NANDHA ENGINEERING COLLEGE

ERODE–638052 (Autonomous)

(Affiliated to Anna University, Chennai)

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

22AIC14 – INTERNET OF THINGS AND ITS APPLICATION

MINI PROJECT REPORT ON

TOPIC – SMART PET FEEDER MACHINE

Submitted by

REGISTER NUMBER	NAME
22AI014	FELLAH. A
22AI032	OVIYA SHRI. K
22AI035	PRADEEPA. P. S. R

NANDHA ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

This is to certify that the project work entitled "SMART PET FEEDER MACHINE" is the Bonafide work of FELLAH A(22AI014), OVIYA SHRI K(22AI032), PRADEEPA P S R (22AI035) who carried out the work under my supervision.

Signature of the Supervisor
Dr. K. Lalitha,
Professor,
Department of AI & DS,
Nandha Engineering College,
Erode – 638052.

Signature of the HOD
Dr. P. Karunakaran,
Head of the Department,
Department of AI & DS,
Nandha Engineering College,
Erode – 638052.

Submitted for End semester PBL review held on _____

SMART PET FEEDER MACHINE

AIM:

The aim of this project is to design and implement a smart pet feeder machine that automates the feeding process for pets, ensuring timely and accurate distribution of food. The system integrates IoT technologies for remote monitoring and control, making pet care more convenient for owners.

SCOPE:

This project addresses the common challenges faced by pet owners in maintaining consistent feeding schedules. The IoT-based smart feeder offers remote accessibility, precise portion control using a load cell, and automated operations powered by ESP8266. This solution is particularly suitable for working professionals or individuals frequently away from home, ensuring pets are well-fed at appropriate times.

BRIEF HISTORY:

Pet feeders have undergone significant advancements over the years. Early models were simple mechanical devices that relied on timers to dispense food at fixed intervals. While functional, these systems lacked precision, flexibility, and the ability to customize feeding schedules. As technology advanced, automated feeders with microcontrollers and basic sensors began to emerge, offering improved portion control and the ability to program feeding times. However, these solutions still required manual oversight and were not remotely accessible.

The introduction of IoT technology revolutionized pet care by enabling smart feeders with internet connectivity. This system allow pet owners to control feeding schedules, monitor food levels, and receive notifications through mobile applications. By integrating sensors like load cells for precise measurement and motorized dispensers for automation, IoT-based feeders have become indispensable tools for busy pet owners, ensuring their pets are well-fed and cared for even when they are away.

PROPOSED METHODOLOGY:

1. Hardware Setup:

- Attach the load cell to the food dispenser to measure food weight.
- Use the L298N driver to control the DC motor responsible for operating the dispensing mechanism.

- Connect the ESP8266 module to establish communication between the system and the IoT platform.

2. **Software Configuration:**
   - Program the ESP8266 to handle user commands and automate feeding schedules.
   - Calibrate the load cell using the HX711 amplifier for accurate measurements.
   - Use a web interface to allow users to set feeding schedules, monitor food levels, and receive notifications.
3. **Operation:**
   - The user sets the feeding schedule through the app.
   - At the scheduled time, the DC motor operates to dispense food.
   - The load cell measures the dispensed food and ensures the portion size matches user settings.
   - Notifications are sent to the user after each feeding event or when the food container is low.

COMPONENTS REQUIRED:

S.NO	HARDWARE	QUANTITY
1.	ESP8266 Module	1
2.	L298N Motor Driver	1
3.	DC Motor (PR500)	1
4.	Load Cell (20kg)	1
5.	Load Cell Amplifier (HX711)	1
6.	Power Supply	1
7.	Jumper Wires	As Required
8.	Food Container and Dispenser Mechanism	1

DESCRIPTION:

The IoT-based smart pet feeder machine uses an ESP8266 microcontroller to automate the feeding process. The load cell, along with its amplifier, measures the weight of food dispensed, ensuring accurate portion sizes. The L298N motor driver controls the DC motor to operate the food dispensing mechanism.

The system connects to the internet via ESP8266, enabling pet owners to monitor and control feeding schedules remotely through a web application or smartphone app. Users can set specific feeding times and portion sizes. The load cell ensures real-time monitoring of food levels in the dispenser, and notifications are sent to alert owners when food levels are low.

