



# Akıllı Ofis Sistemi

*"Akıllı Teknoloji, Güvenli Ofis"*

2025

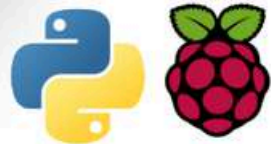


Hazırlayanlar

2022123019 - Ahmet Mert Kabak  
2022123067 - Sıla Erdoğan

Danışman Hoca

Ahmet Gürkan YÜKSEK



# İÇİNDEKİLER BÖLÜMÜ

## i. ÖNSÖZ

### 01 GİRİŞ

- 1.1. Projenin Amacı ve Kapsamı ..... 3
- 1.2. Problem Tanımı ve Çözüm Önerisi ..... 4
- 1.3. Projenin Hedefleri ..... 5
- 1.4. Sistemin Genel Yapısı ..... 6

### 02 SİSTEM GELİŞTİRME VE UYGULAMA

- 2.1. Donanım Bileşenleri ..... 7
  - 2.1.1. Raspberry Pi Kurulumu ..... 8
  - 2.1.2. Gpio Yapılandırması ..... 9
  - 2.1.3. IoT Sensörlerin Bağlantıları ..... 10
  - 2.1.4. LCD Ekran Kurulumu ..... 11
  - 2.1.5. Devre Şemaları ve Bağlantılar .... 12
- 2.2. Yazılım Geliştirme .... 13
  - 2.2.1. AWS İot Core Ve MQTT İşlemleri .... 14
  - 2.2.2. Veritabanı Mimarisi-AWS... 16
  - 2.2.3. Veritabanı İşlemleri .... 17
  - 2.2.4. Web Arayüzü .... 18

### 03 TEST VE SONUÇLAR

- 3.1. Sistem Testleri ve Sonuçları .... 20
- 3.2. Maliyet Analizi .... 21
- 3.3. Karşılaşılan Sorunlar Ve Çözümleri.... 22

### 04 SONUÇ VE KAYNAKÇA

- 4.1 Sonuç Ve Kaynakça 23
- 4.2 Ofisimizin Taslağı ve Teşekkürler 24

# ÖNSÖZ

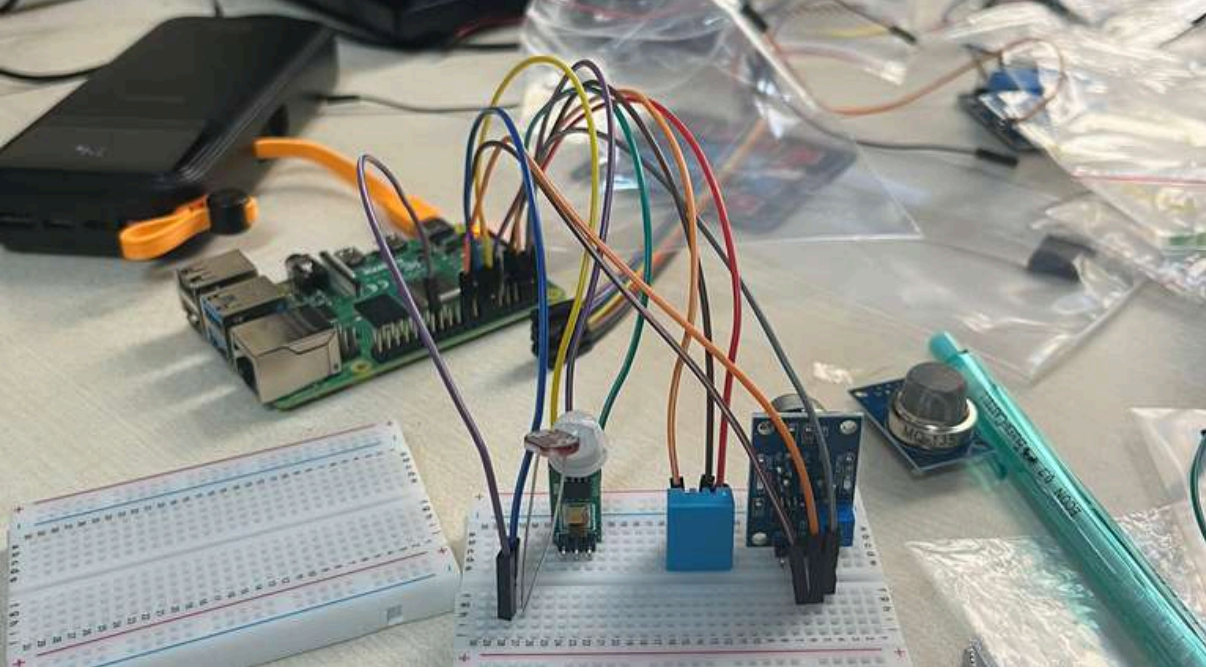
Bu projede, modern ofislerin güvenlik ve otomasyon ihtiyaçlarını karşılamak amacıyla yapay zeka destekli bir akıllı ofis sistemi geliştirilmiştir. Raspberry Pi tabanlı sistemimiz, yüz tanıma teknolojisi ve IoT sensörler kullanarak ofis ortamının güvenliğini ve kontrolünü sağlamaktadır.

Projemiz, yazılım ve donanım entegrasyonunu başarıyla gerçekleştirerek, ofis ortamlarında kullanılacak pratik bir çözüm sunmaktadır. Web tabanlı arayüzü sayesinde uzaktan erişim ve kontrol imkanı sağlayan sistemimiz, modern ofislerin dijital dönüşümüne katkıda bulunmayı hedeflemektedir.

Bu raporda, projenin geliştirilme sürecinden test aşamalarına kadar tüm detaylar adım adım anlatılmıştır. Amacımız, benzer projeler geliştirmek isteyen araştırmacılara ve öğrencilere yol gösterici bir kaynak oluşturmaktır.



# GİRİŞ



Ofis ortamlarında güvenlik sadece fiziksel giriş-çıkış kontrolünden ibaret değildir. Aynı zamanda ortam koşullarının takibi, enerji verimliliği, çalışan konforu ve acil durum yönetimi gibi birçok faktörü de içermektedir. Bu projenin temel amacı, ofis ortamlarında güvenliği ve kontrolü sağlayacak entegre bir sistem geliştirmektir. Projemiz yapay zeka destekli yüz tanıma teknolojisi ile giriş-çıkış kontrolü, IoT sensörler ile ortam takibi, web tabanlı izleme ve kontrol sistemi, veritabanı yönetimi ve gerçek zamanlı bildirim sistemi gibi temel bileşenleri içermektedir.

## PROJENİN AMACI, TANIMI VE HEDEFLERİ



Bu projenin temel amacı, ofis ortamlarında güvenliğini ve kontrolünü sağlayacak entegre bir sistem geliştirmektir. Projemiz şu temel bileşenleri içermektedir:

- **Yüz tanıma teknolojisi ile giriş-çıkış kontrolü**
- **IoT sensörler ile ortam takibi**
- **Web tabanlı izleme ve kontrol sistemi**
- **Veritabanı yönetimi**
- **Gerçek zamanlı bildirim sistemi**

# PROBLEM TANIMI VE ÇÖZÜM ÖNERİSİ

Modern ofis ortamlarında karşılaşılan temel problemler şu şekilde sıralanabilir:



## No 01 - Güvenlik Sorunları

- Yetkisiz kişilerin ofise ve özel alanlara erişimi
- Giriş-çıkışların düzenli takip edilememesi
- Mesai saatlerine uyumun kontrol edilememesi
- Önemli alanların (müdür odası gibi) güvenliğinin sağlanamaması



## No 02 - Ortam Koşulları

- Sıcaklık ve nem değerlerinin optimal seviyelerde tutulamaması
- Hava kalitesinin düzenli ölçülememesi
- Gaz kaçağı gibi tehlikeli durumların geç fark edilmesi



## No 03 - Verimlilik Sorunları

- Toplantı odası ve tuvalet gibi ortak alanların kullanım durumunun bilinmemesi
- Gereksiz bekleme süreleri
- Ofis kaynaklarının verimsiz kullanımı

Çözüm Önerimiz:

Geliştirdiğimiz "Akıllı Ofis Güvenlik ve İzleme Sistemi", bu problemlere yönelik entegre bir çözüm sunmaktadır. Sistem, Raspberry Pi tabanlı merkezi bir kontrol ünitesi etrafında tasarlanmıştır. Çeşitli sensörler ve aktuatörler kullanılarak ofis ortamının tam kontrolü sağlanmaktadır.

# PROJENİN HEDEFLERİ

## Ana Hedefler:

Ofis güvenliğini en üst seviyeye çıkarmak için kapsamlı bir kontrol sistemi oluşturmak. Bu sistem, personel giriş-çıkışlarından özel alan erişimlerine kadar tüm güvenlik ihtiyaçlarını karşılamalıdır.

Çalışanlar için ideal çalışma ortamı sağlamak amacıyla ortam koşullarını sürekli izlemek ve optimize etmek. Sıcaklık, nem ve hava kalitesi gibi parametrelerin ideal seviyelerde tutulması hedeflenmektedir.

