

**DÜZCE ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**  
**ELEKTRİK-ELEKTRONİK VE BİLGİSAYAR MÜHENDİSLİĞİ**  
**İLERİ VERİTABANI YÖNETİM SİSTEMLERİ**

**MONGODB İLE IOT PROJESİ**

**MUSTAFA ÖZER**  
[mustafa206955@ogr.duzce.edu.tr](mailto:mustafa206955@ogr.duzce.edu.tr)

**Projenin Özeti**

*Bu proje kapsamında Raspberry Pi ile sensörlerden toplanan verilerin Python'la MongoDB'ye yazılıp, görselleştirilmesi uygulaması gerçekleştirilmiştir. Bu proje ilkokullarda bulunan süt odalarının sıcaklık ve nem takibini otomatik hale getirmek amacıyla düşünülmüştür.*

*Gün geçtikçe artan IoT cihazlarının okullarımızda da yer alması önem arz etmektedir. IoT cihazlarından gelen sensör verilerinin sürekli, hacimli ve çeşitli olması nedeniyle NoSql veritabanlarının kullanılması kaçınılmazdır. Bu nedenle projede ilgili birimdeki sıcaklık ve nem değerlerinin ölçülmesi, ölçüm verilerinin saklanması ve grafiklerle görselleştirilmesi için Raspberry Pi 3B+ kart bilgisayar, DHT11 sıcaklık ve nem sensörü, ilişkisel olmayan MongoDB veritabanı ve Python 2 programlama dili ile ThingSpeak kullanılmıştır.*

**1.GİRİŞ**

‘MongoDB ile IoT Projesi’ ağırlıklı olarak masaüstü uygulama tabanlı bir projedir. Raspberry Pi kart bilgisayarı üzerinde Raspian işletim sistemi kullanılmıştır. Raspian işletim sistemi üzerine MongoDB 2.4.10 kurulumu gerçekleştirilmiştir. Python ile MongoDB işlemlerinin gerçekleştirilebilmesi için

pymongo modülü yüklenmiştir. Masaüstü tasarımlar Tkinter modülü ile hazırlanmıştır.

DHT11 sıcaklık ve nem sensöründen gelen veriler yerelde çalışan mongoDB sunucusundaki ‘iotDB’ isimli veritabanının ‘dht11’ koleksiyonuna (tablo gibi düşünülebilir) eklenmektedir. Ayrıca kullanıcı adları ve parolaları da masaüstü uygulama üzerinden ‘users’ koleksiyonuna kayıt edilmektedir.

Sensörden gelen verilerin görselleştirilmesi için ThingSpeak altyapısı kullanılmıştır. Ayrıca son beş sıcaklık değeri python programlama dili ve Tkinter ile grafik şeklinde gösterilmiştir. Sıcaklık ve nem değerleri RaspController mobil uygulamasından da takip edilebilmektedir.

**2.UYGULAMANIN GELİŞTİRİLMESİ**

Projenin özeti ve Giriş kısmında donanımsal ve yazılımsal bileşenler hakkında yüzeysel bilgiler verilmiştir. Bu bölümde kaynak kod ve masaüstü uygulama hakkında detaylı bilgiler verilecektir.

**a) Veritabanı Hazırlama İşlemleri**

Veritabanı ve koleksiyonların hazırlanması için ilk önce bir kereye mahsus *veritabanikurma.py* kaynak kodu çalıştırılmalıdır. Bunun sonucunda mongoDB sunucusunda iotDB veritabanı ve users, dht11 koleksiyonları (tablolar) Görsel 1 de gösterildiği şekilde oluşturulmuştur.

```
pi@raspberrypi ~  
File Edit Tabs Help  
Tue Apr 27 10:17:05.579 [initandlisten]  
> help  
db.help() help on db methods  
db.mycoll.help() help on collection methods  
sh.help() sharding helpers  
rs.help() replica set helpers  
help admin administrative help  
help connect connecting to a db help  
help keys key shortcuts  
help misc misc things to know  
help mr mapreduce  
  
show dbs show database names  
show collections show collections in current database  
show users show users in current database  
show profile show most recent system.profile entries wit  
h time >= 1ms  
show logs show the accessible logger names  
show log [name] prints out the last segment of log in memor  
y. 'global' is default  
use <db_name> set current database  
db.foo.find() list objects in collection foo  
db.foo.find( { a : 1 } ) list objects in foo where a == 1  
it result of the last line evaluated; use to f  
urther iterate  
DBQuery.shellBatchSize = x set default number of items to display on s  
hell  
exit quit the mongo shell  
> show dbs  
denemeDB 0.0625GB  
iotDB 0.0625GB  
local 0.03125GB  
mydatabase 0.0625GB  
myvt 0.0625GB  
> use iotDB  
switched to db iotDB  
> db.users.find()  
{ "_id" : ObjectId("608759ae74fece06bad94107"), "password" : "1234", "user" : "admin" }  
{ "_id" : ObjectId("6087adab74fece178c13f430"), "password" : "12345", "user" : "ali" }  
{ "_id" : ObjectId("6087ae1274fece17baedea52"), "password" : "1234", "user" : "mehmet" }
```

