



---

**CUMHURİYET ÜNİVERSİTESİ BİLGİSAYAR MÜHENDİSLİĞİ**  
**BİL 209 NESNELERİN İNTERNETİ**

---

**AKILLI İKLİMLENDİRME SİSTEMİ RAPORU**

**ÖĞRENCİLER:**

- 2022123006 – AYŞENUR NAVRUZ
- 2022123011 – ELİF NUR ÇAKICI
- 2022123048 – KAMER BALTA
- 2022123042 – MUHAMMED ÇALIŞKAN

## İçindekiler

1. Projenin Tanıtımı ve Amacı .....	2
2. Kullanılan Teknolojilerin ve Araçların Açıklaması.....	2
3. Mikroişlemci ve Yazılım Modelinin İşleyişi.....	6
1- Veri Algılama ve İşleme: .....	6
2- Veri İşleme ve Depolama: .....	7
3- AWS RDS'deki Veri İşleme: .....	7
4. Proje Görseli .....	9
5. Beklenen Sonuçlar ve Sistem Çalışma Tahminleri .....	9
6. Seçim Gerekçeleri.....	9

## 1. Projenin Tanıtımı ve Amacı

Bu proje, bir mikroişlemci ve sensör teknolojileri kullanarak uzaktan kontrol edilebilir bir akıllı iklimlendirme sistemi geliştirmeyi amaçlamaktadır. Sistem, bir seradaki iklim koşullarını izlemek ve optimize etmek için tasarlanmıştır. Kullanıcılar, mobil bir uygulama üzerinden seradaki sıcaklık, nem, hava kalitesi, ışık seviyesi ve toprak nemi gibi verileri takip edebilir ve gerektiğinde sistemi kontrol edebilir. Proje, bitki yetiştiriciliği yapılan seralarda verimliliği artırmak ve manuel müdahale ihtiyacını azaltmak amacıyla oluşturulmuştur.

