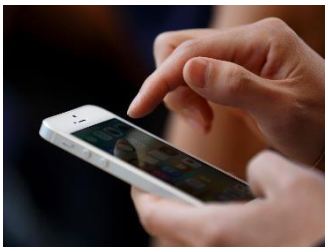
PROJENİN TANITIMI

Projemizin asıl amacı kısaca uzaktan pencereyi açıp kapatmayı sağlayabilmektir. Ancak şu anda kullanılan pencere modelleri uzaktan kontrol edilmeye pek elverişli değildir. Bundan dolayı projemizin yaptıkları şunlardır:

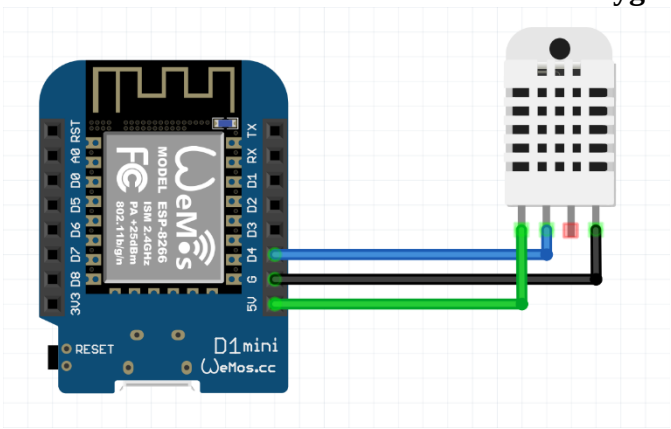
1. Wemos D1 ve DHT 11 sensörü kullanarak odanın sıcaklığını ve nemini ölçüyor.
2. MQTT ile Adafruit bağlantısı kurularak sıcaklık ve nem değerlerini grafik olarak gösteriyor.
3. Firebase Realtime Database ile anlık sıcaklık ve nem değerleri kaydediliyor.
4. MIT App Inventor ile yapılan mobil uygulama sayesinde de anlık sıcaklık ve nem değerleri ölçülebiliyor.
5. Ek olarak mobil uygulama içinde sanal olarak düşünülen pencere açıp kapatma metodu eklendi. Bu sayede uzaktan kontrol etmeyi simüle edebildik.
6. Uygulamadan açık kapatılan pencere durumu da Firebase üzerindeki veritabanına kaydedildi.



Wemos D1 **Adafruit ve Firebase** **Mobil Uygulama**

Kullanılan Ekipmanlar ve Teknolojiler:

- Wemos D1
- DHT11 Sensör
- Jumper Kablo
- Firebase
- Adafruit
- MIT APP INVENTOR 2



Wemos D1: Arduino UNO dizilimine sahip ESP8266EX tabanlı mikrokontrolcü kartıdır.

DHT11 Sensör: DHT11 Isı ve Nem Sensör Kartı, üzerinde DHT11 sensörü bulunan, bağlantıları çekilip breadboard veya farklı kullanımlar için kolaylaştıtılmış hale sokulmuş modüldür.

