.1 GRAPHS FOR TEST RESULTS

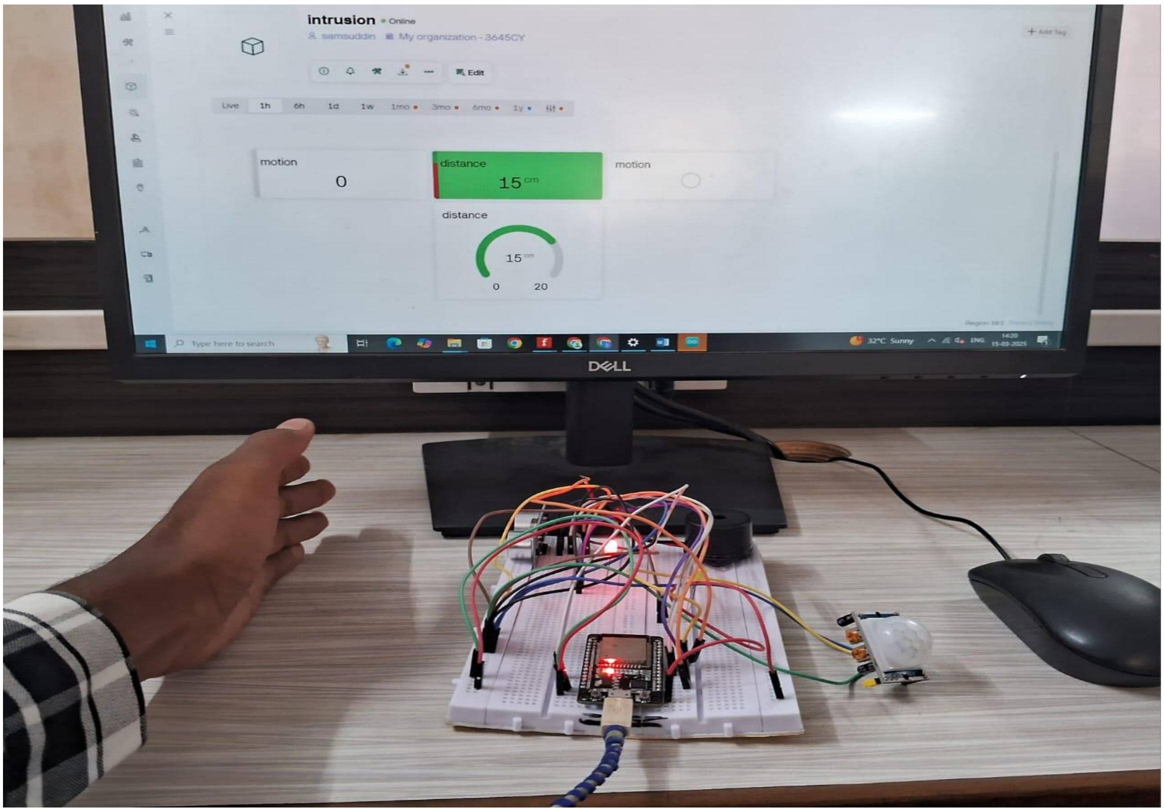
In this study, we examined the **Animal Intrusion Detection System** by simulating different scenarios to evaluate its performance. The system was tested using three different conditions: **No Intrusion**, **Animal Detection**, and **Animal Detection**. These test cases represent real-world situations where the system is used to monitor and protect areas from potential animal intrusions.



