Category Garbage

```
#include <Servo.h>

Servo servo1; // Renamed for clarity (metal detection servo)
Servo servo2; // Renamed for clarity (tap servo)

const int sensorPin = 10; // Replace 10 with your metal sensor's digital pin
const int metalThreshold = 500; // Adjust based on your sensor

const int tapSensorPin = 4; // Replace 4 with your tap sensor's digital pin

void setup() {
  Serial.begin(9600);
  pinMode(sensorPin, INPUT);
  pinMode(tapSensorPin, INPUT);
  servo1.attach(9); // Attach metal detection servo to pin 9
  servo2.attach(5); // Attach tap servo to pin 5 (adjust if different)
}

void loop() {
  // Metal detection
  int sensorValue = digitalRead(sensorPin);
  if (sensorValue == HIGH && sensorValue > metalThreshold) {
    Serial.println("Metal detected!");
    servo1.write(90);
    delay(2000);
    servo1.write(0); // Optional: Reset servo to original position
  }

  // Tap detection (optional)
  int tapValue = digitalRead(tapSensorPin);
  if (tapValue == LOW) { // Adjust for tap sensor behavior (normally open/closed)
    servo2.write(90);
    delay(500); // Adjust tap duration as needed
    servo2.write(0); // Optional: Reset servo to original position
  }
}
```

Indicate Level and app message system

```
#include <LiquidCrystal.h> // Includes the LiquidCrystal Library
#include <WiFi.h>
#include <ESP32Firebase.h>

#define FIREBASE_HOST
"https://eco-bin-36651-default-rtdb.asia-southeast1.firebasedatabase.app/"
#define FIREBASE_AUTH "AIzaSyCit9uzNQ1sNzJirQhdXjWZQGObs6cfB8Y" // Replace with your
Firebase project's authentication secret
#define WIFI_SSID "" // Replace with your Wi-Fi network SSID
#define WIFI_PASSWORD "" // Replace with your Wi-Fi network password

// Firebase data objects
FirebaseData firebaseData;
FirebaseJson json;

// Sensor pins
const int sensorPin = A0; // Analog input pin for level sensor (replace if needed)
const int trigPin = 9;
const int echoPin = 10;

// Variables
long duration;
int distance;
String levelMessage = ""; // To store the level message
int height = 10; // Adjust based on your sensor's output range for full level

LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // Creates an LCD object (adjust pin connections if needed)

void setup() {
  lcd.begin(16, 2); // Initializes the interface to the LCD screen

  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(sensorPin, INPUT); // Set sensor pin as input

  Serial.begin(9600);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
```

```
  Serial.print("Connecting to WiFi... ");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println("Connected!");

  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH); // Initialize Firebase
  Serial.println("Firebase connected!");
}

void loop() {
  // Ultrasonic sensor measurement (optional, comment out if not used)
  /*
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.034 / 2;
  */

  // Level sensor reading
  int sensorValue = analogRead(sensorPin);
  if (sensorValue > height) {
    levelMessage = "Empty";
  } else if (sensorValue > (height / 2)) {
    levelMessage = "Use Me";
  } else {
    levelMessage = "Full";
  }

  // Display level on LCD
  lcd.clear(); // Clear LCD before displaying new content
  lcd.print(levelMessage);

  // Send level message to Firebase (replace "level" with your desired path)
```

```
  if (Firebase.setJSON(firebaseData, "/level", levelMessage)) {
    Serial.println("Level message sent to Firebase!");
  } else {
    Serial.println("Firebase error sending level message:");
    Serial.println(firebaseData.errorReason());
  }

  delay(2000); // Adjust delay as needed
}
```