---

## 2. Kullanılan Teknolojilerin ve Araçların Açıklaması

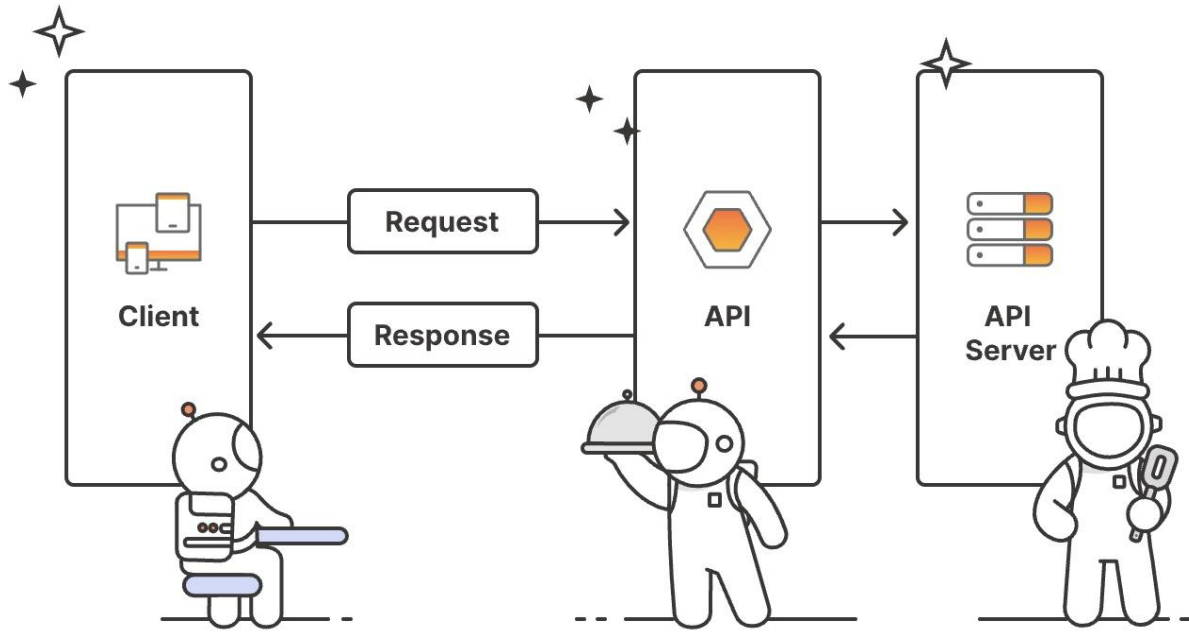
- **ESP32:**

ESP32, düşük maliyetli, yüksek performanslı ve enerji verimli bir mikrodenetleyici kartıdır. Wi-Fi ve Bluetooth özelliklerine sahip olan ESP32, IoT (Internet of Things - Nesnelerin İnterneti), gömülü sistemler, akıllı cihazlar ve otomasyon projelerinde sıkça kullanılmaktadır. Çift çekirdekli bir mikroişlemciye sahip olan bu cihaz, güçlü işlem kapasitesinin yanı sıra birçok sensör ve cihaz ile uyumlu bir şekilde çalışabilir. **ESP32**, akıllı iklimlendirme projemizde sistemin merkezi kontrol ünitesi olarak kullanılmıştır. Bu cihaz, çevreden aldığı verileri işleyerek, mobil uygulama üzerinden kullanıcıya bilgi ve öneriler sunar.



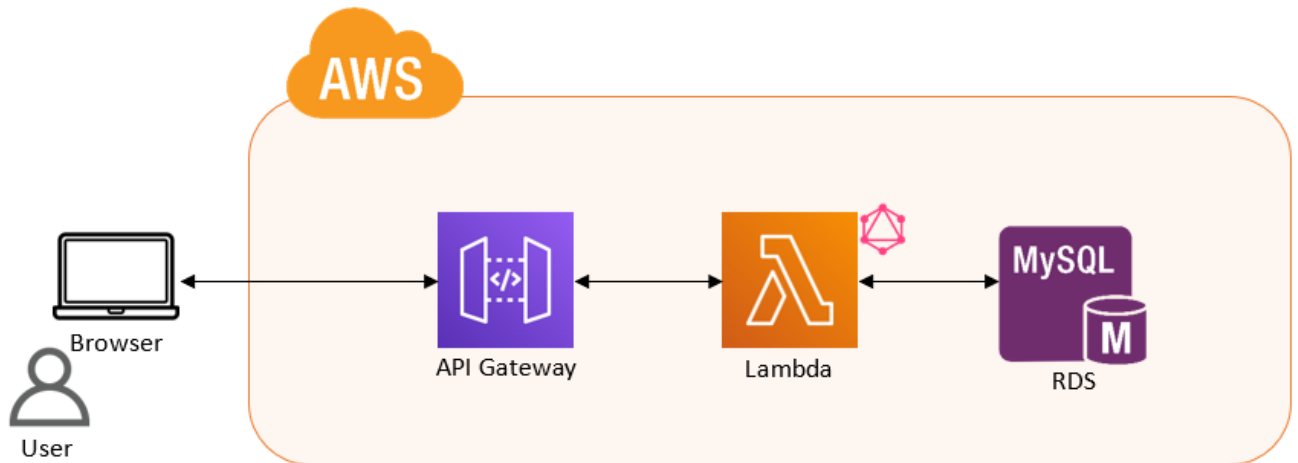
Şekil 1: ESP32 Kartı

- **YL-69 Toprak Nem Sensörü:** Seradaki toprak nemini ölçerek sulama sistemi için veri sağlar.
- **Sıcaklık ve Nem Sensörü (DHT11):** Seradaki ortam sıcaklığı ve nem seviyelerini ölçmek için kullanılır.
- **Hava Kalitesi Sensörü (MQ-135):** Seradaki hava kalitesini analiz eder ve gerekli durumlarda havalandırmayı kontrol eder.
- **Işık Sensörü (LDR):** Seradaki ışık seviyesini ölçerek aydınlatma sistemini optimize eder.
- **API Mimarisi:** **API (Application Programming Interface)**, bir yazılım uygulamasının diğer yazılımlar veya hizmetlerle iletişim kurmasını sağlayan bir arayüzdür. API, uygulamalar arasında veri alışverişini ve işlevselliği mümkün kılar, genellikle belirli işlevlerin gerçekleştirilmesine olanak tanır. API'ler, bir yazılımın nasıl kullanılacağını ve diğer yazılımlarla nasıl etkileşimde bulunacağını belirten talimatları içerir.