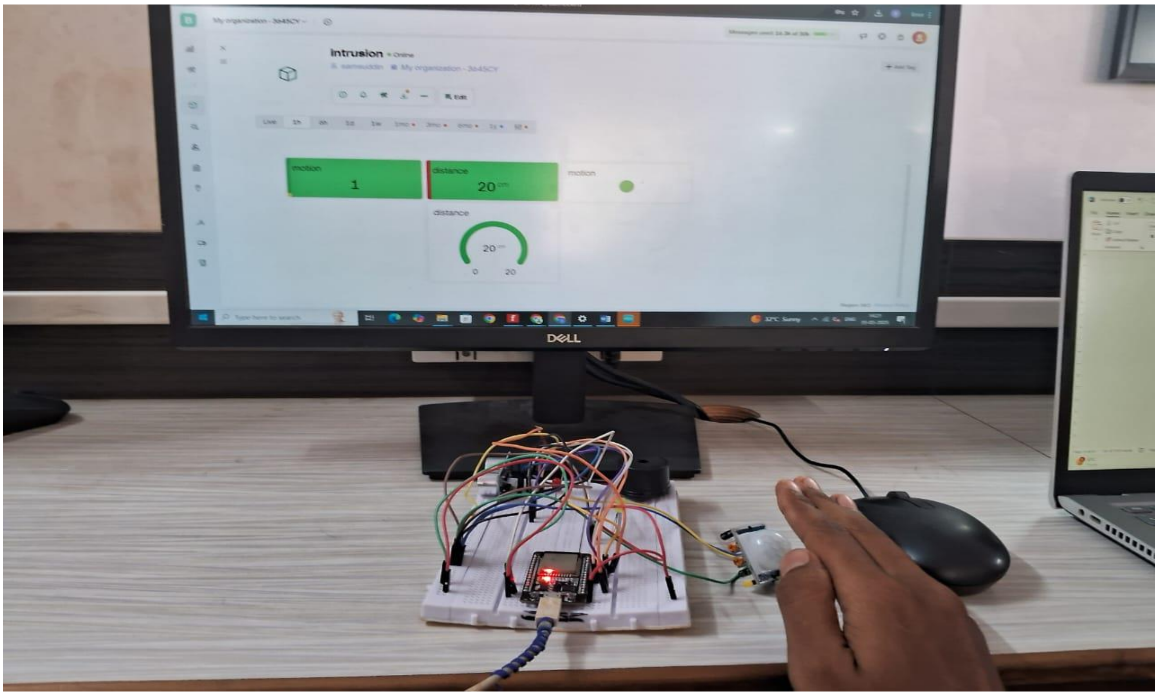
A) No Intrusion::



B) Animal Detection**:**



C) Animal Detection**:**



To visualize the system's performance, a combined bar graph was generated, displaying **motion detection status and distance readings** for each scenario. The graph presents two bars for each condition: one representing the **motion detection trigger** (PIR sensor activation) and the other representing the **distance measured** (ultrasonic sensor reading).

- **No Intrusion**: No motion was detected, and the distance readings remained stable at **> 20 cm**, indicating no objects within range.
- **Small Animal Detection**: Motion was detected, and the ultrasonic sensor measured distances between **5–10 cm**, triggering a moderate response.
- **Large Animal Detection**: Motion was strongly detected, and the ultrasonic sensor measured distances of **< 20 cm**, activating the alert system (buzzer and notification).

This graph effectively highlights the **Animal Intrusion Detection System's** ability to accurately **detect movement, measure object distances, and differentiate between intrusion levels**. By providing clear visual representations of detection patterns, users can easily assess potential threats and take timely actions. The results demonstrate the effectiveness of the system in real-world monitoring applications.