Görsel 1 Veritabanı ve koleksiyonların kontrolü

## b) Kullanıcı İşlemleri

Uygulamanın başlaması ve kullanıcı işlemlerinin gerçekleştirilebilmesi için *giris.py* kaynak kodu çalıştırılmalıdır. Diğer tüm masaüstü arayüzler ilgili koşullara göre doğrudan, kaynak kod çalıştırmadan açılacaktır.

Kullanıcı girişi arayüzünde takip sistemine giriş, kullanıcı kayıt, kullanıcı silme ve parola güncelleme yapılabilmektedir (Görsel 2).



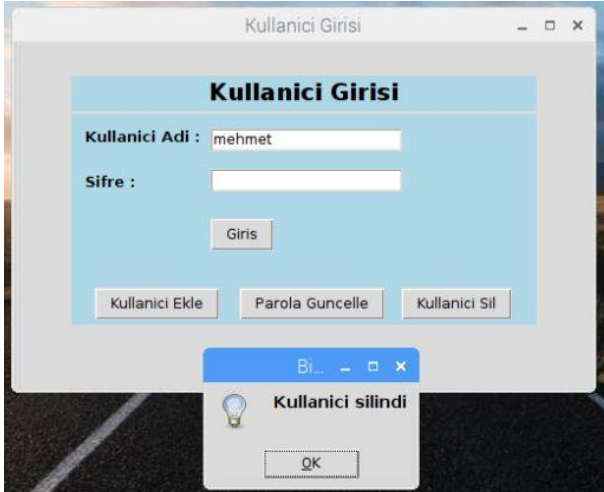
Görsel 2 Kullanıcı Ekleme

Kayıtlı olan kullanıcı adı tekrar kayıt edilmeye çalışıldığında uyarı verip, kayıt işlemini gerçekleştirmeyecektir (Görsel 3).



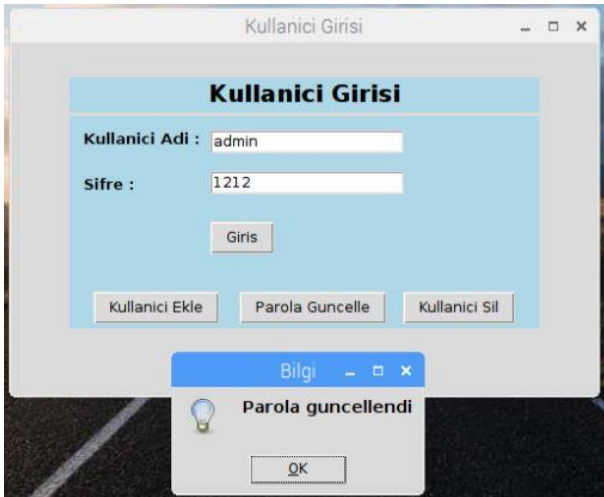
Görsel 3 Kullanıcı Ekleme Uyarısı

Kullanıcı silme işlemi kullanıcı adına göre gerçekleşmektedir (Görsel 4).



Görsel 4 Kullanıcı Silme

Kullanıcı adına göre parola güncelleme işlemi gerçekleşmektedir (Görsel 5).



Görsel 5 Parola Güncelleme

Veritabanının 'users' koleksiyonunda bulunan herhangi bir kullanıcı adı ve parolası ile takip sistemine giriş yapılabilir (Görsel 6).



