# IIoT and Cloud Computing Project Title

Kaushsal G. CS21B1003

## I. AIM

To develop a Smart Pet Feeder that utilizes IoT technology to automate the pet feeding process.

## II. APPARATUS/SOFTWARE REQUIRED

List of Hardware

- Arduino Uno R3
- Servo Motor
- RFID
- Connecting wires and cables

List of Software

- Arduino IDE for programming the Arduino Uno R3 microcontroller
- PySerial library for serial communication between Arduino and Python
- Google Cloud Platform (GCP) for cloud-based services, Google Cloud APIs for authentication and interaction with Google Cloud services
- Google Sheets API for data visualization

## III. PROCEDURE

Conceptualization and Planning: The project begins with brainstorming ideas and conceptualizing the Smart Pet Feeder, defining its objectives, features, and functionalities. Project requirements are identified, including pet identification, feeding automation, data collection, and user interface design.

Component Selection: Hardware components such as the Arduino Uno R3 microcontroller, RFID reader, servo motor, and power supply are selected based on their compatibility, performance, and reliability. Software tools and libraries, including Arduino IDE, Python, and Google Cloud APIs, are chosen to develop the firmware and interface for the Smart Pet Feeder.

System Design: A detailed system architecture is designed, outlining the integration of hardware and software components, communication protocols, and data flow. The system design includes the pet identification process, feeding automation mechanism, data storage and visualization, and user interaction features.

Hardware Integration: The selected hardware components are assembled and integrated into the Smart Pet Feeder enclosure, ensuring proper connections and physical stability. Wiring and connections are made between the Arduino Uno R3, RFID reader, servo motor, and power supply, following the system architecture design.

Software Development: Firmware for the Arduino Uno R3 microcontroller is developed using Arduino IDE, implementing functionalities for pet identification, feeding automation, and servo motor control. Python scripts are written to establish serial communication between the Arduino and a computer or Raspberry Pi, retrieve pet feeding data from Google Cloud Storage, and visualize data using Google Sheets API.

Cloud Integration: Google Cloud Platform (GCP) services are set up, including Google Cloud Storage for storing pet feeding data and Google Sheets API for data visualization. Authentication with Google Cloud APIs is implemented using service account credentials, ensuring secure access to Google Cloud resources.

Testing and Optimization: Rigorous testing procedures are conducted to verify the functionality and reliability of the Smart Pet Feeder system under various conditions. Iterative optimization is performed based on testing results and user feedback, addressing any issues and improving system performance and usability.

Documentation and Presentation: Comprehensive documentation is prepared, including system architecture diagrams, hardware and software specifications, user manuals, and technical guides. A professional presentation is created to showcase the Smart Pet Feeder project, highlighting its objectives, features, outcomes, and future directions.

### A. Introduction

A solution merging technology and pet care. In our fast-paced world, balancing busy schedules with the well-being of our pets can be a challenge. The Smart Pet Feeder aims to simplify this task by leveraging IoT technology and advanced hardware components. Join to delve into how this innovative device revolutionizes pet feeding, offering convenience and peace of mind for pet owners.

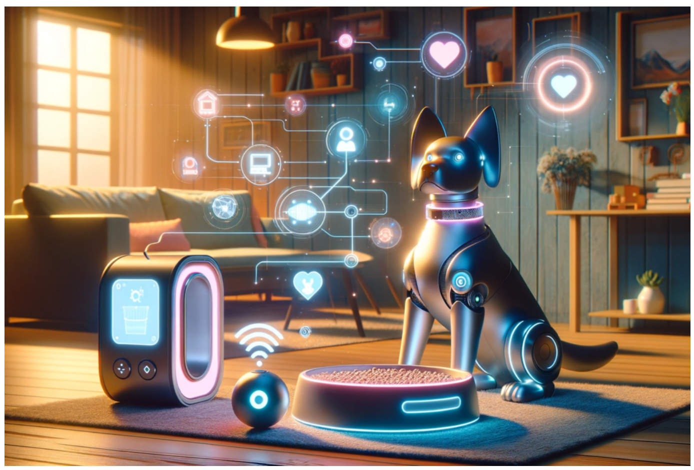


Fig. 1. Technology and Pet Care

### B. Methodology/Problem Statement

Problem Statement: The problem statement revolves around simplifying and enhancing the pet feeding process, ensuring pets receive appropriate meals while providing pet owners with convenient monitoring and control capabilities. Key challenges include pet identification, feeding automation, and data management.