Ofis kaynaklarının verimli kullanımını sağlamak için akıllı izleme ve yönetim sistemleri geliştirmek. Özellikle ortak alanların kullanımında verimliliği artırmak amaçlanmaktadır.

Teknik Hedefler	Kullanıcı Deneyimi	İş Hedefleri
<ul style="list-style-type: none"><li>Raspberry Pi tabanlı güvenilir ve stabil çalışan bir sistem geliştirmek</li></ul>	<ul style="list-style-type: none"><li>Kolay kullanılabilir web arayüzü tasarımı</li></ul>	<ul style="list-style-type: none"><li>Ofis güvenlik maliyetlerinde %30 azalma</li></ul>
<ul style="list-style-type: none"><li>Sensör verilerinin %99.9 doğrulukla toplanması ve işlenmesi</li></ul>	<ul style="list-style-type: none"><li>Anlık bildirim sistemi</li></ul>	<ul style="list-style-type: none"><li>Çalışan verimliliğinde %15 artış</li></ul>
<ul style="list-style-type: none"><li>7/24 kesintisiz çalışma kapasitesi</li></ul>	<ul style="list-style-type: none"><li>Detaylı raporlama özellikleri</li></ul>	<ul style="list-style-type: none"><li>Sistem kurulum maliyetinin 1 yıl içinde geri dönüşü</li></ul>

# Sistemin Genel Yapısı

Akıllı Ofis Güvenlik ve İzleme Sistemi, üç ana katmandan oluşmaktadır:

## Donanım Katmanı

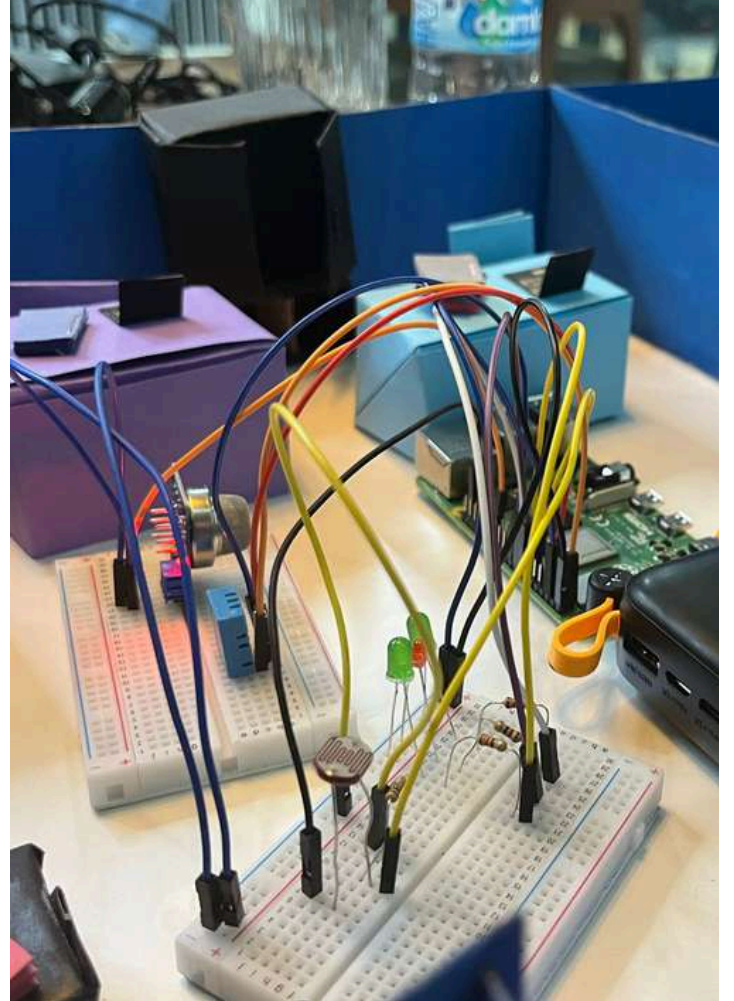
- Merkezi İşlem Birimi: Raspberry Pi 4 Model B
- Sensör Ağı:
  - DHT11 Sıcaklık/Nem Sensörü
  - MQ2 Gaz Sensörü
  - PIR Hareket Sensörleri
  - LDR Işık Sensörü
  - MQ135 Hava Kalitesi Sensörü
- Kontrol Birimleri:
  - LCD Ekranlar
  - LED'ler
  - Servo Motor
  - Buzzer

## Yazılım Katmanı

- Python tabanlı ana kontrol yazılımı
- OpenCV ile yüz tanıma sistemi
- MySQL veritabanı
- Flask web framework'ü
- AWS IoT Core entegrasyonu

## Bulut Katmanı

- Amazon RDS veritabanı servisi
- AWS IoT Core servisi
- Web tabanlı yönetim paneli



**%90**  
**ORAN**

Sistem bileşenleri birbirleriyle sürekli iletişim halinde çalışmaktadır. Sensörlerden toplanan veriler Raspberry Pi üzerinde işlendikten sonra hem yerel veritabanında saklanmakta hem de bulut sistemine aktarılmaktadır. Başarı oranımızı saklamıyoruz :)



# DONANIM BİLEŞENLERİ

Projemizde kullanılan temel donanım bileşenleri ve özellikleri:

## Merkezi İşlem Birimi:

Raspberry Pi 4 Model B

- 4GB RAM
- 64-bit quad-core işlemci
- Dual-band WiFi
- Bluetooth 5.0
- 32GB SD Kart



## Sensörler:

### DHT11 Sıcaklık/Nem Sensörü

- Sıcaklık ölçüm aralığı: 0-50°C
- Nem ölçüm aralığı: 20-90%

### MQ2 -MQ135 Gaz/Kalite Sensörü

- LPG, propan, hidrojen ,alkol gazı tespiti
- Analog ve dijital çıkış
- Yüksek hassasiyet

### PIR Hareket Sensörü (HC-SR501)

- Algılama mesafesi: 7 metre
- Algılama açısı: 120°
- Ayarlanabilir hassasiyet

### LDR Işık Sensörü



## Kontrol ve Görüntüleme Birimleri:

- 16x2 LCD Ekran
- RGB LED'ler
- SG90 Servo Motor
- Aktif Buzzer
- 4x4 Tuş Takımı



## Güç Kaynağı:

- Güç kaynağı için 5V'luk kullanabilirsiniz
- Biz Raspberry Pi'mizin girişi type-c olduğu için
- Kendi şarj aletimizle çalıştırdık.

# RASPBERRY Pİ KURULUMU

Raspberry Pi'nin kurulum ve yapılandırma adımları:

## İşletim Sistemi Kurulumu:

- 1.Raspberry Pi OS (64-bit) tercih edildi
- 2.Raspberry Pi Imager kullanılarak SD karta yazıldı
- 3.İlk açılışta temel yapılandırmalar tamamlandı:
  - o Klavye düzeni
  - o Zaman dilimi
  - o WiFi bağlantısı
  - o SSH erişimi



**GEREKLİ**  
**KÜTÜPHANELER**

```
sudo apt-get update  
sudo apt-get upgrade  
sudo apt-get install python3-pip  
sudo apt-get install python3-opencv  
sudo apt-get install mysql-server  
Ekleyeceğiniz sensörlere göre  
gpiopin,adafruit-dns,os,time gibi  
kütüphaneleri de kurabilirsiniz.
```

# GPIO Yapılandırması



3v3 Power	1		2	5v Power
GPIO 2 (I2C1 SDA)	3		4	5v Power
GPIO 3 (I2C1 SCL)	5		6	Ground
GPIO 4 (GPCLK0)	7		8	GPIO 14 (UART TX)
Ground	9		10	GPIO 15 (UART RX)
GPIO 17	11		12	GPIO 18 (PCM CLK)
GPIO 27	13		14	Ground
GPIO 22	15		16	GPIO 23
3v3 Power	17		18	GPIO 24
GPIO 10 (SPI0 MOSI)	19		20	Ground
GPIO 9 (SPI0 MISO)	21		22	GPIO 25
GPIO 11 (SPI0 SCLK)	23		24	GPIO 8 (SPI0 CE0)
Ground	25		26	GPIO 7 (SPI0 CE1)
GPIO 0 (EEPROM SDA)	27		28	GPIO 1 (EEPROM SCL)
GPIO 5	29		30	Ground
GPIO 6	31		32	GPIO 12 (PWM0)
GPIO 13 (PWM1)	33		34	Ground
GPIO 19 (PCM FS)	35		36	GPIO 16
GPIO 26	37		38	GPIO 20 (PCM DIN)
Ground	39		40	GPIO 21 (PCM DOUT)

## Pin Bağlantıları:

- DHT11: GPIO2
- MQ2: GPIO18
- PIR Sensörler: GPIO23, GPIO24
- LCD: I2C Pinleri (SDA, SCL)
- LED'ler: GPIO5, GPIO6, GPIO13
- Servo: GPIO12

## Güvenlik Önlemleri:

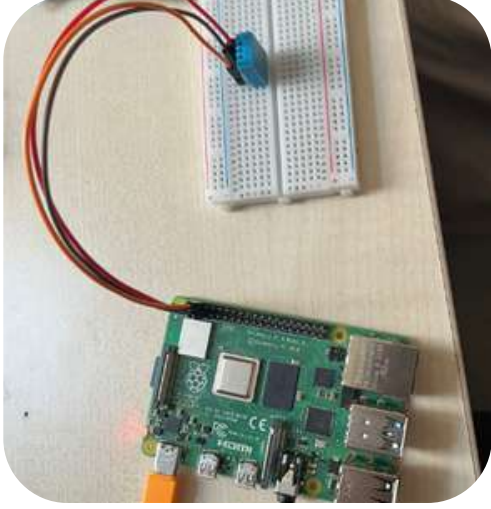
- Firewall yapılandırması
- SSH güvenlik ayarları
- Düzenli yedekleme sistemi

Biz pin bağlantılarını böyle yaptık,siz taktığınız pine göre gpio pinini bularak ona göre kodları düzenleyebilirsiniz.