Şekil 2: API işleyiş diyagramı

- **Amazon AWS:**  
Amazon Web Services (AWS), Amazon tarafından sunulan ve çevrimiçi ortamda çeşitli bilgi işlem hizmetlerini kullanmamıza olanak tanıyan bir bulut bilişim platformudur. **Akıllı İklimlendirme Sistemi** projemizde AWS'ye bağlı aşağıdaki hizmetler kullanılmıştır:
- **Amazon RDS (Relational Database Service):**  
Bu hizmet, ilişkisel veritabanlarının bulut ortamında kolayca oluşturulmasını, çalıştırılmasını ve ölçeklendirilmesini sağlar. Projede, sensör verilerinin saklanması ve analiz edilmesi için MySQL veritabanı motoru kullanılarak bir veritabanı oluşturulmuştur.
- **AWS Lambda:**  
AWS Lambda, sunucusuz bir hesaplama hizmetidir ve projede veri tabanı API kodlarının çalıştırılmasında kullanılmıştır. Sensörlerden gelen verilerin işlenmesi ve mobil uygulamaya aktarılması gibi süreçler Lambda üzerinden yönetilmiştir. Kod, yalnızca belirli bir olay tetiklendiğinde çalıştırıldığı için sistem maliyetleri optimize edilmiştir.
- **API Gateway:**  
API Gateway, istemcilerden gelen API isteklerini alır ve Lambda fonksiyonlarına yönlendirir. Bu projede, mobil uygulamadan gelen talepler API Gateway aracılığıyla Lambda fonksiyonlarına iletilmiştir. Bu mimari, sistemin esnekliğini artırarak farklı cihazlarla entegrasyonu kolaylaştırmıştır.
- **Amazon EC2 (Elastic Compute Cloud):**  
Amazon EC2, bulut üzerinde sanal sunucular oluşturulmasını ve yönetilmesini sağlar. Projede, WebSocket protokolü üzerinden gerçek zamanlı iletişim ve video aktarımı sağlamak için Node.js tabanlı bir uygulama bu sunucuda çalıştırılmıştır. Bu yapı, sera içerisindeki sensörlerden gelen verilerin ve kameradan alınan görüntülerin kullanıcıya hızlı bir şekilde aktarılmasını sağlamıştır.