Methodology: The methodology involves integrating hardware components such as Arduino Uno R3, RFID reader, and servo motor to automate the feeding process. Software development using Arduino IDE and Python enables communication between hardware components, data collection, and visualization. Cloud integration with Google Cloud Platform facilitates data storage and analysis. Iterative testing and optimization ensure system reliability and usability.

### C. Implementation

*1) Hardware Design:* The hardware components used in the Smart Pet Feeder project are carefully selected and integrated to ensure seamless functionality and reliability. Here's a detailed description of each hardware component:

1. Arduino Uno R3 Microcontroller: The Arduino Uno R3 serves as the brain of the Smart Pet Feeder, controlling the overall operation of the system. It features a microcontroller unit (MCU), offering sufficient processing power and memory for running the required software functionalities. The Uno R3 provides

various digital and analog input/output pins, which are utilized to interface with other hardware components such as the RFID reader and servo motor.

2. RFID Reader: The RFID (Radio-Frequency Identification) reader is used for pet identification in the Smart Pet Feeder. It consists of an antenna and a reader module capable of detecting RFID tags within its range. When a pet approaches the feeder, the RFID reader scans the unique RFID tag attached to the pet's collar or ID, allowing the system to identify the pet and retrieve its feeding schedule from the database.

3. RFID Tags: RFID tags are small electronic devices containing a unique identifier that is detected by the RFID reader. Each pet is assigned a specific RFID tag, which is attached to its collar or ID. These tags are programmed with pet-specific information, such as feeding schedules, portion sizes, and dietary restrictions, allowing the Smart Pet Feeder to provide personalized feeding experiences.

4. Servo Motor: The servo motor is responsible for dispensing the pet's food from the feeder. It is a type of rotary actuator that allows for precise control of angular position, making it ideal for applications requiring accurate motion control. The servo motor is connected to the Arduino Uno R3 via one of its digital output pins, allowing the microcontroller to control the rotation of the motor shaft and thus the amount of food dispensed.

5. Power Supply: A stable power supply is essential for the reliable operation of the Smart Pet Feeder. The system may be powered using a standard AC adapter or rechargeable batteries, depending on the user's preference and intended usage environment. Adequate voltage regulation and current capacity are ensured to meet the power requirements of all components.

By carefully designing and integrating these hardware components, the Smart Pet Feeder is able to provide reliable and efficient pet feeding solutions.