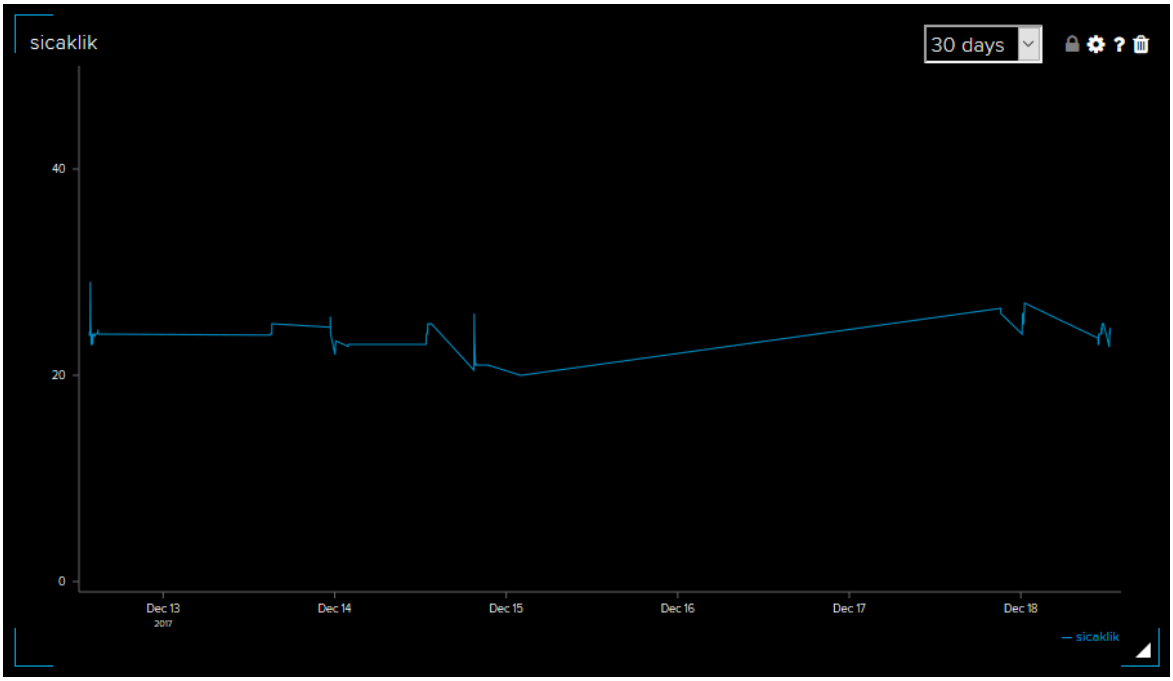
DHT11 sıcaklık ve nem algılayıcı kalibre edilmiş dijital sinyal çıkışı veren gelişmiş bir algılayıcı birimidir. Yüksek güvenilirliktedir ve uzun dönem çalışmalarda dengelidir. 8 bit mikroişlemci içerir, hızlı ve kaliteli tepki verir. 0 ile 50°C arasında 2°C hata payı ile sıcaklık ölçen birim, 20-90% RH arasında 5% RH hata payı ile nem ölçer.
Pin kullanımı: “-“ pin GND, “s” pini dijital sinyal çıkış ve ortadaki pin ise 5V gerilim pinidir.