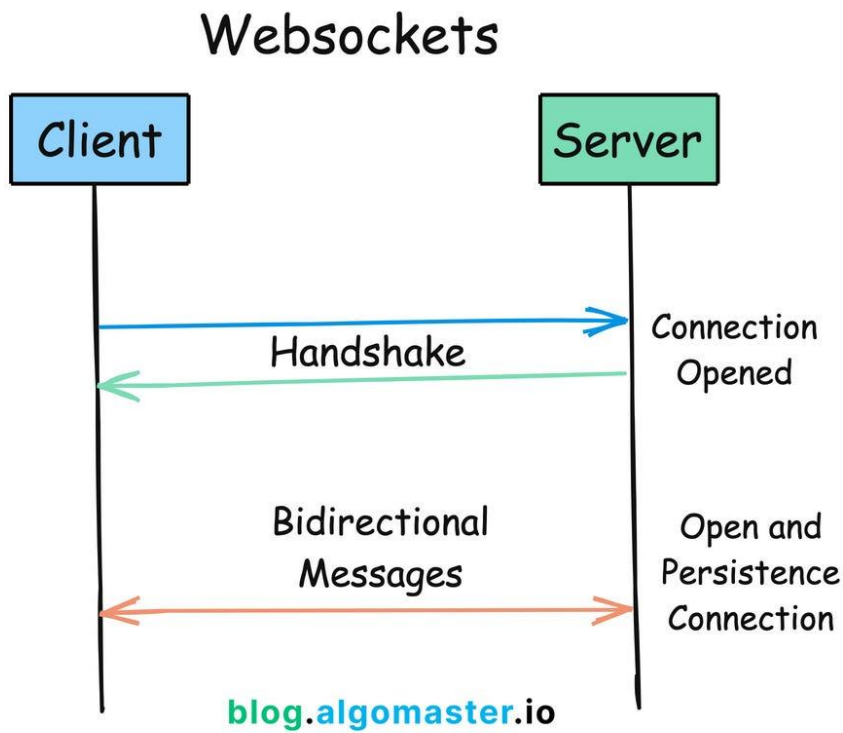
Şekil 3: AWS Servislerinin İşleyişi

- **WebSocket Protokolü:** WebSocket protokolü, istemci (örneğin mobil uygulama) ile sunucu arasında çift yönlü, sürekli bir iletişim kanalı kurulmasını sağlayan bir internet iletişim protokolüdür. Projemizde, ESP32 ile mobil uygulama arasındaki iletişimde

WebSocket protokolü kullanılmıştır. Protokol, sensör verilerini gerçek zamanlı olarak iletme amacıyla kullanılmaktadır.

- **Sistem Hareketleri:**

- **Aktuatör Hareketi:**Sistem belirli eşik değerlere ulaşınca aktuatör hareketlerini API vasıtası ile veri tabanındaki failure tablosuna gönderiyor.Örneğin;Sulama eşik değerine ulaşılınca su motoru açılıyor ve veri tabanına “Sulama Açıldı” saat ve tarih ile beraber yazdırılıyor.
- **Günlük Ortalama Ölçümü:**Her günün sonunda o günki sensör verilerinin ortalaması veri tabanındaki daily\_sensor\_data tablosuna gönderiliyor.



Şekil 4: WebSocket Protokolü

- **Java:** Nesne yönelimli bir programlama dilidir ve genellikle Android uygulamaları geliştirmek için kullanılır. Java, güçlü, taşınabilir ve platform bağımsız bir dildir, yani aynı Java kodu farklı cihazlarda çalışabilir.Mobil Uygulamanın back-end kısmı java kullanılarak yazılmıştır.
- **Figma:** **Figma**, web tabanlı bir **tasarım ve prototipleme** aracıdır. Özellikle kullanıcı arayüzü (UI) ve kullanıcı deneyimi (UX) tasarımı yapan profesyoneller tarafından yaygın olarak kullanılır. Figma, tasarım sürecinde işbirliğini kolaylaştırmak için gerçek zamanlı birlikte çalışmayı destekler ve tasarımcılar, geliştiriciler ve diğer paydaşlar arasında etkin bir iş akışı sağlar.Mobil uygulamanın front-end kısmının tamamı figma ile tasarlanıp Android Studio’ya aktarılmıştır.
- **Android Studio:** **Android** platformunda uygulama geliştirmek için kullanılan resmi **IDE (Entegre Geliştirme Ortamı)**'dir. Android Studio, Java (veya Kotlin) dilinde

yazılmış uygulamaları geliştirmek, derlemek, test etmek ve dağıtmak için kapsamlı araçlar sunar. Android Studio, kullanıcı arayüzü tasarımı, emülatörler, hata ayıklama ve performans analizi gibi birçok özellik sağlar.