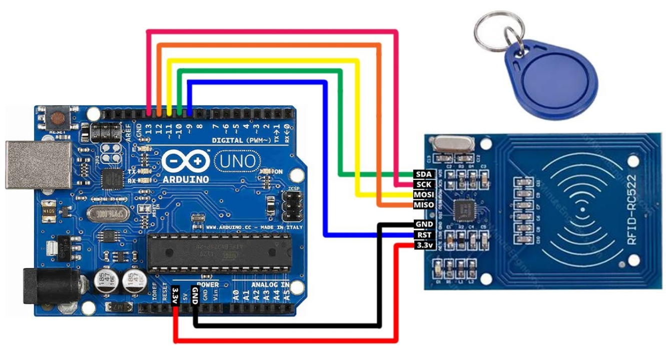



Fig. 2. RFID Arduino Connection

*2) Executable Code:* This code initializes an RFID reader, waits for a compatible tag to be detected, and if the detected tag matches a predefined MasterTag, it activates a servo motor to perform an action like opening a door, and then resets the servo motor position.

```
#include <SPI.h>
#include <MFRC522.h>
#include <Servo.h>

// Define servo motor pin
const int servoPin = 8;

// Create a servo object
Servo servoMotor;
#define SS_PIN 10
#define RST_PIN 9
byte readCard[4];
String MasterTag = "328E824";
```

```
String tagID = "";
// Create instances
MFRC522 mfrc522(SS_PIN, RST_PIN);

/*****************************************************************************
 * setup() function
 *****************************************************************************/
void setup()
{
  Serial.begin(9600);
  SPI.begin();
  mfrc522.PCD_Init();
  delay(4);
  //Show details of PCD - MFRC522 Card Reader
  mfrc522.PCD_DumpVersionToSerial();
  servoMotor.attach(servoPin);
  servoMotor.write(0);
}
/*****************************************************************************
 * loop() function
 *****************************************************************************/
void loop()
{
  //----------------------------------------------------------------------
  //Wait until new tag is available
  while (getID()) {
    if (tagID == MasterTag){
      unsigned long currentMillis = millis();
      Serial.println(currentMillis);
      servoMotor.write(90);
      delay(1000); // Wait for 1 second

      // Rotate the servo back to 0 degrees
      servoMotor.write(0);
      delay(1000);
       //You can write any code here like, opening doors,
       //switching ON a relay, lighting up an LED etc...
    }
    else{

    }
    delay(100);

  }
  //----------------------------------------------------------------------
}
/*****************************************************************************
 * getID() function
 * Read new tag if available
 ****************************************************************************/
boolean getID()
{
  // Getting ready for Reading PICCs
  //If a new PICC placed to RFID reader continue
  if ( ! mfrc522.PICC_IsNewCardPresent()) {
    return false;
  }
```

```
  //Since a PICC placed get Serial and continue
  if ( ! mfrc522.PICC_ReadCardSerial()) {
  return false;
  }
  tagID = "";
  // The MIFARE PICCs that we use have 4 byte UID
  for ( uint8_t i = 0; i < 4; i++) {
  //readCard[i] = mfrc522.uid.uidByte[i];
  // Adds the 4 bytes in a single String variable
  tagID.concat(String(mfrc522.uid.uidByte[i], HEX));
  }

  tagID.toUpperCase();
  mfrc522.PICC_HaltA(); // Stop reading
  return true;
}
```

The following Python script establishes a serial connection with an Arduino device, reads data from the serial port continuously, and prints the received data along with the current timestamp indicating the feeding time. If the script is interrupted by a keyboard interrupt (Ctrl+C), it closes the serial connection.

```
import serial
from datetime import datetime
# Open serial connection with Arduino
ser = serial.Serial('COM5', 9600)  # Change '/dev/ttyUSB0' to the appropriate serial port
try:
    while True:
        # Read data from serial
        data = ser.readline().decode().strip()  # Decode bytes to string and remove trailing newline
        c=datetime.now()
        # Print received data
        print("Feeding time : ", c)
except KeyboardInterrupt:
    # Close serial connection on keyboard interrupt
    ser.close()
    print("Serial connection closed.")
```