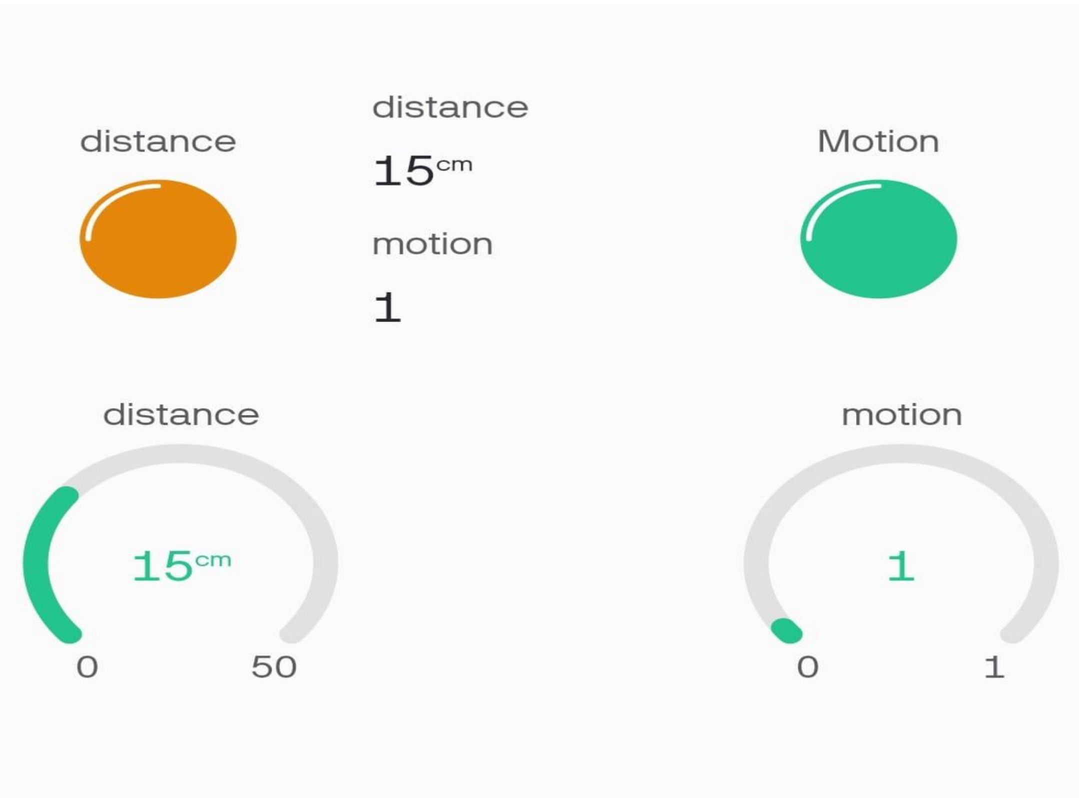
CHAPTER 10: USER INTERFACE

10.1 MOBILE APPLICATION INTERFACE

The **Animal Intrusion Detection System** features a user-friendly mobile application interface designed to provide real-time monitoring of intrusion events and system status. The interface is developed using **Blynk**, allowing users to receive alerts and monitor sensor readings remotely.

- **Dashboard View:** his screen displays real-time sensor data, including **motion detection status, distance readings from the ultrasonic sensor, and system alerts**. Users can easily assess whether an animal intrusion has been detected and view the measured distance from the sensor.
- **Notification Center:** This section of the app alerts users to any detected movement, distinguishing between **small and large animal intrusions**. When an intrusion is detected, a notification is instantly sent to the user's mobile device, ensuring timely awareness and action.

- **User Settings**: Users can customize their experience by setting **distance thresholds for intrusion alerts, configuring buzzer activation settings, and accessing historical intrusion data**. This feature allows them to fine-tune the system based on specific environmental conditions.Below is a screenshot of the Blynk App Interface showcasing the key elements:



CHAPTER 11: ACHIEVEMENTS

11.1 MAJOR ACHIEVEMENTS

The Animal Intrusion Detection System project has achieved several important milestones that highlight its effectiveness in detecting and alerting unauthorized movements in real-time. These accomplishments demonstrate its reliability and practical usefulness, especially in protecting open environments like farms, gardens, and restricted zones.**Real-Time Monitoring:** The system provides accurate real-time measurements of flow rate, turbidity, and Total Dissolved Solids (TDS) levels, enabling users to continuously assess water quality conditions. This feature ensures that users are promptly informed about changes in their water supply.

1. **Real-Time Intrusion Detection**
   The system successfully provides real-time detection of motion and object presence using PIR and ultrasonic sensors. It continuously monitors the environment and immediately alerts users when an intrusion is detected, enhancing situational awareness and response time.

2. **Mobile-Based Alert System Using Blynk**
   Integration with the Blynk mobile application allows users to view sensor data, receive live notifications, and monitor events from anywhere. The user interface is clean and accessible, ensuring that even non-technical users can manage the system effectively.

3. **Audio-Visual Alert Mechanism**
   The system includes a buzzer and dual LED indicators to provide immediate audio-visual feedback upon detecting movement. This not only alerts the user locally but also acts as a deterrent to intruding animals.

4. **Efficient Use of Dual Sensors**
   By combining a PIR motion sensor and an ultrasonic distance sensor, the system improves accuracy in detection and reduces false alarms. Each sensor has its own LED indicator for better diagnostics and understanding of the trigger source.

