Code for Automatic Hand Sanitizer Dispenser:

```
#include <HTTPClient.h>
#include <WiFi.h>
#include <ArduinoJson.h>
#include <LiquidCrystal.h>
const char* ssid = "Galaxy-M20";
const char* pass = "ac312124";
int count;
const int rs = 22, en = 4, d4 = 15, d5 = 13, d6 = 26, d7 = 21;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
const int trigPin = 5;
const int echoPin = 18;
const int pump = 19;
long duration;
int distance;
const char* url = "https://services1.arcgis.com/0MSEUqKaxRlEPj5g/arcgis/rest/
services/ncov_...
(Country_Region=%27India%27)&returnGeometry=false&outFields=Country_Regio
n,Confirmed,Recovered";
void setup() {
  Serial.begin(115200);
  delay(2000);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(pump, OUTPUT);
  digitalWrite(pump, LOW);
  lcd.begin(16, 2);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Covid19 Tracker");
  lcd.setCursor(0,1);
  lcd.print("Hand Sanitizer");
  Serial.println("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");              // print ... till not connected
  }
  Serial.println("WiFi connected");
}
void ultra(){
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.0340 / 2;
  Serial.println("Distance");
  Serial.println(distance);
  if (distance <= 15){
    Serial.print("Opening Pump");
    digitalWrite(pump, HIGH);
    delay(2000);
    digitalWrite(pump, LOW);
    ESP.restart();
    }
}
void loop() {
  ultra();
  HTTPClient https;
  String data;
  https.begin(url);
  int httpCode = https.GET();
  if (httpCode > 0) { //Check for the returning code
    String payload = https.getString();
    char charBuf[500];
    payload.toCharArray(charBuf, 500);
    //Serial.println(payload);
    const size_t capacity = JSON_ARRAY_SIZE(1) + JSON_ARRAY_SIZE(4) +
JSON_OBJECT_SIZE(1) + 2 * JSON_OBJECT_SIZE(2) + JSON_OBJECT_SIZE(4) +
3 * JSON_OBJECT_SIZE(6) + 2 * JSON_OBJECT_SIZE(7) + 690;
    DynamicJsonDocument doc(capacity);
    deserializeJson(doc, payload);
    JsonArray fields = doc["fields"];
    JsonObject features_0_attributes = doc["features"][0]["attributes"];
    long features_0_attributes_Last_Update = features_0_attributes["Last_Update"];
    int features_0_attributes_Confirmed = features_0_attributes["Confirmed"];
    //int features_0_attributes_Deaths = features_0_attributes["Deaths"];
    int features_0_attributes_Recovered = features_0_attributes["Recovered"];
    if (count < 3){
    //Serial.println(features_0_attributes_Confirmed);
    lcd.setCursor(0, 0);
    lcd.print("IN Confirmed:");
    lcd.print(features_0_attributes_Confirmed);
    //Serial.println(features_0_attributes_Recovered);
    lcd.setCursor(0, 1);
    lcd.print("IN Recovered:");
    lcd.print(features_0_attributes_Recovered);
    }
    if (count > 3){
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Wash Hands");
    lcd.setCursor(0, 1);
    lcd.print("Avoid Contacts");
    }
    if (count > 6){
    count = 0;
    }
}
  else {
    Serial.println("Error on HTTP request");
  }
  https.end();
  count++;
}
```