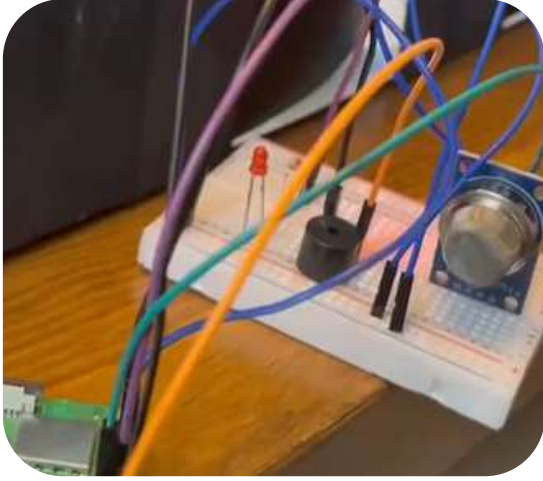


# IoT Sensörlerin Bağlantıları

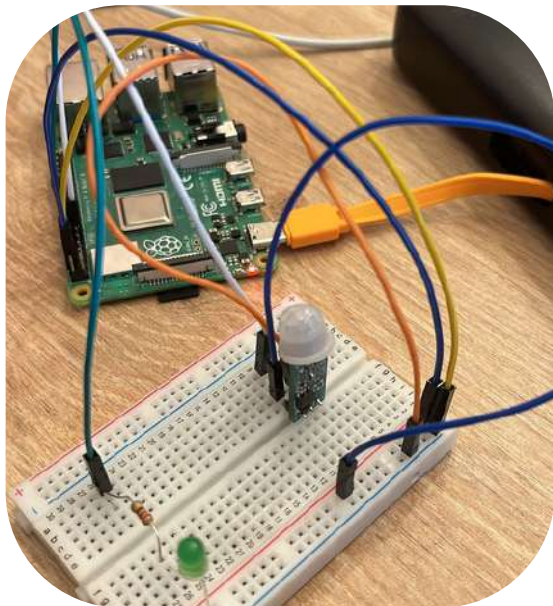
Akıllı ofis sistemimizin sinir sistemini oluşturan sensör ağı, ofis ortamını sürekli olarak izleyerek güvenli ve konforlu bir çalışma ortamı sağlamaktadır. Her bir sensör, belirli bir görevi yerine getirmek üzere stratejik noktalara yerleştirilmiş ve Raspberry Pi ile entegre edilmiştir.



**Sıcaklık ve nem kontrolü için** kullandığımız **DHT11** sensörü, ofis ortamının iklim koşullarını sürekli olarak takip etmektedir. GPIO2 pinine bağlı olan bu sensör, her iki saniyede bir ölçüm yaparak verileri merkezi sisteme iletmektedir. Elde edilen veriler, belirlediğimiz konfor aralığı olan 18-26°C sıcaklık ve %30-60 nem seviyelerine göre değerlendirilmekte, bu değerlerin dışına çıkıldığında sistem otomatik olarak uyarı vermektedir.



Ofis güvenliğinin kritik bir bileşeni olan **MQ2 gaz sensörü**, mutfak alanına yerleştirilmiştir. GPIO18 üzerinden analog veri okuyan bu sensör, olası gaz kaçaklarını anında tespit edebilmektedir. Sensörün doğru çalışması için gerekli olan 20 saniyelik ısınma süresi, sistem başlangıcında otomatik olarak beklenmektedir. 300ppm üzerinde gaz algılandığında, sistem acil durum protokolünü devreye sokmaktadır.



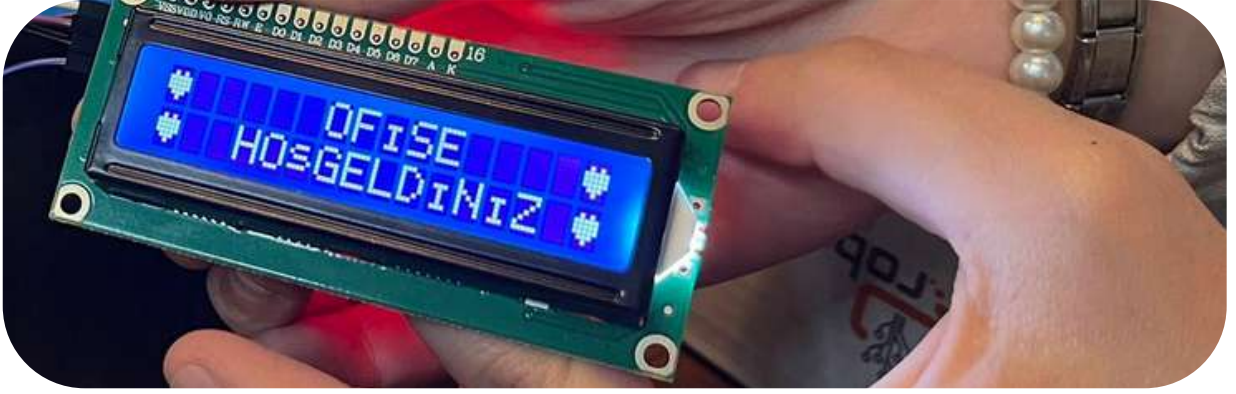
Toplantı odası ve tuvalet gibi ortak alanların kullanım durumunu takip etmek için **PIR hareket sensörleri** kullanılmıştır. GPIO23 ve GPIO24 pinlerine bağlı olan bu sensörler, 0.5 saniyelik tepki süresiyle alanların doluluk durumunu hassas bir şekilde tespit edebilmektedir. Yanlış algılamaları önlemek için sensörlerin konumları ve algılama açıları optimize edilmiştir.



# LCD Ekran Kurulumu

LCD ekran, sistemimizin kullanıcı arayüzünün önemli bir parçasını oluşturmaktadır. Ana giriş kapısına yerleştirdiğimiz 16x2 karakterlik LCD ekran, çalışanlara anlık bilgi ve geri bildirim sağlamaktadır. Ekranın I2C arayüzü üzerinden Raspberry Pi'ye bağlantısı, kablo karmaşasını önlerken aynı zamanda kurulum sürecini de basitleştirmektedir.

Kurulum sürecinde öncelikle I2C haberleşme protokolünü Raspberry Pi üzerinde etkinleştirdik. Ardından gerekli Python kütüphanelerini yükleyerek LCD ekranın adresini tespit ettik. Sistemimizde kullandığımız PCF8574 I2C dönüştürücü modülü sayesinde, ekran kontrolü için sadece iki pin (SDA ve SCL) kullanmamız yeterli oldu.



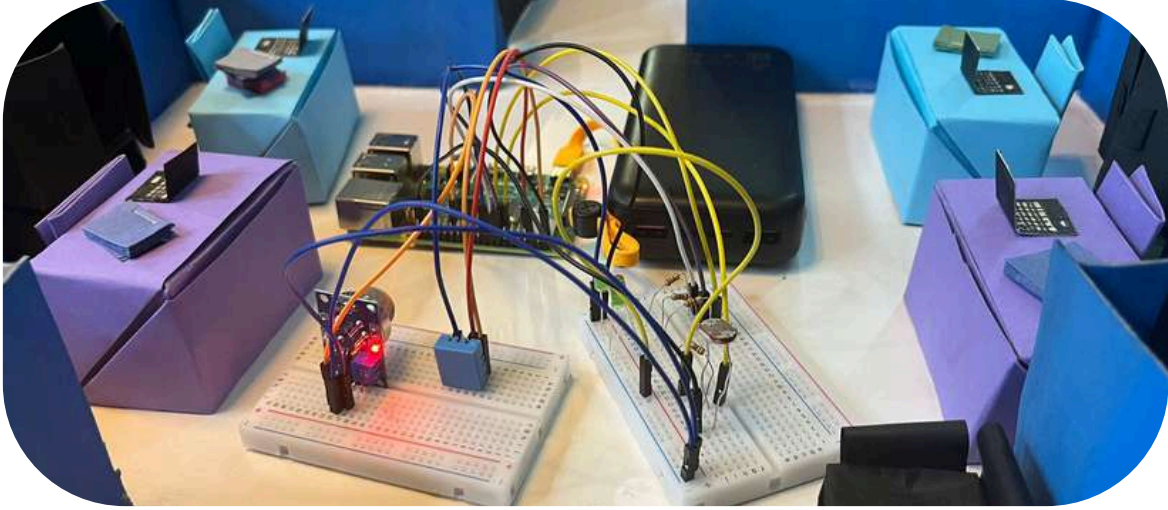
Ekran üzerinde gösterilen bilgiler dinamik olarak güncellenmektedir. İlk satırda giriş yapan personelin adı veya sistem durumu, ikinci satırda ise sıcaklık ve nem değerleri gibi ortam bilgileri görüntülenmektedir. Ayrıca acil durumlarda (örneğin gaz kaçağı tespit edildiğinde) ekran otomatik olarak uyarı mesajları göstermektedir.