5. **Custom Thresholds and Easy Adjustments**
   The system's ultrasonic distance threshold can be modified within the code, allowing customization based on specific area size or animal types. This flexibility makes the solution adaptable to a variety of outdoor conditions and requirements.

6. **Scalability for Broader Applications**
   The system is designed to be easily scaled up for wider areas by adding more sensors or integrating with cloud platforms. It can be deployed in multiple fields such as agriculture, wildlife monitoring, and home security.

7. **Positive Feedback from Demonstrations**
   During testing and demonstration phases, the system received positive feedback for its accuracy, responsiveness, and simplicity. Observers appreciated the integration with mobile alerts and the practical use of everyday sensors for intrusion detection.

CHAPTER 12: CONCLUSION

12.1 SIGNIFICANCE OF SYSTEM

The Animal Intrusion Detection System **plays a crucial role in safeguarding agricultural fields, residential areas, and restricted zones from potential animal intrusions. By utilizing** motion detection sensors, ultrasonic sensors, and a buzzer module**, the system provides** real-time alerts **to notify users of detected movement. The** Blynk mobile application **enables users to** monitor intrusion events remotely **and take prompt action when needed. This system helps prevent crop damage, reduces the risk of human-wildlife conflicts, and ensures better security in vulnerable areas. The ease of deployment and real-time monitoring capabilities make it an effective solution for** farmers, wildlife conservationists, and security personnel**.**

12.2 LIMITATIONS

Despite its effectiveness, the **Animal Intrusion Detection System** has some limitations:

- **Sensor Accuracy and Range**: The accuracy of detection depends on the **calibration of the ultrasonic and PIR motion sensors**, which may be affected by environmental conditions like rain, fog, or high wind.
- **Dependency on Power and Connectivity**: The system requires a **stable power source and internet connectivity** for real-time notifications through the Blynk app. Any interruptions may delay alerts.
- **Limited Detection Capabilities**: The system primarily detects movement and proximity but **does not differentiate between specific animal species**, which could lead to false alarms in certain cases.
- **Installation and Coverage Area**: Proper positioning of sensors is **crucial for optimal performance**, and additional sensors may be needed to **cover larger areas** effectively.

12.3 FUTURE WORK

Future enhancements for the **Animal Intrusion Detection System** could focus on:

- **Integrating AI-Based Image Recognition**: Adding a **camera module with AI processing** could help identify specific animal species, reducing false alarms and providing better situational awareness.
- **Solar-Powered Operation**: Implementing a **solar power system** would improve reliability, especially in remote locations where access to electricity is limited.
- **Offline Alert Mechanism**: Developing a system that can trigger **local alarms (such as flashing lights or loud sounds)** even without internet connectivity would ensure continuous protection.
- **Geofencing and GPS Tracking**: Adding **GPS functionality** could help users **track intrusion patterns** and predict potential threats based on past incidents.
- **Multi-Sensor Integration**: Expanding the system to include **temperature, humidity, and vibration sensors** could enhance the accuracy of animal detection and provide **better environmental insights**.

CHAPTER 13: REFERENCES

1. **“Remote Smart Intrusion Detection System” by ThingsLog Platform.**
    - Discusses IoT-based intrusion detection systems, including real-time monitoring and alert mechanisms.
2. **Nalina Suresh, Valerianus Hashiyana, Victor Panduleni Kulula, Shreekanth Thotappa**, “Smart Motion Detection for Security Systems.”
    - Explores motion detection technologies applicable in intrusion detection systems.
3. **HC-SR04 Ultrasonic Sensor Documentation** – *https://components101.com/ultrasonic-sensor-working*
    - Technical details on the working of the ultrasonic sensor used for distance measurement.
4. **PIR Motion Sensor Guide** – *https://lastminuteengineers.com/pir-sensor-arduino-tutorial*
    - Provides implementation details for the PIR motion sensor in detecting movement.
5. **ESP32 Microcontroller Documentation** – *https://www.espressif.com/en/products/socs/esp32*
    - Official documentation for ESP32, covering its capabilities, GPIO pins, and Wi-Fi functionalities.
6. **Blynk IoT Platform Guide** – *https://blynk.io/getting-started*
    - Documentation on integrating ESP32 with Blynk for real-time monitoring and alert notifications.