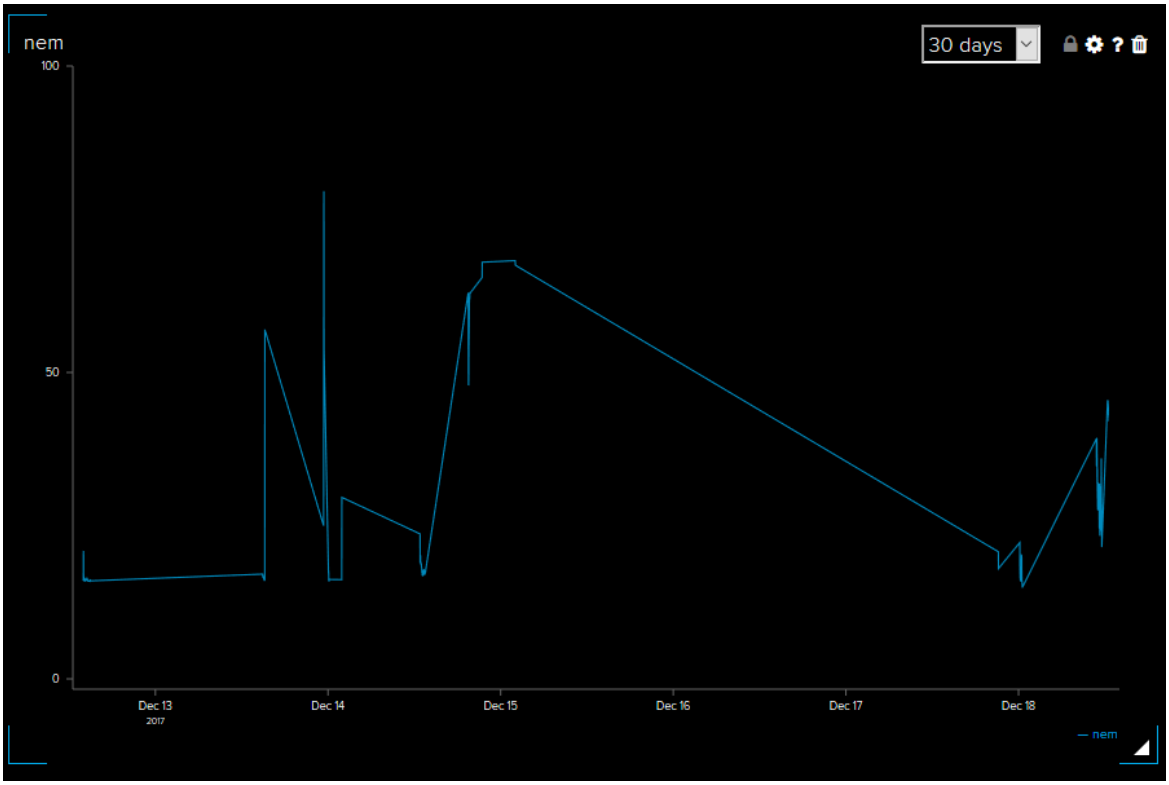
FIREBASE VERİLERİ

https://pencere-84b78.firebaseio.com/

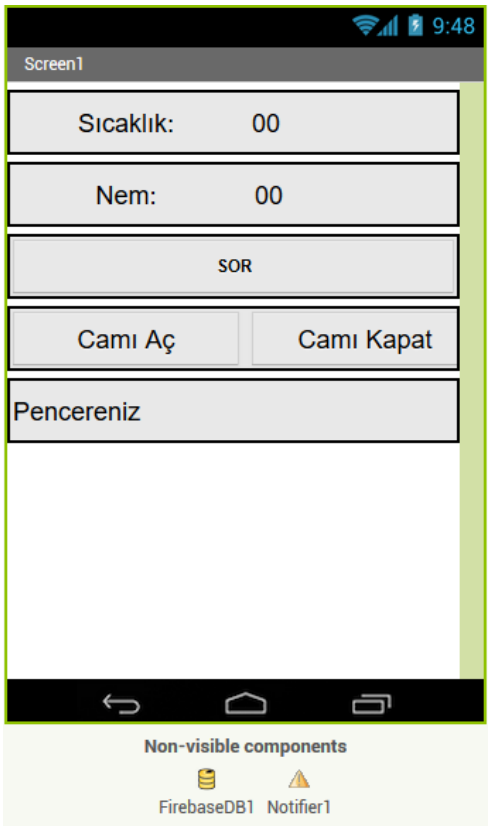

```
pencere-84b78
    nem: "42.00"
    pencere: "\"kapali\""
    sicaklik: "24.00"
```