Görsel 6 Kullanıcı Girişi

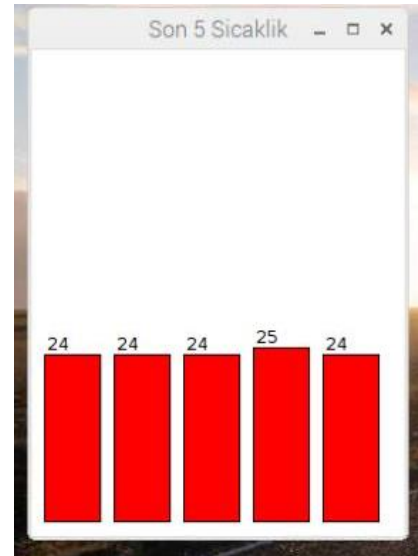
### c) DHT11 Sensör İşlemleri

Kullanıcı girişi doğrulandığında takip sistemi açılmaktadır. Takip sisteminde ortalama 2-3 saniyede bir sensörden alınan sıcaklık ve nem değerleri zaman damgası ile gösterilmektedir. Gösterilen sıcaklık ve nem sensör değerleri zaman damgası ile IoTDB veritabanı dht11 koleksiyonuna kayıt edilmektedir (Görsel 7). Bu sensör verileri ölçüm anında ThingSpeak web sayfasına gönderilerek anlık grafik elde edilmesi de sağlanmaktadır (Görsel 10).



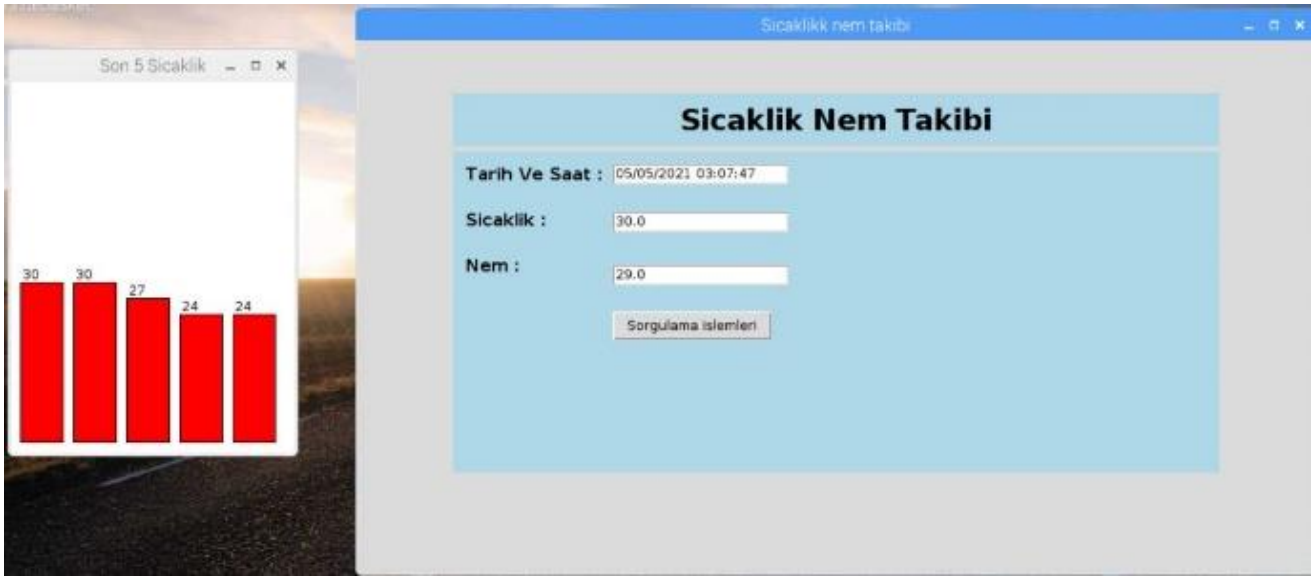
Görsel 7 Sıcaklık ve Nem Takibi

Ayrıca son 5 sıcaklık değeri sürekli olarak grafik şeklinde de gösterilmektedir (Görsel 8).