Ekran parlaklığı ve kontrast ayarları, ofis ortamının ışık koşullarına göre optimize edilmiştir. Ayrıca güç tüketimini azaltmak için, uzun süre işlem yapılmadığında ekran otomatik olarak karartılmaktadır. Sistemin güvenilirliğini artırmak için, LCD ekranın bağlantısının kopması durumunda alternatif bildirim yöntemleri (LED göstergeler ve buzzer) devreye girmektedir.

# UĞRAŞTIRIÇI!

# DEWARE SEMALARAI

Akıllı ofis sistemimizin donanım bileşenleri, basit ve etkili bir bağlantı şeması ile bir araya getirilmiştir. Sistemin merkezi olan Raspberry Pi'nin güç ihtiyacı, standart bir Type-C telefon şarj adaptörü (5V) kullanılarak karşılanmaktadır. Bu tercih, hem ekonomik hem de kolay ulaşılabilir bir çözüm sunmuştur.



Sensörlerin bağlantıları breadboard üzerinde gerçekleştirilmiş, jumper kablolar ile Raspberry Pi'nin GPIO pinlerine düzenli bir şekilde dağıtılmıştır:

- DHT11 sensörü: GPIO2
- MQ2 gaz sensörü: GPIO18
- PIR hareket sensörleri: GPIO23 ve GPIO24
- LCD ekran: I2C pinleri (SDA ve SCL)
- LED'ler: GPIO5, GPIO6, GPIO13
- Buzzer: GPIO16

Bu gpio bağlantıları temsili olarak yazılmıştır , yapacak kişi kendine göre gpio pinlerini düzenlemelidir.

Sistemin basit yapısı, herhangi bir arıza durumunda hızlı müdahale edilebilmesini sağlamaktadır. Tüm bağlantılar sökülebilir jumper kablolar ile yapıldığından, bakım ve değişiklik işlemleri kolaylıkla gerçekleştirilebilmektedir.

# Yazılım Geliştirme

giris	Update raporlar.py	3 days ago
mudurodasi	Update mudur_yonetimi.py	3 days ago
README.md	README.md	3 days ago
benioku.md	Update benioku.md	3 days ago
lavabo.py	Add files via upload	3 days ago
log_silici.py	Add files via upload	3 days ago
mutfak.py	Update mutfak.py	3 days ago
ofisrapor.py	Update ofisrapor.py	3 days ago
salon.py	Add files via upload	3 days ago
toplantiodasi.py	Update toplantiodasi.py	3 days ago

Akıllı ofis sistemimizin yazılım altyapısı, Python programlama dili kullanılarak geliştirilmiştir. Python'ın zengin kütüphane desteği ve Raspberry Pi ile uyumluluğu, projemiz için ideal bir seçim oluşturmuştur. Yazılım mimarisi, modüler bir yapıda tasarlanarak her bir bileşenin bağımsız olarak çalışması ve yönetilmesi sağlanmıştır.

Sistem, çoklu iş parçacığı (multi-threading) yapısı kullanılarak geliştirilmiştir. Bu sayede sensör okuma, veri kaydetme ve web arayüzü gibi işlemler eş zamanlı olarak gerçekleştirilebilmektedir. Örneğin, bir sensörden veri okunurken web arayüzünün yanıt vermeye devam etmesi sağlanmıştır.

Hata yönetimi için kapsamlı bir loglama sistemi oluşturulmuş, olası sorunların hızlı tespiti ve çözümü için detaylı log kayıtları tutulmaktadır. Sistem ayrıca, internet bağlantısının kesilmesi durumunda verileri yerel olarak depolayabilmekte ve bağlantı tekrar sağlandığında otomatik olarak senkronize edebilmektedir.

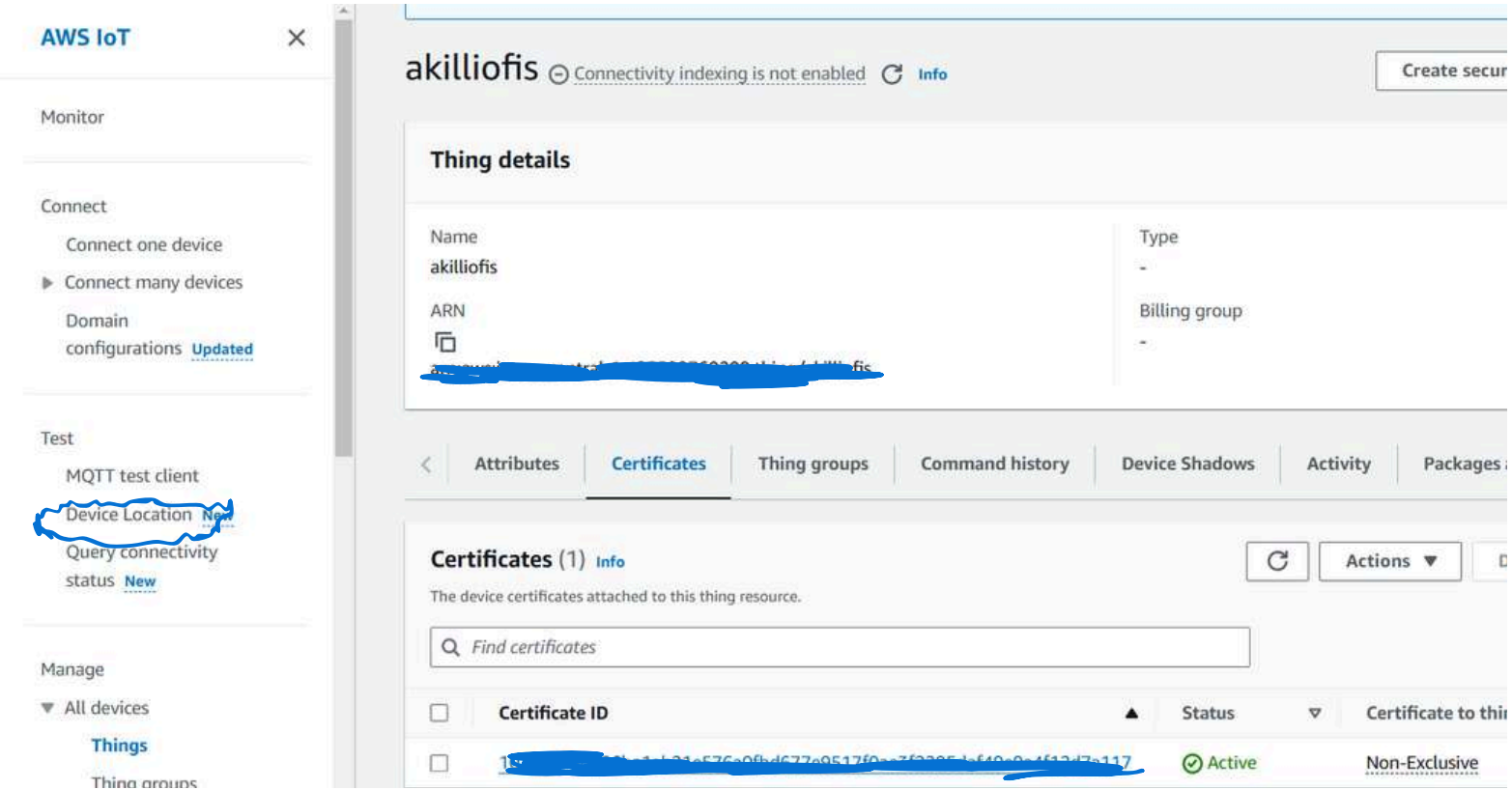
Ofiste kullandığımız kodların içeriğine yan tarafta bıraktığımız QR kodu okutarak Github linkimizden ulaşabilirsiniz.