The following Python script additionally appends the feeding time data to a Google Sheet named ”petfeeder” using the gspread library. The script utilizes OAuth2 credentials stored in a JSON file named ”petfeederkey.json” for authorization. If the script is interrupted by a keyboard interrupt (Ctrl+C), it closes the serial connection.

```
import gspread
from oauth2client.service_account import ServiceAccountCredentials

# Define the scope and credentials
scope = ['https://spreadsheets.google.com/feeds', 'https://www.googleapis.com/auth/drive']
credentials = ServiceAccountCredentials.from_json_keyfile_name('petfeederkey.json', scope)

# Authorize the client
client = gspread.authorize(credentials)

# Open the Google Sheet by its title
sheet = client.open('petfeeder').sheet1
import serial
```

```
from datetime import datetime
# Open serial connection with Arduino
ser = serial.Serial('COM5', 9600)  # Change '/dev/ttyUSB0' to the appropriate serial port
try:
    while True:
        # Read data from serial
        data = ser.readline().decode().strip()  # Decode bytes to string and remove trailing newline
        c=datetime.now()
        c=str(c)
        d,t=c.split()
        # Print received data
        print("Feeding time : ", c)
        sheet.append_row([d,t])
except KeyboardInterrupt:
    # Close serial connection on keyboard interrupt
    ser.close()
    print("Serial connection closed.")
```

The following Python script integrates an Arduino device with Google Sheets. It reads data from the Arduino via serial communication, which likely indicates feeding times for a pet. Each time data is received, it appends a new row to a specific worksheet named ”petfeeder” in a Google Sheets document.

```
import gspread
from oauth2client.service_account import ServiceAccountCredentials
# Define the scope and credentials
scope = ['https://spreadsheets.google.com/feeds', 'https://www.googleapis.com/auth/drive']
credentials = ServiceAccountCredentials.from_json_keyfile_name('petfeederkey.json', scope)
# Authorize the client
client = gspread.authorize(credentials)
# Open the Google Sheet by its title
sheet = client.open('petfeeder').sheet1
import serial
from datetime import datetime
# Open serial connection with Arduino
ser = serial.Serial('COM5', 9600)  # Change '/dev/ttyUSB0' to the appropriate serial port
try:
    while True:
        # Read data from serial
        data = ser.readline().decode().strip()  # Decode bytes to string and remove trailing newline
        c=datetime.now()
        c=str(c)
        d,t=c.split()
        # Print received data
        print("Feeding time : ", c)
        sheet.append_row([d,t])
except KeyboardInterrupt:
    # Close serial connection on keyboard interrupt
    ser.close()
    print("Serial connection closed.")
```

*3) Software Implementation:* Software Architecture:

Arduino Sketch (Firmware): The Arduino sketch serves as the firmware for the Smart Pet Feeder, running on the Arduino Uno R3 microcontroller. It controls the operation of the feeder, including pet identification, feeding schedule management, and servo motor control. The sketch interacts with hardware components such as the RFID reader and servo motor via GPIO pins.

Python Scripts: Python scripts run on a computer or Raspberry Pi and serve as the interface between the Arduino and Google Cloud services. They handle serial communication with the Arduino to receive pet identification data and feeding instructions. Python scripts authenticate with Google Cloud APIs and interact with Google Cloud Storage and Google Sheets to store and visualize pet feeding data.

Google Cloud Services: Google Cloud Platform (GCP) provides cloud-based services for data storage, computation, and analytics. Google Cloud Storage is used to store pet-related data such as feeding schedules and logs. Google Sheets API is utilized for visualizing pet feeding data in a user-friendly format.

Implementation Details:

Arduino Sketch: The Arduino sketch initializes hardware peripherals and sets up communication protocols. It continuously reads data from the RFID reader to identify pets based on RFID tags. Upon pet identification, the sketch retrieves the corresponding feeding schedule from memory or sends the schedule to Python for retrieval from Google Cloud. The servo motor is controlled to dispense the appropriate amount of food based on the pet.

Python Scripts: Python scripts establish a serial connection with the Arduino using PySerial library. They listen for incoming data from the Arduino, parse pet identification information, and communicate with Google Cloud APIs. Authentication with Google Cloud is achieved using service account credentials stored securely on the device. Python scripts upload pet feeding data to Google Cloud Storage for storage and retrieve feeding schedules as needed. Feeding logs are appended to Google Sheets using the Google Sheets API for visualization.

