```
#define BLYNK_TEMPLATE_ID   "YOUR_TEMPLATE_ID"
#define BLYNK_TEMPLATE_NAME "YOUR_TEMPLATE_NAME"
#define BLYNK_AUTH_TOKEN    "YOUR_AUTH_TOKEN"

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <SPI.h>
#include <MFRC522.h>
#include <Servo.h>

// --- PIN CONFIGURATION ---
#define SS_PIN    D4
#define RST_PIN   D3
#define SERVO_PIN D2
#define TRIG_PIN  D5
#define ECHO_PIN  D6

#define MAX_PETS 10

MFRC522 rfid(SS_PIN, RST_PIN);
Servo feeder;
BlynkTimer timer;

// --- NETWORK CREDENTIALS ---
char ssid[] = "YOUR_WIFI_SSID";
char pass[] = "YOUR_WIFI_PASSWORD";

// --- SYSTEM VARIABLES ---
String petName[MAX_PETS];
String petUID[MAX_PETS];
int petCount = 0;
bool servoOpen = false;

// --- ULTRASONIC FOOD LEVEL FUNCTION ---
void checkFoodLevel() {
digitalWrite(TRIG_PIN, LOW);
delayMicroseconds(2);
digitalWrite(TRIG_PIN, HIGH);
delayMicroseconds(10);
digitalWrite(TRIG_PIN, LOW);

long duration = pulseIn(ECHO_PIN, HIGH);
int distance = duration * 0.034 / 2;

// Mapping: 20cm (Container Empty) to 2cm (Container Full)
int foodLevel = map(distance, 20, 2, 0, 100);
foodLevel = constrain(foodLevel, 0, 100);

// Send data to Blynk Gauge Widget on Virtual Pin V20
Blynk.virtualWrite(V20, foodLevel);

if (foodLevel < 20) {
  Blynk.logEvent("low_food", "Alert: Food level is critically low!");
```

```
}
}

// --- SERVO CONTROL ---
void closeServo() {
feeder.write(0);
servoOpen = false;
Blynk.virtualWrite(V7, 0); // Reset App Switch to OFF
Serial.println("Feeder Gate: CLOSED");
}

// --- RFID SCAN LOGIC ---
void checkRFID() {
if (!rfid.PICC_IsNewCardPresent() || !rfid.PICC_ReadCardSerial()) return;

String uid = "";
for (byte i = 0; i < rfid.uid.size; i++) {
  uid += String(rfid.uid.uidByte[i] < 0x10 ? "0" : "");
  uid += String(rfid.uid.uidByte[i], HEX);
}
uid.toUpperCase();

bool authorized = false;
for (int i = 0; i < petCount; i++) {
  if (uid == petUID[i]) {
    authorized = true;
    Serial.println("Access Granted: " + petName[i]);
    Blynk.logEvent("pet_detected", "Feeding session started for: " + petName[i]);

    feeder.write(180); // Open Gate
    servoOpen = true;
    timer.setTimeout(10000L, closeServo); // Auto-close after 10 seconds
    break;
  }
}

if (!authorized) {
  Blynk.logEvent("unknown_pet", "Unauthorized RFID Attempt! UID: " + uid);
  Serial.println("Access Denied: Unknown Tag.");
}

rfid.PICC_HaltA();
rfid.PCD_StopCrypto1();
}

// --- BLYNK MANUAL OVERRIDE (V7 Switch) ---
BLYNK_WRITE(V7) {
if (param.asInt()) {
  feeder.write(180);
  servoOpen = true;
  timer.setTimeout(10000L, closeServo);
  Serial.println("Manual Override: OPEN");
} else {
  closeServo();
```

```
}
}

// --- INITIALIZATION ---
void setup() {
Serial.begin(115200);
SPI.begin();
rfid.PCD_Init();
feeder.attach(SERVO_PIN);
feeder.write(0); // Ensure closed on boot

pinMode(TRIG_PIN, OUTPUT);
pinMode(ECHO_PIN, INPUT);

Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);

// Set non-blocking intervals
timer.setInterval(500L, checkRFID);      // Check RFID twice per second
timer.setInterval(5000L, checkFoodLevel); // Check Food Level every 5 seconds

Serial.println("System Online. Awaiting Pet Authentication...");
}

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