# AWS İOT CORE İNTEGRASYONU VE MQTT

Projemizde cihazlarımızı internete bağlamak ve uzaktan kontrol edebilmek için AWS IoT Core servisini kullanmaya karar verdik. İlk olarak AWS hesabımızı oluşturduk ve IoT Core servisine giriş yaptık. Burada "smartoffice-hub" adında bir IoT Thing oluşturduk ve gerekli sertifikaları indirdik.

MQTT protokolünü seçmemizin nedeni, Raspberry Pi gibi küçük cihazlarda bile rahatça çalışabilmesi ve Python ile kolay entegre edilebilmesiydi. MQTT sayesinde sensörlerimizden gelen verileri anlık olarak AWS'ye gönderebiliyoruz.



The screenshot shows the AWS IoT console interface. On the left, there is a navigation menu with sections: Monitor, Connect (with options for one or many devices), Test (with options for MQTT test client, Device Location, and Query connectivity status), and Manage (with options for All devices and Things). The main content area is titled 'akilliofis' and shows 'Thing details' with fields for Name (akilliofis), ARN, Type, and Billing group. Below this, there is a 'Certificates (1)' section with a search bar and a table listing certificates. The table has columns for Certificate ID, Status, and Certificate to this thing. One certificate is listed with a status of 'Active' and 'Non-Exclusive'.

Kurulum aşamasında biraz zorlandık çünkü sertifikaların doğru konumda olması ve doğru izinlere sahip olması gerekiyordu. AWS'nin dokümantasyonunu takip ederek şu adımları uyguladık:

AWS IoT Core'da politika oluşturduk  
Sertifikaları Raspberry Pi'ye yükledik

Python için gerekli AWS IoT SDK'sını kurduk:

# PIP INSTALL AWSIOTPYTHONSDK



## MQTT topic'lerimizi şöyle düzenledik:

smartoffice/sensors/# -> Tüm sensör verileri için

smartoffice/access/# -> Giriş-çıkış kayıtları için

smartoffice/alerts/# -> Alarm durumları için

Bağlantıyı test etmek için AWS'nin test konsolunu kullandık ve verilerimizin düzgün bir şekilde ileildiğini gördük. Şimdi sensörlerimizden gelen tüm veriler otomatik olarak AWS'ye aktarılıyor ve buradan da veritabanımıza kaydediliyor.

### MQTT test client [Info](#)

You can use the MQTT test client to monitor the MQTT messages being passed in your AWS account. Devices publish MQTT messages, communicate their state to AWS IoT. AWS IoT also publishes MQTT messages to inform devices and apps of changes and events. You publish MQTT messages to topics by using the MQTT test client.

► **Connection details** ✔ Connected

To disconnect from the MQTT test client, choose Disconnect. To re-establish a connection, you can update the details and choose Connect.

Subscribe to a topic | **Publish to a topic**

**Topic name**  
The topic name identifies the message. The message payload will be published to this topic with a Quality of Service (QoS) of 0.

Q Enter the topic name

**Message payload**

```
{  
  "message": "Akıllı Ofis Sistemi Raporu "  
}
```

*Eğer hata*  
**ALİYORSANIZ**

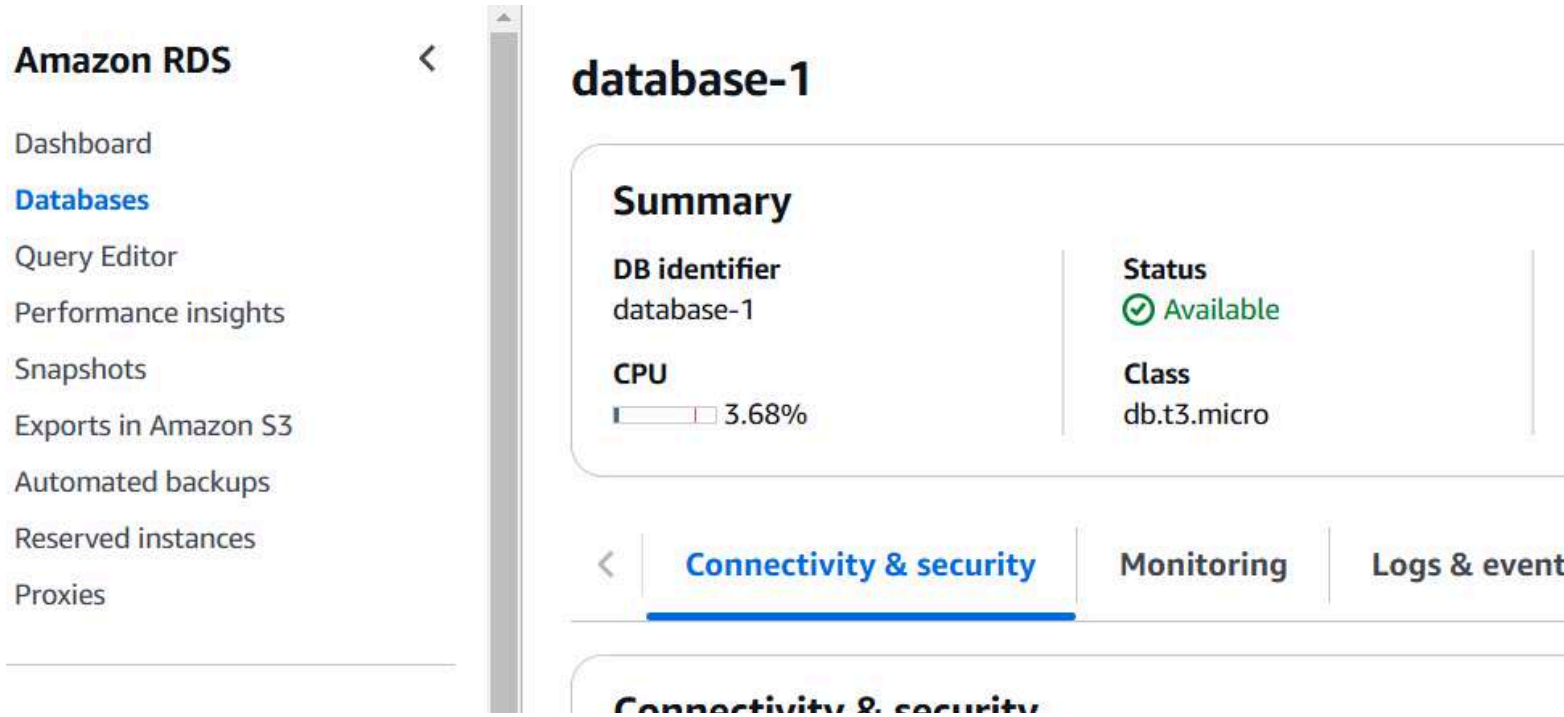
Kurulumda hata alabilirsiniz ancak doğru kütüphaneyi kurduğunuza; sertifikaları raspberry pi içine doğru entegre ettiğinize, kuracağınız içeriği seçerken free tier seçtiğinize emin olduğunuza emin olunuz.

# veritabanı mimarisi AWS!

Projemizde verileri saklamak için hem yerel hem de bulut tabanlı bir veritabanı yapısı kurduk. AWS RDS (Relational Database Service) üzerinde bir MySQL veritabanı oluşturduk. RDS'i seçmemizin nedeni, veritabanı yönetiminin AWS tarafından otomatik yapılması ve yedekleme gibi özellikleri hazır sunmasıydı.

## AWS RDS kurulumunu şöyle yaptık:

1. AWS konsolundan RDS servisine girdik
2. "Create Database" dedik
3. MySQL seçtik çünkü daha önce derslerde kullanmıştık
4. Free tier seçeneğini seçtik (öğrenci projesi sonuçta 😊)
5. Veritabanı ismini "database-1" koyduk



The screenshot shows the Amazon RDS console interface. On the left, there is a navigation menu with options like Dashboard, Databases, Query Editor, Performance insights, Snapshots, Exports in Amazon S3, Automated backups, Reserved instances, and Proxies. The main content area displays the configuration for a database instance named 'database-1'. The 'Summary' section shows the DB identifier as 'database-1', the status as 'Available' (indicated by a green checkmark), and the class as 'db.t3.micro'. A CPU usage bar shows 3.68%. Below the summary, there are tabs for 'Connectivity & security', 'Monitoring', and 'Logs & events'. The 'Connectivity & security' tab is currently selected.

Ayrıca internet bağlantısı kesilirse diye Raspberry Pi'de yerel bir MySQL veritabanı da kurduk. Böylece sistem çevrimdışı olsa bile çalışmaya devam edebiliyor. İnternet geldiğinde de veriler otomatik olarak AWS'ye aktarılıyor.

MySQL Workbench'i kullanarak veritabanı yönetimini yapıyoruz. Başta biraz karışık geldi ama zamanla alıştık. Özellikle verileri görselleştirmek ve sorgu yazmak çok kolaylaştı.