Google Cloud Integration: Google Cloud Storage buckets are created to store pet feeding data, organized by date or pet ID. The Google Sheets API is enabled and configured to allow Python scripts to interact with Google Sheets. Python scripts authenticate with Google Cloud using service account credentials, ensuring secure access to Google Cloud resources. By implementing this software architecture, the Smart Pet Feeder can effectively manage pet feeding schedules, store feeding data securely in the cloud, and visualize feeding patterns for pet owners' convenience and peace of mind.

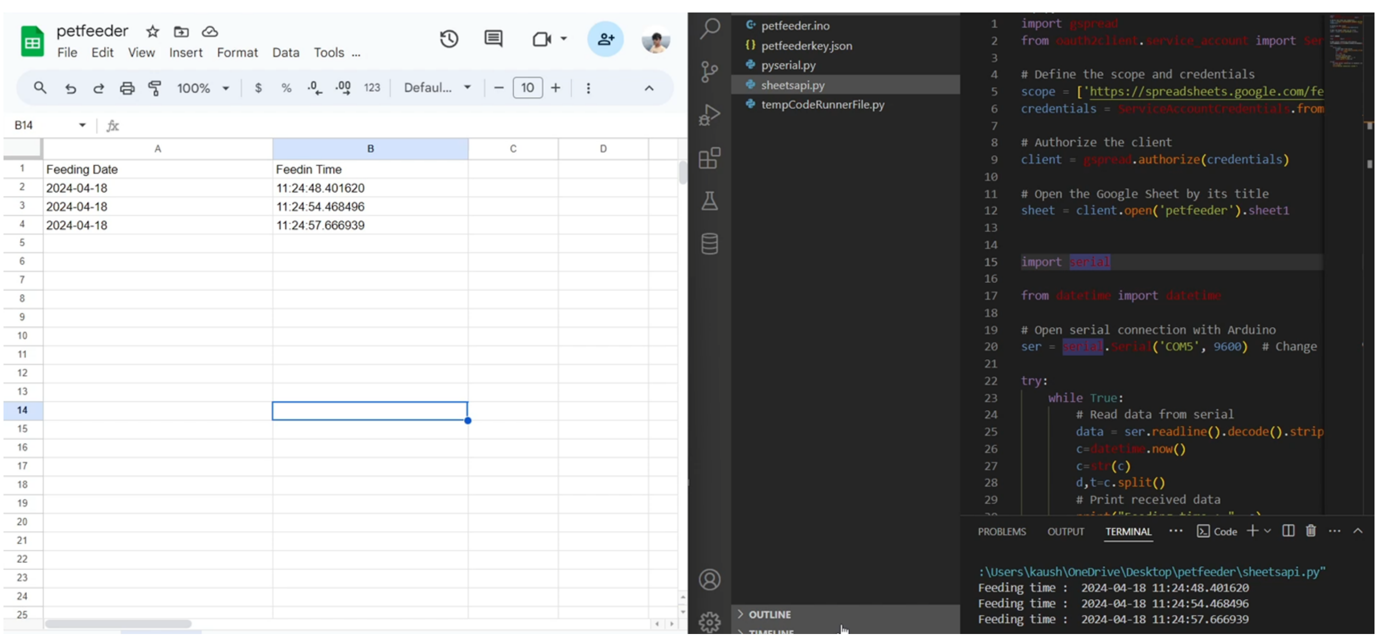



Fig. 3. Software Implementation

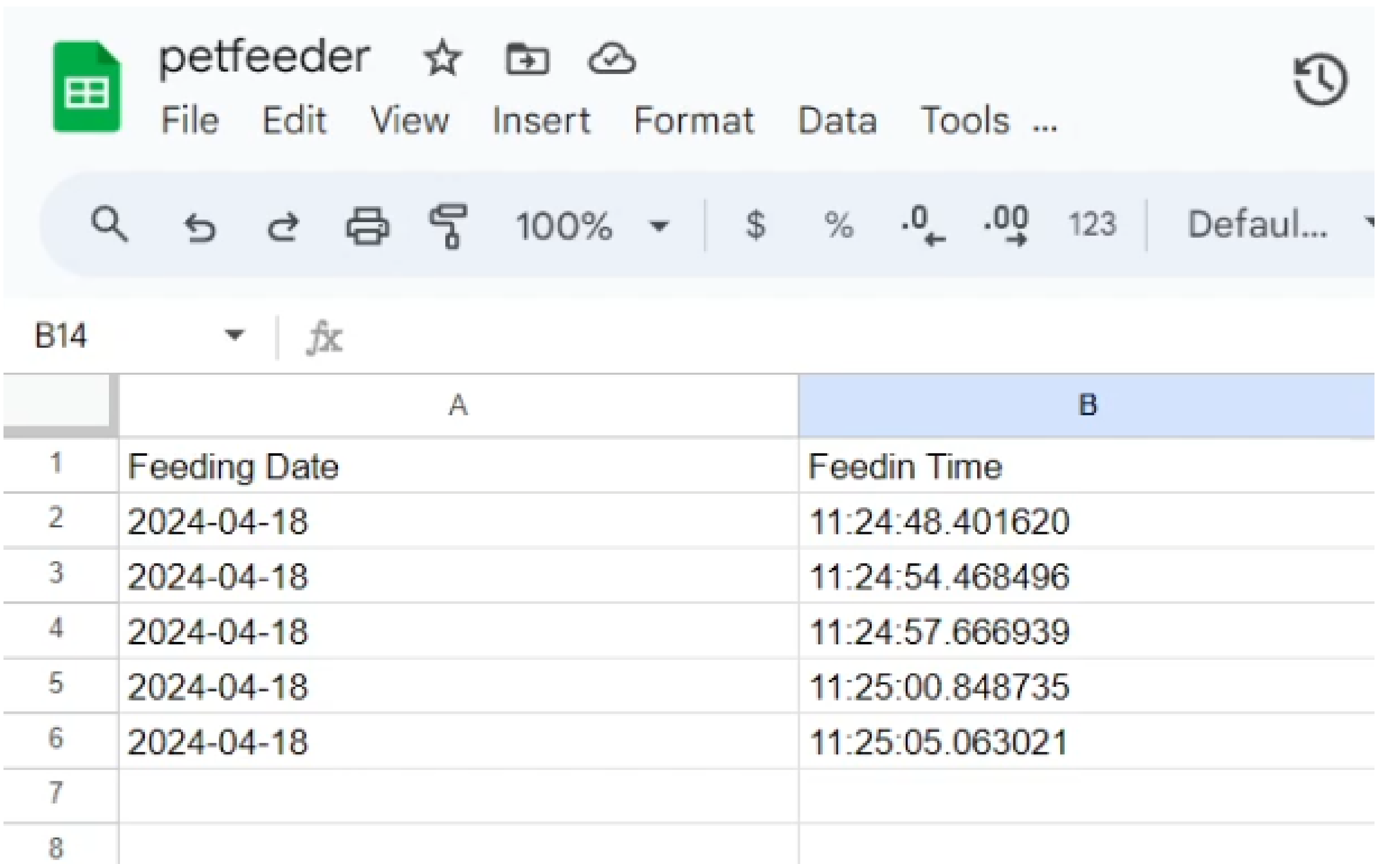
## IV. Results/Output

Project Outcomes: The Smart Pet Feeder project has yielded several significant outcomes, including successful implementation of pet identification, feeding automation, data collection, and analysis. Here are the key outcomes: Pet Identification and Feeding Automation:

The Smart Pet Feeder accurately identifies pets using RFID tags and dispenses food. Pet owners can customize feeding schedules based on their pets' dietary needs and feeding preferences, ensuring optimal nutrition and portion control.

Data Collection: The Smart Pet Feeder collects comprehensive data on pet feeding habits, including feeding times, portion sizes, and frequency of feeding. Feeding data is logged and stored securely in Google Cloud Storage, providing a centralized repository for pet-related information.