- **NodeJs: Node.js**, sunucu tarafında çalışan, açık kaynaklı bir JavaScript çalışma ortamıdır (runtime environment). JavaScript, genellikle web tarayıcılarında çalışan bir programlama dili olarak bilinse de, Node.js sayesinde JavaScript, sunucu tarafında da çalışabilir. Bu, JavaScript'in yalnızca tarayıcıda değil, sunucularda da kullanılmasını mümkün kılar. Node.js, hızlı ve ölçeklenebilir ağ uygulamaları geliştirmek için kullanılır. WebSocket protokolü için gerekli kodlar nodejs üzerinden ec2 sunucusunda çalışmaktadır.
- **Arduino IDE/C++**: Arduino IDE, Arduino platformu için kullanılan açık kaynaklı bir geliştirme ortamıdır. Kullanıcıların Arduino kartlarını programlamalarını ve projelerini geliştirmelerini sağlar. C++ dilini temel alarak çalışan Arduino IDE, donanım ile yazılım arasındaki etkileşimi kolaylaştıran bir yapı sunar. Projemizde, Arduino IDE ve C++ kullanılarak sensörlerden gelen verilerin işlenmesi, sulama motorunun ve havalandırma fanının kontrol edilmesi gibi işlemler gerçekleştirilmiştir. Bu yapı, sensör verilerinin doğru bir şekilde alınmasını ve sistemin sorunsuz bir şekilde çalışmasını sağlamaktadır.
- **MySQL**: MySQL, açık kaynaklı bir **ilişkisel veri tabanı yönetim sistemi (RDBMS)**'dir. Verilerin yapılandırılmış bir şekilde saklanması, düzenlenmesini ve işlenmesini sağlar. MySQL, genellikle web uygulamaları, iş yazılımları ve veri analizi projelerinde kullanılan en popüler veri tabanı sistemlerinden biridir. Projede veri tabanı motoru olarak MySQL kullanıldı. Veri Tabanında daily\_sensor\_data ve failure tabloları bulunmaktadır.
- **Donanımlar**:
  - **Sulama Motoru 6V**: Toprak nem seviyesi belirlenen eşiğin altına düştüğünde sulama işlemini gerçekleştirmek için kullanılmıştır.
  - **Havalandırma Fanı 5V**: Sera içindeki hava akışını sağlamak ve sıcaklık seviyelerini dengelemek için kullanılmıştır.
  - **Aydınlatma LED 5V**: Tek renkli bir LED, sera içindeki bitkilerin ışık ihtiyacını karşılamak için kullanılmıştır.

---

### 3. Mikroişlemci ve Yazılım Modelinin İşleyişi

#### 1- Veri Algılama ve İşleme:

- Projede kullanılan sensörler, sera içindeki hava kalitesi, sıcaklık, nem, ışık ve toprak nemi gibi çevresel verileri sürekli olarak algılar.
- Her sensör, belirlenen eşik değerleri aşıldığında veya belirli aralıklarla Python uygulaması tarafından okunur.
- Veriler sensör tipine göre ilgili işlevlerle işlenir:
  - **DHT11 Nem ve Sıcaklık Sensörü**: Dht11 kütüphanesinde bulunan dht.readTemperature() ve dht.readHumidity() fonksiyonları ile sıcaklık ve nem ölçümü yapılıyor.
  - **Toprak nem sensörü**: Toprak nem sensöründen gelen direnç değişim değeri dönüşüm formülü ile yüzde değerine dönüşüyor.

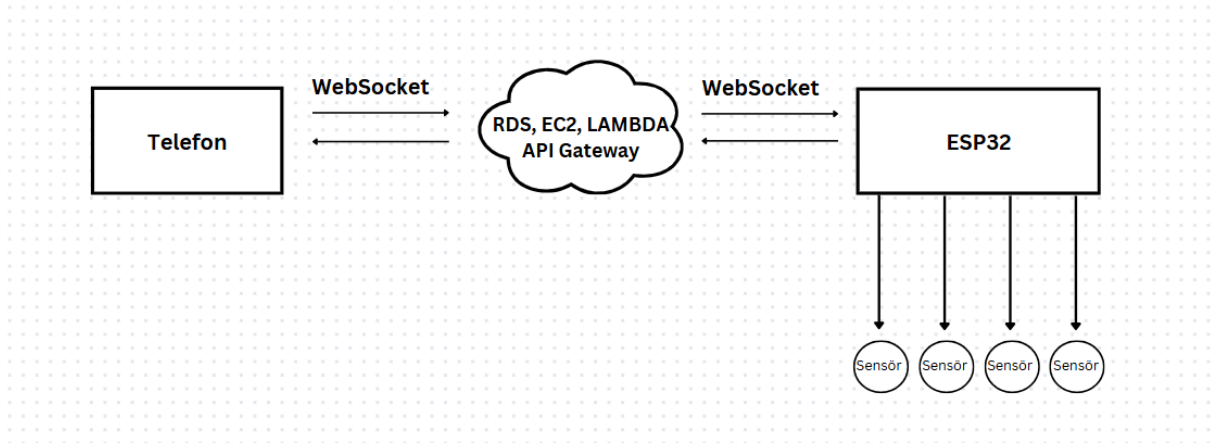
- Mq-135 Hava Kalite sensörü:Buradan gelen voltaj değişim değerini referans voltaj değerlerini kullanarak havada CO2 derişimini ppm olarak ölçtük.