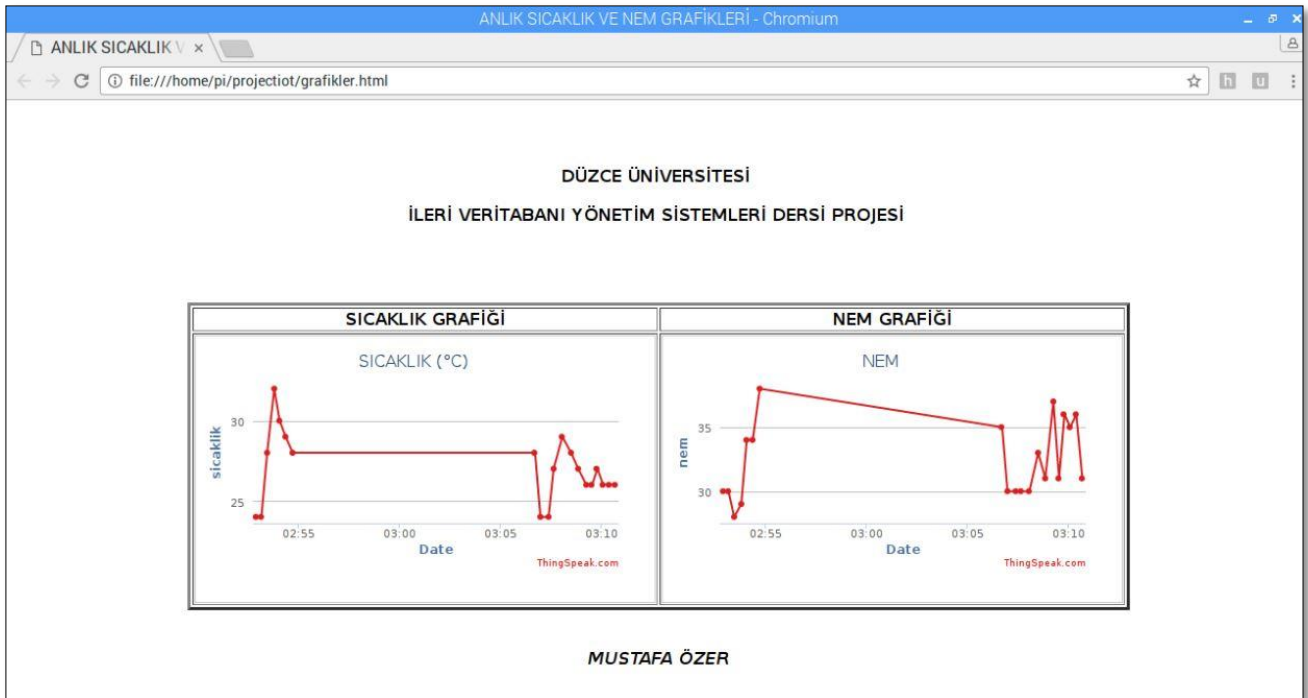
Görsel 8 Sıcaklık Grafiği

Sıcaklık Nem Takibi arayüzü ile birlikte açılan ‘Son 5 Sıcaklık’ grafiği, iotDB veritabanı dht11 koleksiyonu (tablosu) içerisinde son beş kaydın sıcaklık değerleri çekilerek oluşturulmuştur (Görsel 9).



Görsel 9 Takip Sistemi

Sıcaklık Nem Takibi arayüzü ile birlikte *grafikler.html* web sayfası da açılmaktadır. Bu web sayfasında ThingSpeak sıcaklık ve nem grafikleri anlık olarak güncellenmektedir (Görsel 10).



Görsel 10 Anlık Sıcaklık ve Nem Grafikleri

Sıcaklık Nem Takibi arayüzünde ‘Sorgulama işlemleri’ butonuna tıklandığında çeşitli sorgulamaların yapılabileceği arayüz açılacaktır (Görsel 11, Görsel 12 ve Görsel 13).

The screenshot shows a window titled 'Sorgulama işlemleri'. It has a search bar with 'Tarih : 27/04/2021' and a 'Goster' button. Below it are two rows of filters for 'Sıcaklık' and 'Nem', each with a 'Buyuk Olani Goster' and 'Kucuk Olani Goster' button. A 'Tumunu Listele' button is also present. The main area displays a list of sensor data for the date 27/04/2021.