Data Analysis: Python scripts retrieve feeding data from Google Cloud Storage and visualize it using Google Sheets API. Pet owners can analyze feeding patterns, identify trends, and make informed decisions about their pets' diet and feeding schedules. Insights gained from data analysis help optimize feeding schedules, improve pet health, and enhance the overall pet care experience.




| | A | B |
|---|---|---|
| 1 | Feeding Date | Feedin Time |
| 2 | 2024-04-18 | 11:24:48.401620 |
| 3 | 2024-04-18 | 11:24:54.468496 |
| 4 | 2024-04-18 | 11:24:57.666939 |
| 5 | 2024-04-18 | 11:25:00.848735 |
| 6 | 2024-04-18 | 11:25:05.063021 |
| 7 | | |
| 8 | | |

Fig. 4. Data collection

Meaning of Results: Improved Pet Well-Being: The successful implementation of pet identification and feeding automation means that pets receive timely and consistent meals, leading to improved health and well-being. By customizing feeding schedules based on individual pet requirements, the Smart Pet Feeder promotes healthy eating habits and prevents overfeeding or underfeeding.

Enhanced User Experience: The Smart Pet Feeder provides pet owners with convenience and peace of mind by automating the feeding process and offering real-time monitoring and control capabilities. Data analysis features empower pet owners to make data-driven decisions about their pets' care, leading to a more fulfilling and rewarding pet ownership experience.

Future Improvements and Iterations: The results obtained from the Smart Pet Feeder project serve as a foundation for future improvements and iterations. By incorporating user feedback and refining the system based on data analysis insights, the Smart Pet Feeder can evolve to meet the evolving needs of pet owners and their furry companions. Overall, the outcomes of the Smart Pet Feeder project demonstrate the effectiveness of IoT technology in revolutionizing pet care and strengthening the bond between pets and their owners. By combining automation, data collection, and analysis, the Smart Pet Feeder offers a holistic solution for pet feeding management, promoting pet health, happiness, and well-being.

Fig. 5. Prototype of the proposed Smart Pet Feeder

## V. Conclusion and Future Directions

Key Findings of the Project:

Successful Implementation of Pet Feeding Automation: The Smart Pet Feeder effectively automates the pet feeding process using RFID technology and servo motor control, ensuring pets receive timely and consistent meals according to their feeding schedules.

Comprehensive Data Collection and Analysis: The Smart Pet Feeder collects and logs detailed data on pet feeding habits, including feeding times, portion sizes, and frequency of feeding. Python scripts retrieve this data from Google Cloud Storage and visualize it using Google Sheets API, enabling pet owners to gain insights into their pets' dietary patterns and make informed decisions about their care.

Enhanced User Experience: By providing real-time monitoring and control capabilities, customizable feeding schedules, and data analysis features, the Smart Pet Feeder enhances the overall pet care experience for pet owners. It offers convenience, peace of mind, and improved pet well-being, strengthening the bond between pets and their owners.

Significance of the Project: Promoting Pet Health and Well-Being: The Smart Pet Feeder addresses the critical need for reliable and efficient pet feeding solutions in today's fast-paced world. By ensuring pets receive optimal nutrition and portion control, the Smart Pet Feeder promotes pet health and well-being, reducing the risk of overfeeding, obesity, and related health issues.

Empowering Pet Owners: By providing pet owners with insights into their pets' feeding habits and enabling them to customize feeding schedules based on individual pet requirements, the Smart Pet Feeder empowers pet owners to take an active role in their pets' care. It fosters a sense of responsibility and engagement, strengthening the bond between pets and their owners.

Advancing IoT Technology in Pet Care: The Smart Pet Feeder showcases the potential of IoT technology in revolutionizing pet care. By integrating hardware components, software development, and cloud-based services, the project demonstrates how IoT can enhance pet feeding management, streamline pet care routines, and improve the overall pet ownership experience.