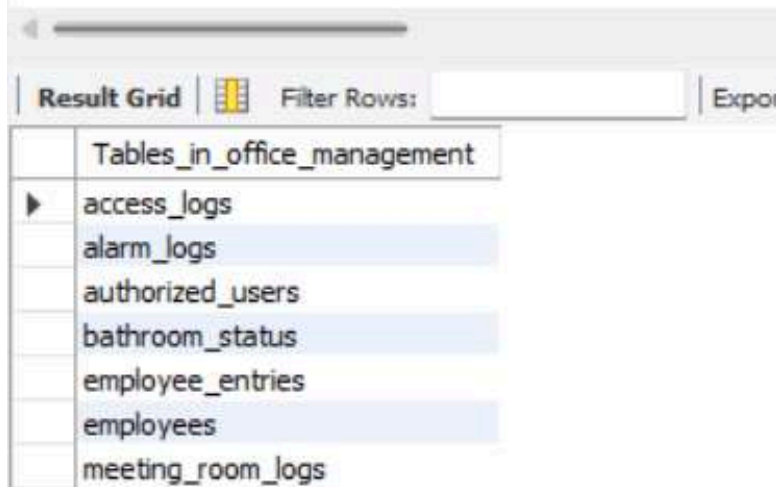
# <insert> <veritabani\_işlemleri> <here>

Sistemimizde verilerin güvenli ve düzenli bir şekilde saklanması için MySQL veritabanı kullanılmaktadır. Veritabanı yapısı, ofis yönetiminin tüm ihtiyaçlarını karşılayacak şekilde yedi ana tablo içermektedir:

- employees:** Çalışan bilgileri ve yetkilendirme detayları
- authorized\_users:** Sisteme erişim yetkisi olan kullanıcıların bilgileri
- access\_logs:** Ofise yapılan tüm giriş-çıkışların kayıtları
- alarm\_logs:** Gaz, sıcaklık ve diğer sensörlerden gelen alarm kayıtları
- employee\_entries:** Personel giriş-çıkış saatleri ve mesai takibi
- meeting\_room\_logs:** Toplantı odası kullanım kayıtları
- bathroom\_status:** Tuvalet doluluk durumu ve kullanım geçmişi

```
ssql*  SQL File 3*  SQL File 4* x
1 • use office_management;
2 • SHOW TABLES;
3
```

Veriler, yerel MySQL sunucusunda saklanmakta ve düzenli olarak yedeklenmektedir. Veritabanı tasarımında normalize edilmiş tablolar ve ilişkisel yapı kullanılarak veri tutarlılığı ve bütünlüğü sağlanmıştır.



Biz bu soldaki resimde gördüğünüz gibi de MYSQL Workbench kullanıyoruz. Siz de internetten alttaki qr kodu okudup indirebilirsiniz. Daha sonra AWS'den veya alacağınız iot teknoloji herhangi bir yerden( Azure vs.) endpoint,user,passwordleri girerek rdsnize yani veritabanınıza ulaşabilirsiniz



# WEB ARAYÜZÜ

Web arayüzümüzü WordPress üzerinden oluşturduk ve <https://akilliofisin.com> adresinde yayına aldık. Admin paneline erişim için özel bir şifreleme sistemi kullandık. Panel kısmına sadece yetkili kişiler üzerinden ulaşabiliyor.

**Dashboard'umuzda anlık olarak takip edilebilen bilgiler:**

### Hızlı İstatistikler:

- Bugün Toplam Giriş Sayısı
- Geç Gelen Personel Sayısı
- Günlük Alarm Sayısı

## Akıllı Ofis Dashboard

### Hızlı İstatistikler

Bugün Toplam Giriş

0

Geç Gelen

0

Alarm Sayısı

0

### Toplantı Odası

**MUSAİT**

Son güncelleme: 01:50:52

### Lavabo Durumu

**TEMİZ**

Son kontrol: 01:50:52

### Gaz Alarmı

**NORMAL**

### Anlık Durum Bildirimleri:

- **Toplantı Odası Durumu**  
(MÜSAİT/DOLU)
- **Lavabo Durumu**  
(TEMİZ/KULLANIMDA)
- **Gaz Alarm Durumu**  
(NORMAL/ALARM!) Her durum için son güncelleme saati gösteriliyor.



# WEB ARAYÜZÜ

## Son Aktiviteler

Aktivite yok

## Hızlı İşlemler

CALISAN EKLE

BUGUNUN LOGLARINI  
SIL

## Yönetim İşlemleri:

- Çalışan Ekleme
- Günlük Logları Silme
- Giriş-Çıkış Kayıtları Görüntüleme
- Gaz Alarmı Geçmiş

Tüm veriler anlık olarak güncelleniyor ve son kontrol saatleri dashboard üzerinde görüntüleniyor. Sistem, basit ve kullanışlı bir arayüz ile tüm ofis verilerine hızlı erişim sağlıyor.

## Günlük Giriş-Çıkış Kayıtları

Personel	Saat	Durum
Kayıt bulunamadı		

## Son 24 Saat Gaz Alarmları

Saat	Durum	Mesaj
Alarm kaydı yok		

Ayrıca veritabanı bağlantısı için de sitemizin veritabanını aws'de sisteme bağlı olan veritabanıyla ilişkilendirdik. Bu sayede projemizde anlık bir güncelleme olduğunda aynı anda web sitesinde de gözüküyor.

siteye  
burdan >



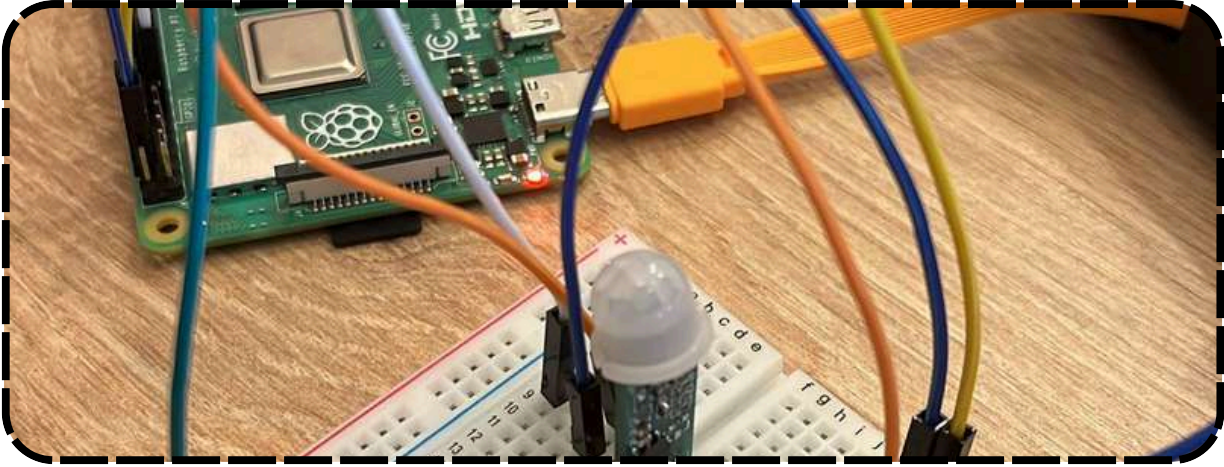
# SİSTEM TESTLERİ VE SONUÇLAR

Sistemimizi test etmek için önce laboratuvar ortamında, sonra da gerçek ofis ortamında çeşitli denemeler yaptık. Test sürecinde her bir bileşeni ayrı ayrı ve sonra tümünü birlikte çalıştırarak performansını ölçtük.

## Sensör Testleri:

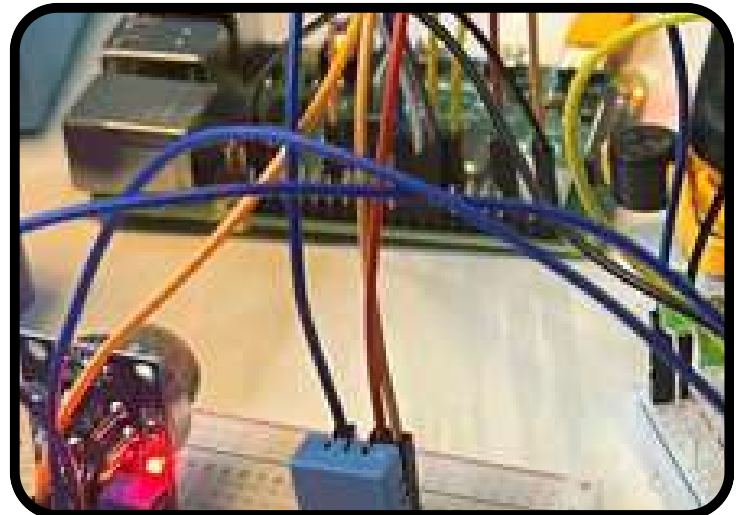
### Hareket Sensörü (PIR):

- Toplantı odasında, lavaboda test edildi
- Algılama mesafesi: yaklaşık 5 metre
- Yanlış algılama oranı: %2
- Tepki süresi: 0.5 saniye