## 2- Veri İşleme ve Depolama:

- ESP32, sensörlerden gelen tüm verileri belirli zaman aralıklarında okur ve işler.
- İşlenen veriler:
  - Anlık Sensör değerleri:Bu değerler önce her bir döngüde kümülatif olarak toplanır.Kümülatif olarak toplanan sensör verilerinin gün sonu ortalaması alınarak veri tabanına gönderilir.Ayrıca anlık değerler Websocket üzerinden anlık olarak telefona gönderilir.
  - Eşik Değer Kontrolü:Ölçülen sensör verileri eşik değerlerle karşılaştırılır.Eğer veri eşik değere ulaşmışsa ilgili aktuatöre sinyal gönderilir.(Örn:Havalandırma Fanının Açılması)

## 3- AWS RDS'deki Veri İşleme:

- Tüm eylemler tarih ve saat bilgisiyle birlikte kaydedilir.



Şekil 5: Akış Diyagramı

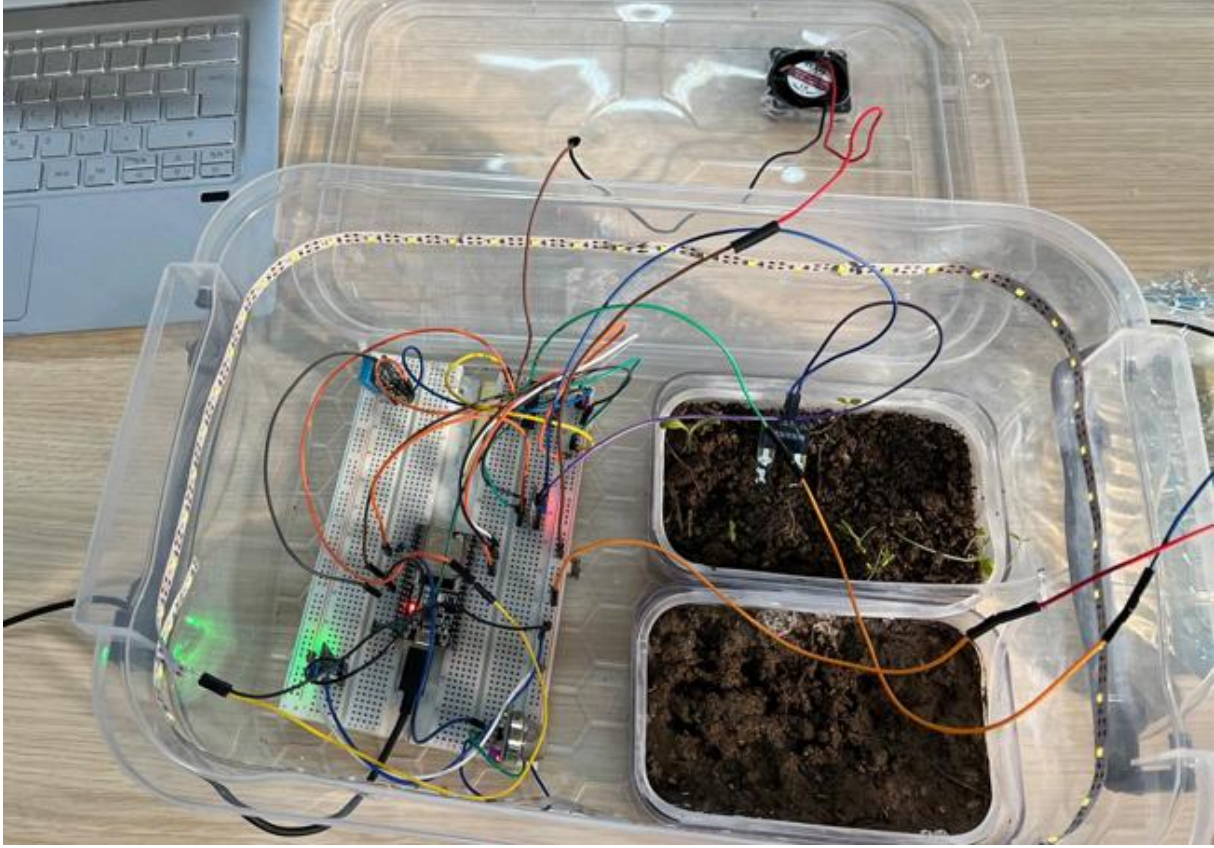
### 1. Mobil Uygulama Entegrasyonu(SmartAgno):