In summary, the key findings of the Smart Pet Feeder project highlight its significance in promoting pet health, empowering pet owners, and advancing IoT technology in pet care. By providing automation, data collection, and analysis capabilities, the Smart Pet Feeder offers a comprehensive solution for pet feeding management, ultimately enriching the lives of pets and their owners.

Future directions of the project:

Enhanced Feeding Customization: Introduce additional customization options for feeding schedules, such as meal frequency, portion sizes, and dietary preferences. Incorporate machine learning algorithms to adapt feeding schedules based on pet behaviour patterns and nutritional requirements.

Smartphone Integration: Develop dedicated mobile applications for iOS and Android platforms, allowing pet owners to remotely monitor and control the Smart Pet Feeder, receive feeding notifications, and access pet feeding data on-the-go.

Integration with Smart Home Systems: Expand compatibility with popular smart home systems such as Amazon Alexa and Google Assistant, enabling voice-controlled pet feeding and seamless integration with existing smart home environments.

Health Monitoring Features: Integrate health monitoring sensors such as weight sensors or activity trackers to monitor pet health indicators such as weight, activity levels, and food consumption. Provide insights into pet health trends and early detection of potential health issues.

Environmental Sustainability: Explore eco-friendly materials and energy-efficient designs for the Smart Pet Feeder to reduce environmental impact. Integrate features for monitoring and optimizing food consumption to minimize waste and promote sustainability.

By exploring these future directions, the Smart Pet Feeder project can continue to innovate and evolve, providing pet owners with advanced solutions for pet feeding management and enhancing the well-being of pets and their owners.

Fig. 6. Futuristic Smart Pet Feeder

## REFERENCES

[1] Dong, H.J., Abdulla, R., Selvaperumal, S.K., Duraikannan, S., Lakshmanan, R. and Abbas, M.K. , “Interactive on smart classroom system using beacon technology,”. International Journal of Electrical Computer Engineering, vol. 15(9), pp. 2088-8708, 2019.

[2] S.SUBAASHRI, ET AL., “Automatic Pet Monitoring and Feeding System Using IoT,”. International Journal of ChemTech Research, 10(14), pp. 253-258, 2017.

[3] JIGARMASEKAR, JASHSOHNI SHARMA, S.,“Review of IOT in Pet Management,”. IOSR Journal of Engineering, Volume 12, pp. 59-63, 2018.

[4] SINGH, P., SHARMA, A. K., SOOD, P. SINGH, P.,“REMOTE CONTROLLED AND GSM BASED AUTOMATED PET FEEDER,”.International Journal of Electrical Electronics Engineering, 2(2), pp. 14-18, 2015.

[5] TIWARI, M. S. ET AL., “Automatic Pet Feeder Using Arduino,”. International Journal of Innovative Research in Science, Engineering and Technology, 7(3), pp. 2891-2897, 2018.

[6] KARYONO, K. AND NUGROHO, I.H.T., “Smart dog feeder design using wireless communication, MQTT and Android client,”. In 2016 International Conference on Computer, Control, Informatics and its Applications (IC3INA), 3(2), pp. 191-196, 2016.

[7] WANKHEDE, D. K. PEDNEKAR, S., “Animal Tracking and Caring using RFID and IOT,”. IOSR Journal of Computer Engineering, pp. 24-27, 2017.

[8] Akash .D.Apturkar, Afnan. A .Maner, Vishal. P.Jadhav, Sharanabasappa. R. R., “Intelligent Pet Collar. Imperial Journal of Interdisciplinary Research (IJIR), 3(3), PP. 624-628, 2017,”.

[9] MANAKAPURE, P. S. SHAH, A., “Movement Monitoring of Pet Animal Using Internet of Things”.International Research Journal of Engineering and Technology, 5(5), pp. 3319-3325, 2018. A