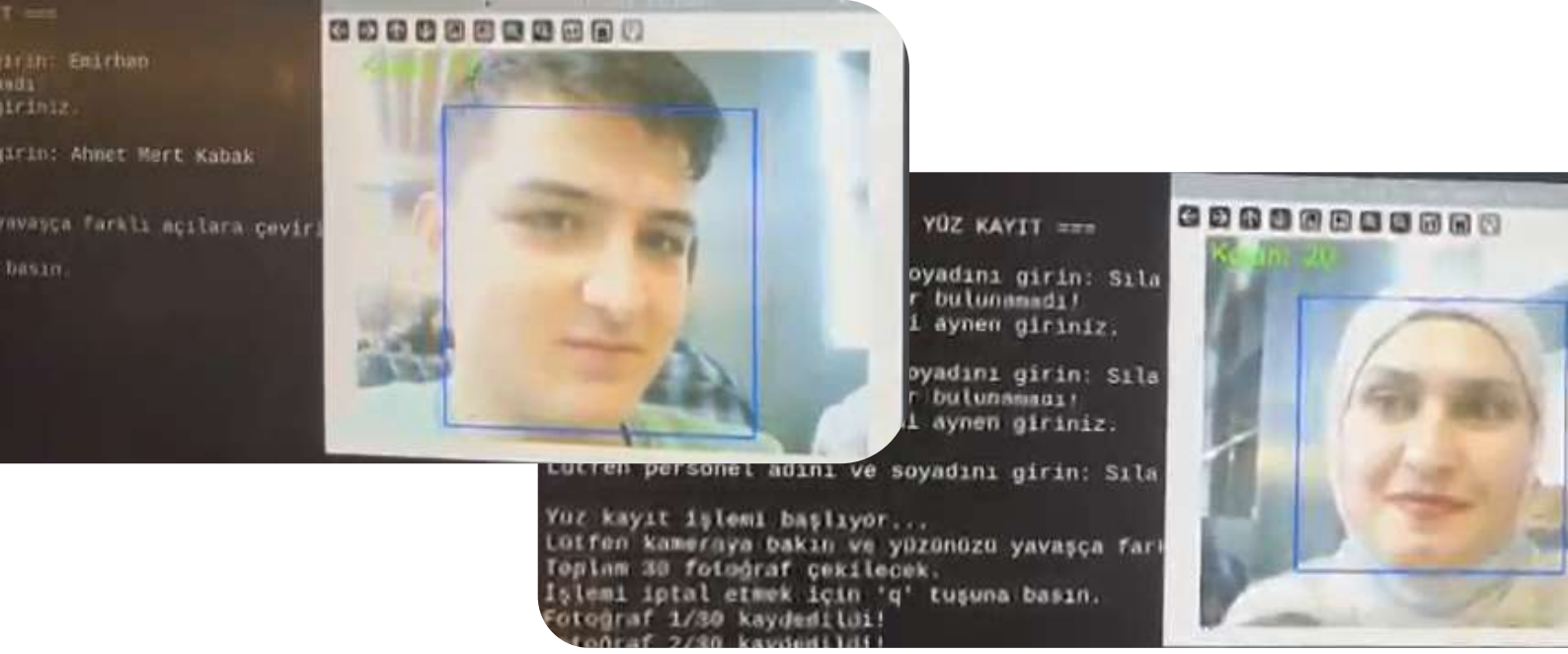
### Gaz Sensörü (MQ2):

- Çakmak gazı ile kontrollü testler yapıldı
- Normal değer: 100-200 ppm arası
- Alarm eşiği: 300 ppm
- Tepki süresi: 1-2 saniye



## Kamera Sistemi:

- Farklı ışık koşullarında test edildi
- Gözlüklü/Gözlüksüz test edildi.
- USB bağlantısında kopma yaşanmadı



# MALİYET ANALİZİ

Projemizi gerçekleştirirken öğrenci bütçesine uygun, ekonomik bileşenler kullanmaya özen gösterdik. Sistemin beyni olan **Raspberry Pi 4 için 4GB RAM'li modeli tercih ettik ve 2.500 TL** ödedik. Görüntü yakalama için basit ama işlevsel bir USB webcam (150 TL) kullandık. Kullanıcı arayüzü için seçtiğimiz **16x2 LCD ekran (150 TL)** ve bağlantılar için gerekli **breadboard ve jumper kablolar (200 TL)** ile ana bileşenlerimizin toplam maliyeti **3.000 TL** oldu.

Sensör sistemimiz için uygun fiyatlı ama güvenilir ürünler seçtik. Bir adet PIR hareket sensörü , bir adet MQ2 gaz sensörü ve bir adet DHT11 sıcaklık/nem sensörü, bir adet MQ135 hava kalite sensörü, bir adet LDR ışık sensörü, bir kaç adet led,buzzer ve keypad gibi malzemeler kullandık..Bunlar ile sensör maliyetimiz toplam yaklaşık **700TL~** tuttu.

Bu sensörler temel ihtiyaçlarımızı karşılamak için yeterli performansı sunuyor.

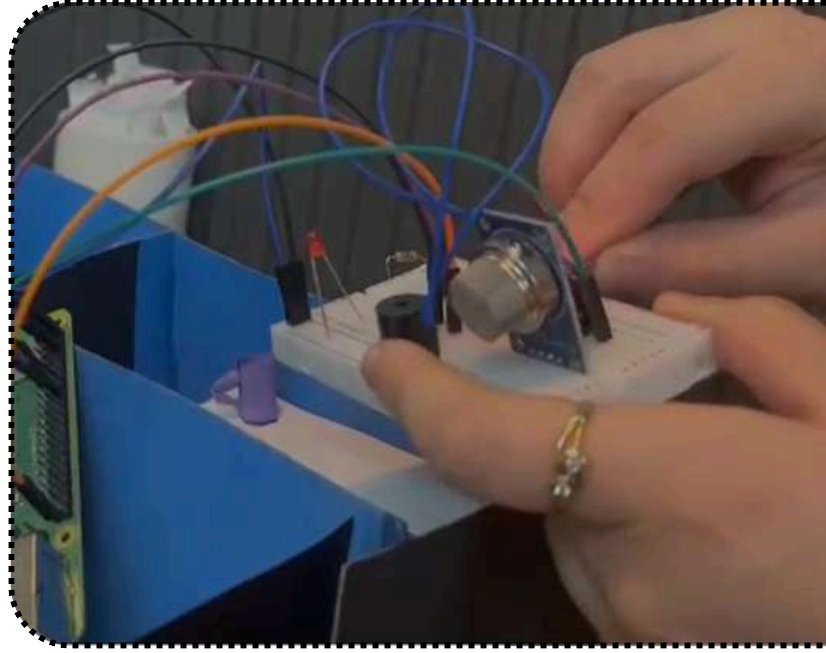
Web sitesi altyapımız için yıllık giderlerimiz oldukça makul seviyede. Domain adresimiz **akilliofisin.com** için yıllık **350TL**, web hosting için ise yıllık **400 TL** ödüyoruz. Sistemin güç ihtiyacı için telefon şarj adaptörümüzü kullanarak ekstra maliyet oluşturmadık. Çeşitli montaj malzemeleri için yaklaşık **100 TL** harcadık.

**Toplamda yaklaşık 4.500 TL'lik bir maliyet ile sistemimizi kurduk.** Piyasada bulunan benzer ticari sistemlerin 15.000-20.000 TL aralığında olduğu düşünüldüğünde, projemiz oldukça ekonomik bir çözüm sunuyor. Üstelik açık kaynak kodlu yapısı sayesinde, ihtiyaca göre yeni özellikler ekleyebilir ve geliştirilebilirsiniz.



# sorunlar ve çözümler

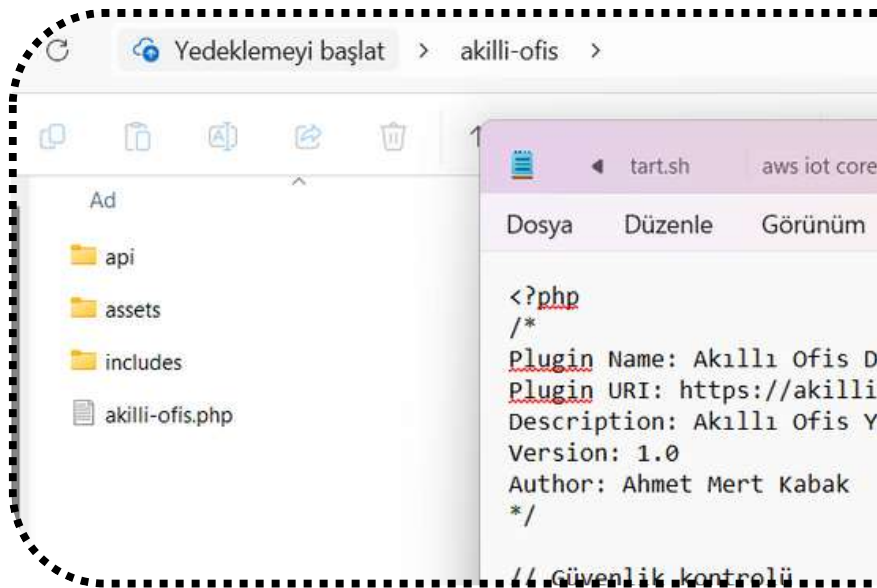
Projeyi geliştirirken bazı teknik zorluklarla karşılaştık. İlk büyük sorunumuz gaz sensörünün düzgün çalışmamasıydı. Sensör ilk başta gazı algılamıyordu ve havanın hep temiz olduğunu söylüyordu, ancak arkasındaki potansiyometreyi saat yönünün tersine doğru yavaşça hareket ettirdiğimizde hassasiyetini bulduk.