- Java kullanılarak Android Studio 'da yazılmıştır.
- Uygulama Ana ekranında 4 adet Custom Progress Bar bulunmaktadır.Bu barlara Websocket sunucusu üzerinden anlık sensör verileri gelmektedir(Sıcaklık,Nem,Toprak Nemi,Hava Kalitesi)
- Ayrıca ölçüm aralığını kendimiz değiştirebiliyoruz şuan varsayılan olarak 1 sn de bir lçm yapılıyor.
- WebSocket API si Aws üzerinde 3001 portundan uygulama ile iletişim kuruyor.
- ESP32 ile uygulama arasında çift yönlü bir iletişim vardır.ESP32 anlık olarak sensör verilerini telefona iletirken telefonda ise ESP32 ye eşik değerlerin değişimi ve ölçüm süresi verileri gönderilmektedir.



-

#### 4. Proje Görseli



#### 5. Beklenen Sonuçlar ve Sistem Çalışma Tahminleri

- **Doğruluk ve Güvenlik:** Akıllı iklimlendirme sisteminde, sensörlerden alınan verilerin doğru şekilde işlenmesi beklenmektedir. Nem, sıcaklık, toprak nemi ve ışık gibi çevresel parametrelerin doğru ölçülmesi ve kullanıcılara güvenilir bilgiler sunulması hedeflenmektedir.
- **Otomasyon:** Tüm çevresel koşullar (nem, sıcaklık, toprak nemi, ışık) otomatik olarak algılanacak ve ilgili sistemler (sulama motoru, havalandırma fanı, aydınlatma) gerektiğinde aktif hale getirilecektir. Sistemdeki tüm önemli verilerin (sensör verileri, sistem hareketleri vb.) kaydedilmesi sağlanacaktır. Sistem, her parametreyi izleyip yönetebileceği şekilde tasarlanmış olup, sadece gerekli durumlarda müdahaleye gerek duyulacaktır.
- **Kolay Erişim:** Mobil uygulama üzerinden sera içindeki koşulların izlenmesi ve sistemin kontrol edilmesi kolayca yapılabilecektir. Kullanıcılar ortamın sıcaklık ve nem seviyelerini izleyebilir.
- **Esneklik ve Ölçeklenebilirlik:** AWS altyapısı sayesinde büyük veri tabanı işlemlerine ve daha fazla kullanıcı eklenmesine uygun bir yapı.

#### 6. Seçim Gerekçeleri

**Kullanılan Teknolojilerin Seçim Nedenleri:**

- **ESP32:** Düşük maliyetli, enerji verimli ve Wi-Fi/Bluetooth özelliklerine sahip bir mikrodenetleyici platformu olarak, sistemimizin merkezi kontrol ünitesidir.
- **AWS Hizmetleri:** Belirli bir süre ücretsiz sunulan hizmetleri sayesinde, sensörlerden gelen verilerin işlenmesi ve saklanması sırasında hem ESP32 hem de mobil uygulama tarafında işlem yükünü önemli ölçüde azalttık.
- **API Mimarisi:** Veritabanı işlemlerini ESP32 üzerinde çalıştırmak yerine, sunucu tabanlı API mimarisi kullandık. Bu sayede hem işlem yükünü azalttık hem de ileride ihtiyaç duyulursa farklı platformların sisteme kolayca entegre olmasını sağladık.
- **WebSocket:** Sensörlerden gelen verilerin mobil uygulama ile gerçek zamanlı ve düşük gecikmeli bir şekilde paylaşılmasını sağladık.
- **Figma:** Kullanıcı arayüzü tasarımlarını hızlı ve kolay bir şekilde oluşturduk ve Android Studio'ya sorunsuz bir şekilde entegre ettik.