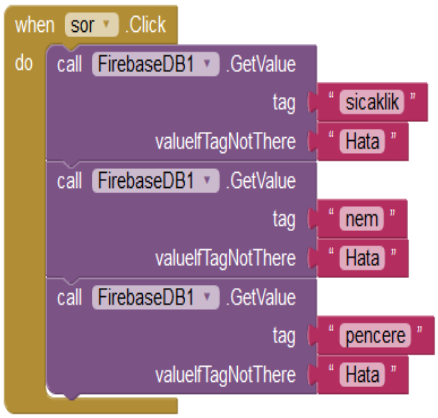
ADAFRUIT VERİLERİ

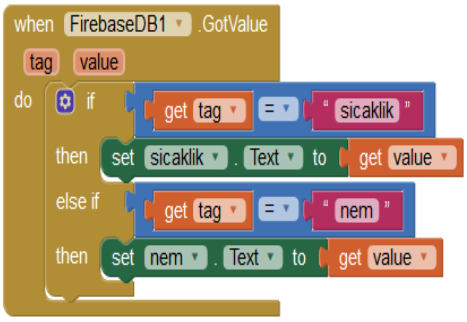


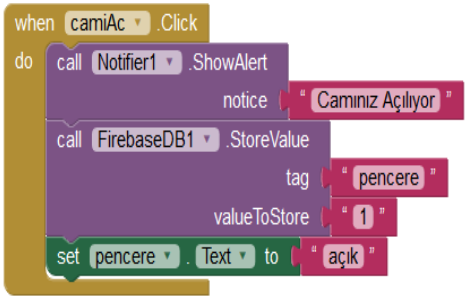


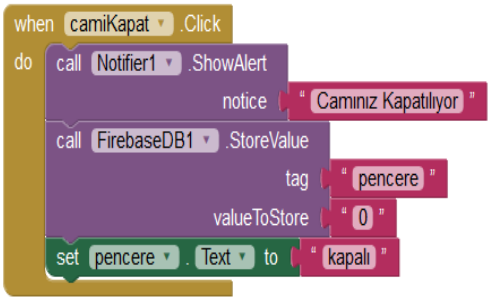
MIT APP INVENTOR 2







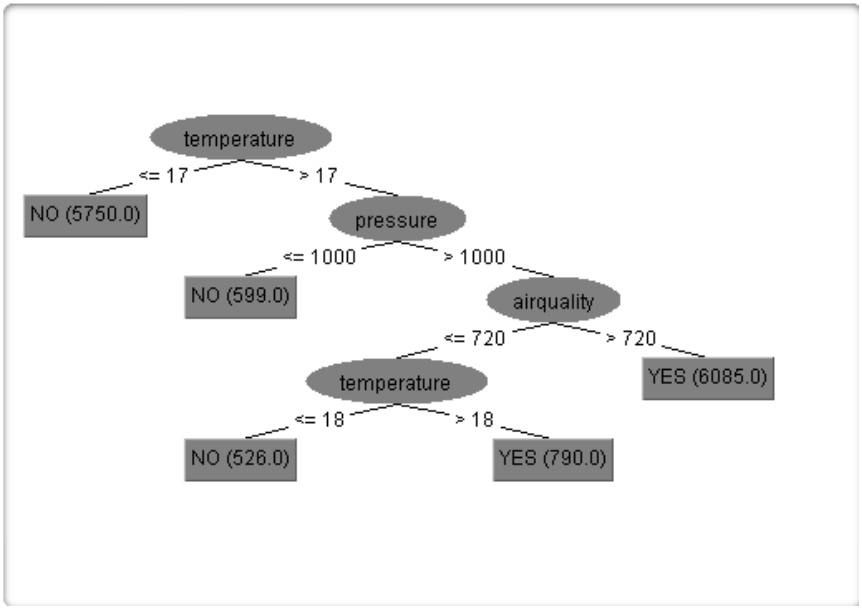




WEKA VERİLERİ

```
=== Classifier model (full training set) ===

J48 pruned tree
------------------

temperature <= 17: NO (5750.0)
temperature > 17
|   pressure <= 1000: NO (599.0)
|   pressure > 1000
|   |   airquality <= 720
|   |   |   temperature <= 18: NO (526.0)
|   |   |   temperature > 18: YES (790.0)
|   |   airquality > 720: YES (6085.0)

Number of Leaves  :     5

Size of the tree :      9


Time taken to build model: 0.18 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       13750               100      %
Kappa statistic                          1
Mean absolute error                      0
Root mean squared error                  0
Relative absolute error                  0      %
Root relative squared error              0      %
Total Number of Instances            13750

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000     YES
                 1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000     NO
Weighted Avg.    1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000

=== Confusion Matrix ===

    a    b   <-- classified as
 6875    0 |    a = YES
    0 6875 |    b = NO
```