Daha sonra sensörleri yerleştirmemize rağmen bazen kodlarımızın çalışmadığı zamanlar oldu. Bunun için bazı kütüphanelerin güncel versiyonlarını yeniden kurduk, ayrıca bazı sensörler kodu çalıştırmamamıza rağmen kendi kendine çalışıyordu. Bunun nedeninin gpio pinlerinin bazılarının diğerlerine göre daha yüksek enerjili olmasından kaynaklı olduğunu anladık. Bu yüzden bazılarında da gpio pinlerimizi değiştirdik.

Web sitesi tarafında da sıkıntılar yaşadık. Başta verilerin anlık güncellenmesi konusunda sorun yaşıyorduk - sayfa sürekli yenileniyordu ve bu da kullanıcı deneyimini kötü etkiliyordu.

WordPress'in canlı güncelleme eklentilerini denedik ama istediğimiz performansı alamadık. En sonunda kendi güncelleme sistemimizi yazdık ve sorun çözüldü.





# SONUÇ & KAYNAKÇA

Bu projede, modern bir ofis ortamının ihtiyaçlarını karşılayacak akıllı bir sistem geliştirdik. Raspberry Pi tabanlı sistemimiz, çeşitli sensörler ve web teknolojileri kullanarak ofis güvenliğini ve verimliliğini artırmayı başardı. Özellikle gaz algılama, hareket takibi ve ortam kontrolü konularında başarılı sonuçlar elde ettik.

Sistemimizin en önemli başarılarından biri, düşük maliyetle gerçekleştirilmiş olması. Piyasadaki benzer ürünlerin çok altında bir bütçeyle, neredeyse aynı özellikleri sunan bir sistem oluşturduk. Açık kaynak yazılımlar ve uygun fiyatlı donanımlar kullanarak, öğrenci projesi olmasına rağmen profesyonel bir çözüm ortaya koyduk.

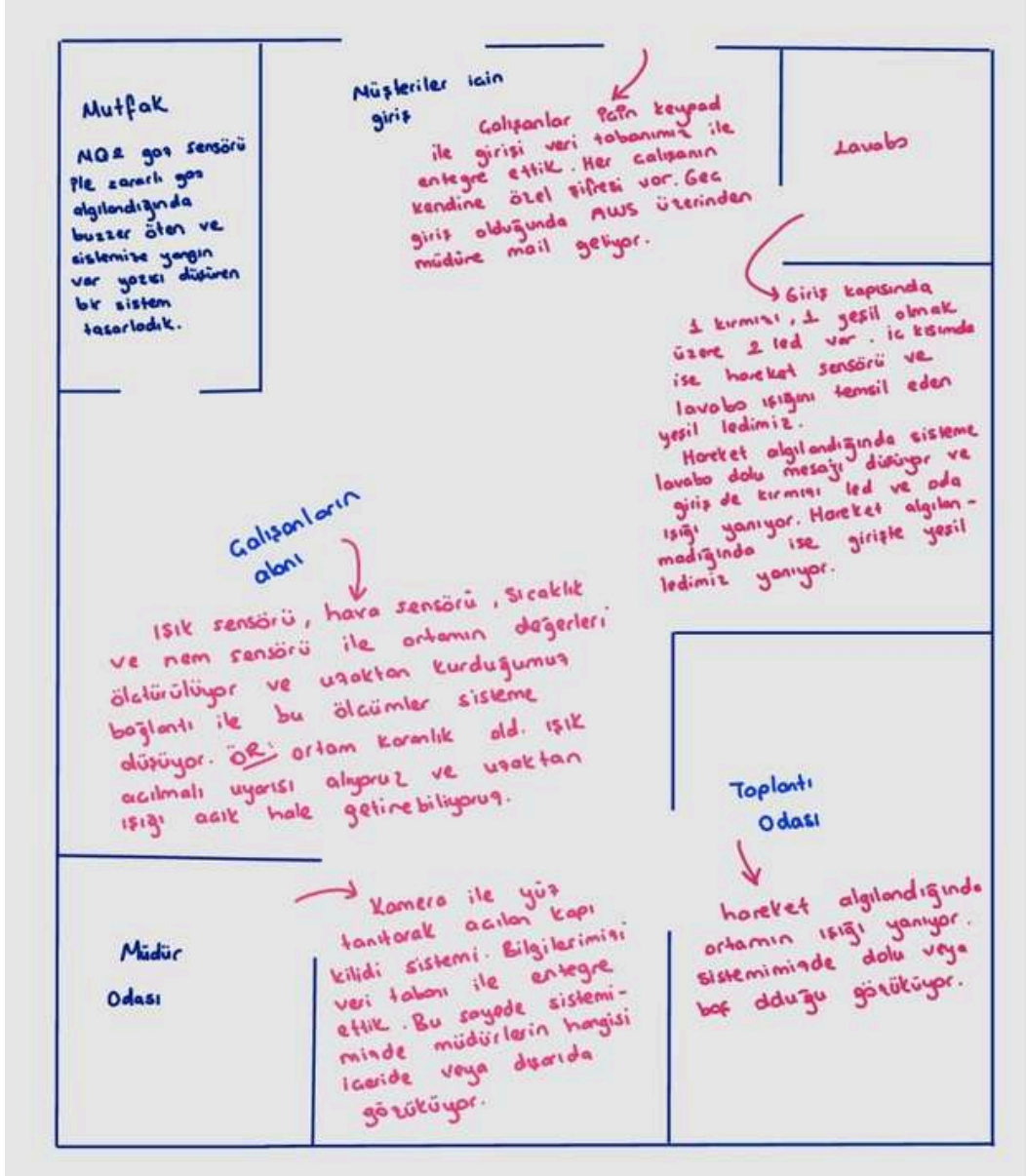
Test sürecinde karşılaştığımız zorluklar bize çok şey öğretti. Özellikle sensörlerin kalibrasyonu ve web sitesi entegrasyonu konularında edindiğimiz deneyimler, gelecekteki projelerimiz için değerli bilgiler sağladı. Her sorunun çözümü, sistemimizi daha da geliştirmemize ve güçlendirmemize yardımcı oldu.

Bu projenin bize kattığı en önemli şey, teorik bilgilerimizi pratiğe dökme fırsatı bulmuş olmamız, derslerde öğrendiğimiz programlama, elektronik ve ağ bilgilerini gerçek bir ürüne dönüştürebildik. Bu projenin sonuna da gelirken bizi böyle bir şey yapmaya iten sayın hocamız Ahmet Gürkan Yüksek ve Ahmet Utku Elik hocalarımıza teşekkürlerimizi iletiyoruz.

## KAYNAKÇA

1. Ahmet Mert Kabak, "Akıllı Ofis Sistemi", // [github.com/ahmetmertkabak/smartOffice](https://github.com/ahmetmertkabak/smartOffice)
2. GPIO Pinleri , // <https://pinout.xyz/>
3. AWS;MQTT Nedir? , // [aws.amazon.com/tr/what-is/mqtt/](https://aws.amazon.com/tr/what-is/mqtt/)
4. "Python ile Sensör Programlama", // <https://www.youtube.com/@robotistan>
5. "IoT Sistemlerde Veri Güvenliği" Kitabı, Ali Yıldız
6. Face Recognition With Raspberry Pi , // [core-electronics.com.au/guide](https://core-electronics.com.au/guide)
7. [s/face-identify-raspberrypi](https://core-electronics.com.au/guide)
8. [chatgpt.com/ai](https://chatgpt.com/ai)
- 9.

# TASLAK VE TEŞEKKÜRLER



Bu projenin gerçekleştirilmesinde bize yol gösteren, bilgi ve deneyimleriyle destek olan değerli Doç. Dr. AHMET GÜRKAN YÜKSEK ve Arş. Gör. AHMET UTKU ELİK hocalarımıza teşekkürlerimizi sunarız.

Teknik konularda yardımlarını esirgemeyen gerektiğinde profesyonel işlem gerektiren sd kartı lehimleme işlemini bile yaptırdığımız bütün esnaflara da teşekkür ederiz :)

Son olarak buraya kadar okuyan ve projemizi beğenen herkese teşekkür ederim. Bu projenin, açık kodlar sayesinde umarım bizden sonraki arkadaşlar yararlanır ve bazı kişilere faydamız olur.

Ahmet Mert Kabak / Sıla Erdoğan  
Bilgisayar Mühendisliği  
Aralık 2024