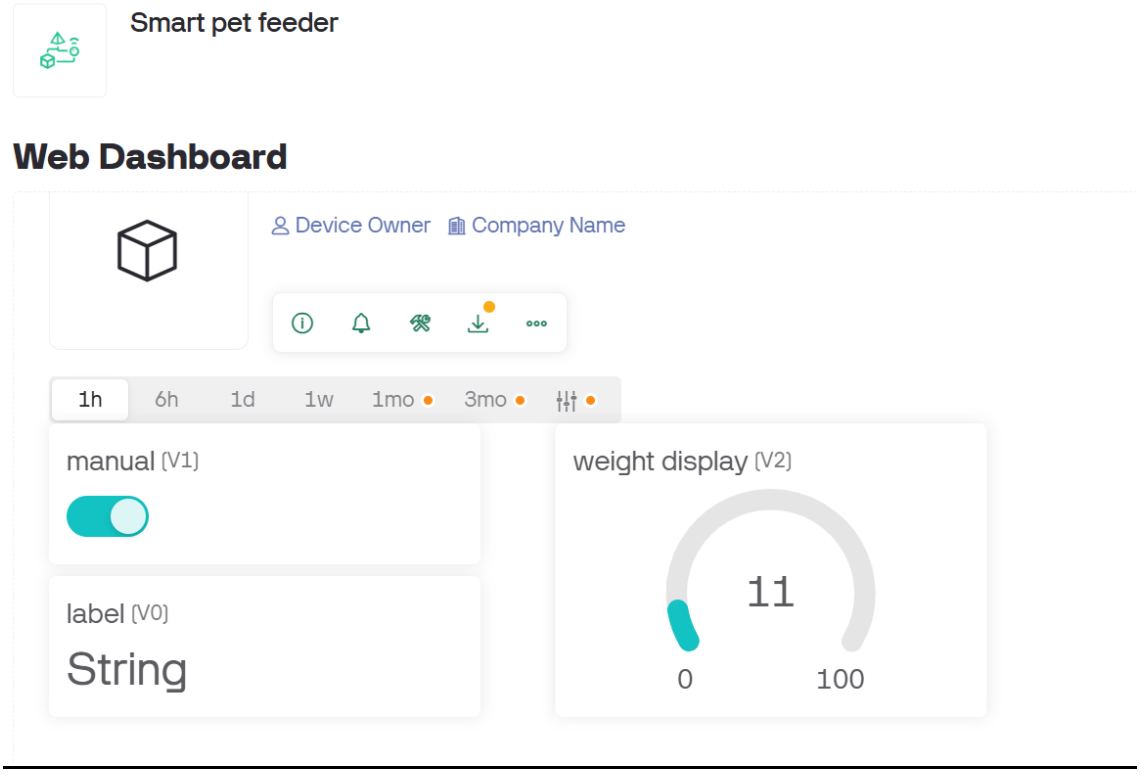
CODING:

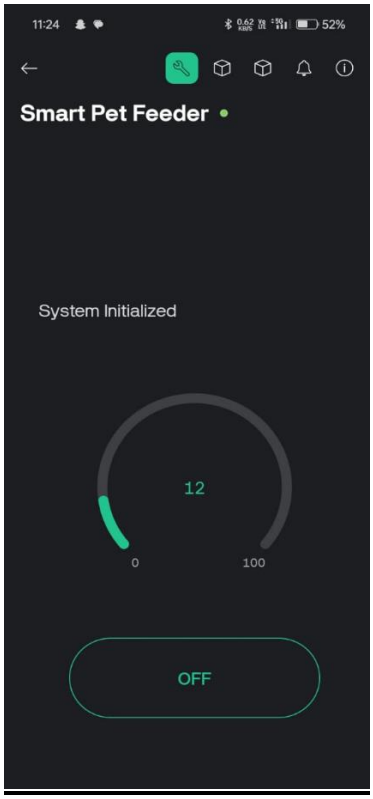
```
#define BLYNK_TEMPLATE_ID "TMPL3q11jvfWk"

#define BLYNK_TEMPLATE_NAME "Smart pet feeder"

#define BLYNK_AUTH_TOKEN "Eri9LmivSaQCTXk_zGW6XK6APoIL9mZq"

#include <HX711.h>

#include <BlynkSimpleEsp8266.h>

char ssid[] = "Raja";

char pass[] = "1234567890";

#define DOUT_PIN D2

#define SCK_PIN D1

#define MOTOR_PIN1 D3

#define MOTOR_PIN2 D4

#define MOTOR_ENA_PIN D5

HX711 scale;

const unsigned long feedInterval = 4 * 3600000;

const unsigned long motorRunTime = 5000;

unsigned long lastFeedTime = 0;

unsigned long motorStartTime = 0;

bool feedFlag = false;

void startFeeding();

void stopFeeding();

void timedFeeding();
```

```
void manualFeeding();
void updateWeightOnBlynk();
void setup() {
  Serial.begin(115200);
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
  pinMode(MOTOR_PIN1, OUTPUT);
  pinMode(MOTOR_PIN2, OUTPUT);
  pinMode(MOTOR_ENA_PIN, OUTPUT);
  scale.begin(DOUT_PIN, SCK_PIN);
  scale.set_scale(2280.f);
  scale.tare();
  Blynk.virtualWrite(V0, "System Initialized");
}
void loop() {
  Blynk.run();
  updateWeightOnBlynk();
  unsigned long currentTime = millis();
  if (currentTime - lastFeedTime >= feedInterval) {
    feedFlag = true;
    lastFeedTime = currentTime;
  }
  if (feedFlag) {
    timedFeeding();
    feedFlag = false;
  }
  if (motorStartTime > 0 && millis() - motorStartTime >= motorRunTime) {
    stopFeeding();
    motorStartTime = 0;
  }
}
```

```
void startFeeding() {
  digitalWrite(MOTOR_PIN1, HIGH);
  digitalWrite(MOTOR_PIN2, LOW);
  analogWrite(MOTOR_ENA_PIN, 255);
  motorStartTime = millis();
}
void stopFeeding() {
  digitalWrite(MOTOR_PIN1, LOW);
  digitalWrite(MOTOR_PIN2, LOW);
  analogWrite(MOTOR_ENA_PIN, 0);
}
void timedFeeding() {
  startFeeding();
}
void manualFeeding() {
  startFeeding();
}
void updateWeightOnBlynk() {
  float weight = scale.get_units(3);
  int weightInt = (int)(weight * 100);
  Blynk.virtualWrite(V2, weightInt);
}
BLYNK_WRITE(V1) {
  int buttonState = param.asInt();
  if (buttonState == 1) {
    manualFeeding();
```

OUTPUT SCREENSHOT:





PROTOCOLS:

1. **ESP8266 Wi-Fi Module:** This module enables communication with the IoT platform and internet connectivity for the device. It allows remote monitoring and control via a web application or smartphone app.
2. **Blynk IoT Protocol:** The system utilizes the Blynk platform, which connects the ESP8266 module to a user-friendly interface. This protocol manages device commands, data transfer, and notifications, ensuring seamless communication between the hardware and user interface.

LIMITATIONS:

- Internet Dependency: Requires a stable Wi-Fi connection for IoT features.
- Limited Portions: Restricted by the feeder's storage capacity for food.
- Hardware Durability: Mechanical components may wear out over time.
- Energy Consumption: Needs a continuous power supply for uninterrupted operation.
- User Configuration: Initial setup and calibration may require technical knowledge.
- Pet Behaviour Variance: Not ideal for pets that need supervision while eating or have unique dietary needs.
- Cost: The addition of IoT features may make it expensive compared to traditional feeders.

FUTURE ENHANCEMENT:

- Camera Integration: Add a live camera feed to monitor pets during feeding.
- Voice Commands: Enable voice control using assistants like Alexa or Google Assistant.

- Mobile App Enhancement: Include advanced features such as tracking feeding habits and health analytics.
- Battery Backup: Add a power backup to ensure operation during power outages.
- Multi-Pet Support: Design feeders to serve multiple pets with individual portions.
- AI-based Analytics: Use AI to predict pet dietary needs based on age, weight, and health conditions.
- Automatic Food Refilling: Add sensors to notify and automatically reorder food supplies when low.

CONCLUSION:

The IoT-based Smart Pet Feeder Machine demonstrates an innovative use of IoT and sensor technologies to address the challenges of pet care. Its ability to automate feeding schedules, ensure portion control, and allow remote monitoring makes it a practical solution for pet owners. Future improvements could include camera integration for visual monitoring, voice commands, and predictive feeding based on pet behaviour.