Tree View



ARDUINO KODLARI

```
#include <ESP8266WiFi.h>
#include <Adafruit_MQTT.h>
#include <Adafruit_MQTT_Client.h>
#include <dht11.h>
//#include <Firebase.h>
#include <FirebaseArduino.h>
//#include <FirebaseCloudMessaging.h>
//#include <FirebaseError.h>
//#include <FirebaseHttpClient.h>
//#include <FirebaseObject.h>


/************************* WiFi Access Point *********************************/
#define WLAN_SSID       "selam"
#define WLAN_PASS       "hopasinanay"
/************************* Adafruit.io Setup *********************************/
#define AIO_SERVER      "io.adafruit.com"
#define AIO_SERVERPORT  1883                   // use 8883 for SSL
#define AIO_USERNAME    "AndySelva"
#define AIO_KEY         "45b47e4d297d4837bee45f9e96790395" //Adafruit AIO Key

#define FIREBASE_HOST "pencere-84b78.firebaseio.com"
#define FIREBASE_AUTH "rjnYg5vDRGNstiCiCyS5UfBZ0zl3aYNw9OSPfdO0"

#define DHT11PIN 4

dht11 DHT11;

// Create an ESP8266 WiFiClient class to connect to the MQTT server.
WiFiClient client;
Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT, AIO_USERNAME, AIO_KEY);


/****************************** Feeds ***************************************/
// Setup a feed called 'pencere' for publishing.
// Notice MQTT paths for AIO follow the form: <username>/feeds/<feedname>
Adafruit_MQTT_Publish feedTemp = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/proje.sicaklik");
Adafruit_MQTT_Publish feedHumi = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/proje.nem");
// Setup a feed called 'onoff' for subscribing to changes.
Adafruit_MQTT_Subscribe onoffbutton = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME "/feeds/onoff");
/*************************** Sketch Code ************************************/
```

```
void MQTT_connect();
void setup()
{


    Serial.begin(115200);
    delay(10);

    Serial.println(F("Akıllı Pencere"));
    // Connect to WiFi access point.
    Serial.println(); Serial.println();
    Serial.print("Connecting to ");
    Serial.println(WLAN_SSID);
    WiFi.begin(WLAN_SSID, WLAN_PASS);
    while (WiFi.status() != WL_CONNECTED)
    {
        delay(500);
        Serial.print(".");
    }
    Serial.println();
    Serial.println("WiFi connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
    // Setup MQTT subscription for onoff feed.
    mqtt.subscribe(&onoffbutton);

    Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
    Firebase.setString("sicaklik", "15");
    Firebase.setString("nem", "10");
}
uint32_t x = 0;
```

```
void loop()
{

    float nem;
    float sicaklik;
    String sSicaklik;
    String sNem;
   // String pencere;
   // int cam;

    // Ensure the connection to the MQTT server is alive (this will make the first
    // connection and automatically reconnect when disconnected).  See the MQTT_connect
    // function definition further below.
    MQTT_connect();
    sicakliknem();
    // this is our 'wait for incoming subscription packets' busy subloop
    // try to spend your time here
    Adafruit_MQTT_Subscribe* subscription;
    while ((subscription = mqtt.readSubscription(5000)))
    {
        if (subscription == &onoffbutton)
        {
            Serial.print(F("Got: "));
            Serial.println((char*)onoffbutton.lastread);
        }
    }
    // Now we can publish stuff!
    Serial.print(F("\Deger gönderiliyor "));
    Serial.println((float)DHT11.temperature);
    Serial.println((float)DHT11.humidity);
    sicaklik = DHT11.temperature;
    nem = DHT11.humidity;

    sSicaklik = (String)sicaklik;
    sNem = (String)nem;

    Firebase.setString("sicaklik", sSicaklik);
    Firebase.setString("nem", sNem);
    if (!feedTemp.publish((float)DHT11.temperature))
    {
        Serial.println(F("Failed"));
    }
    else
    {
        Serial.println(F("OK!"));
    }

    if (!feedHumi.publish((float)DHT11.humidity))
    {
        Serial.println(F("Failed"));
    }
    else
    {
        Serial.println(F("OK!"));
    }

}
```

```
void MQTT_connect()
{
    int8_t ret;
    // Stop if already connected.
    if (mqtt.connected())
    {
        return;
    }
    Serial.print("MQTT ye bağlanıyor... ");
    uint8_t retries = 3;
    while ((ret = mqtt.connect()) != 0)
    {
// connect will return 0 for connected
        Serial.println(mqtt.connectErrorString(ret));
        Serial.println("Retrying MQTT connection in 5 seconds...");
        mqtt.disconnect();
        delay(5000);
// wait 5 seconds
        retries--;
        if (retries == 0)
        {
            // basically die and wait for WDT to reset me
            while (1) ;
        }
    }
    Serial.println("MQTT ye bağlandı");
}

void sicakliknem(){

  Serial.println("\n");

  int chk = DHT11.read(DHT11PIN);

}
```