Tarih	Sıcaklık	Nem
27/04/2021 11:57:55	28.0C	31.0%
27/04/2021 11:57:59	28.0C	31.0%
27/04/2021 11:58:12	30.0C	30.0%
27/04/2021 11:58:14	32.0C	30.0%
27/04/2021 11:58:22	30.0C	33.0%
27/04/2021 11:58:24	30.0C	36.0%
27/04/2021 11:58:29	30.0C	34.0%
27/04/2021 11:58:31	30.0C	37.0%
27/04/2021 12:15:18	28.0C	32.0%
27/04/2021 12:20:58	28.0C	33.0%

Görsel 11 27/04/2021 Tarihli Sensör Verilerini Listeleme

The screenshot shows the 'Sorgulama işlemleri' window with the 'Sıcaklık' filter set to 30. The 'Tarih' field is empty. The 'Nem' filter is also empty. The 'Tumunu Listele' button is present. The main area displays a list of sensor data for the temperature value 30.

Tarih	Sıcaklık	Nem
27/04/2021 11:58:14	32.0C	30.0%
05/05/2021 02:53:46	32.0C	29.0%
05/05/2021 02:53:50	32.0C	33.0%

Görsel 12 Sıcaklık Değeri 30'dan Büyük Olan Kayıtlar

The screenshot shows the 'Sorgulama işlemleri' window with the 'Nem' filter set to 30. The 'Tarih' field is empty. The 'Sıcaklık' filter is also empty. The 'Tumunu Listele' button is present. The main area displays a list of sensor data for the humidity value 30.

Tarih	Sıcaklık	Nem
27/04/2021 11:57:55	28.0C	31.0%
27/04/2021 11:57:59	28.0C	31.0%
27/04/2021 11:58:22	30.0C	33.0%
27/04/2021 11:58:24	30.0C	36.0%
27/04/2021 11:58:29	30.0C	34.0%
27/04/2021 11:58:31	30.0C	37.0%
27/04/2021 12:15:18	28.0C	32.0%
27/04/2021 12:20:58	28.0C	33.0%
27/04/2021 12:23:08	27.0C	34.0%
04/05/2021 21:57:36	25.0C	31.0%

Görsel 13 Nem Değeri 30'dan Büyük Olan Kayıtlar



Sıcaklık ve nem deęerleri RaspController mobil uygulamasından ilgili pin (GPIO14) ve sensör türü (DHT11) seçilerek takip edilmektedir (Görsel 14).



Görsel 14 Mobil Uygulamadan Sıcaklık ve Nem Takibi

### 3.KARŞILAŞILAN PROBLEMLER

Raspian işletim sisteminin MongoDB'nin son sürümünü desteklememesi, MongoDB'nin kullanılan eski (2.4.10) sürümünün Python 3 pymongo modülü ile kullanılamaması gibi problemlerle karşılaşıldı. Bu problemlerden dolayı verilerin grafiklerle görselleştirilmesi kısmında matplotlib modülü kullanılamamıştır. Ayrıca Raspian işletim sistemi üzerinde Python 3 için pymongo modülünün yüklenememesi nedeniyle AWS, Azure gibi sistemlerde MongoDB bağlantısı kurulamamıştır.

DHT11 sıcaklık ve nem sensörünün kararlı ölçüm yapabilen kütüphanesi Python 2 ile çalışmamıştır. Daha eski bir kütüphane kurularak çözüm bulunmuştur.

Sıcaklık grafiğinde son 5 deęerin sürekli gösterilmesinde problem yaşanmış sonrasında yazılımsal yöntemle giderilmiştir.

### 4.SONUÇLAR

Bu proje ile ortamdaki sıcaklık ve nem verilerini depolayıp grafikler ile görselleştirme imkanı elde edilmiştir. Bu projede dersin amaçlarından biri olan NoSql veritabanı kullanımı gerçekleşmiştir. Bu sistem ile uzak bir noktadan sensör verileri takip edilebilmekte olup zaman ve maddiyattan tasarruf yapılabilmektedir. IoT projelerinde NoSql veritabanlarının kullanılmasının daha uygun olduğu öğrenilmiştir.

### 5.KAYNAKLAR

<https://pimylifeup.com/mongodb-raspberry-pi/>

<https://www.w3schools.com/>

<https://medium.com/>