**Proje Adı:** SmartIoT

**Kısa Tanım**

SmartIoT, kullanıcıların IoT cihazlarına bağlanmalarını, yönetmelerini ve verilerini izlemelerini sağlayan bir web ve mobil uygulama. Kullanıcılar, ESP8266 tabanlı IoT cihazlarına WiFi üzerinden bağlanarak cihazlarının MQTT broker ile haberleşmesini ve anlık verilerini izleyip komutlar göndermeyi kolayca gerçekleştirebilirler.

**Projenin Amaç ve Hedefleri**

1. **Kullanıcı Dostu Bir Arayüz Sağlamak**: Projenin temel amacı, kullanıcıların IoT cihazlarına kolayca bağlanabilmelerini ve bu cihazları yönetebilmelerini sağlamaktır. SmartIoT, sezgisel bir kullanıcı arayüzü sunarak kullanıcı deneyimini olabildiğince basit ve anlaşılır hale getirecektir.
2. **IoT Cihazlarını Uzaktan Yönetmek**: SmartIoT, kullanıcılara IoT cihazlarına uzaktan erişim ve yönetim imkanı sağlar. Bu sayede kullanıcılar, herhangi bir fiziksel konumdan bağımsız olarak cihazlarını takip edebilir, verilerini izleyebilir ve gerektiğinde komutlar gönderebilir.
3. **Anlık Veri Takibi ve Analiz**: Kullanıcılar, SmartIoT uygulaması üzerinden IoT cihazlarının anlık verilerini izleyebilecek ve bu verileri grafikler ve analizlerle görselleştirerek daha iyi anlayabilecektir.
4. **Kolay Kurulum ve Konfigürasyon**: SmartIoT, IoT cihazlarının kurulum sürecini basitleştirir ve kullanıcıların cihazları hızlı bir şekilde WiFi ağına bağlamalarını sağlar. Ayrıca MQTT broker ve konu (topic) ayarlarını yapma sürecini kullanıcılar için kullanıcı dostu bir şekilde sunar.
5. **Uyarı ve Bildirimler**: SmartIoT, kullanıcılara cihazlarında meydana gelen önemli olaylar veya belirlenen koşullar gerçekleştiğinde uyarılar ve bildirimler gönderecektir. Böylece kullanıcılar, kritik durumları hızlıca fark edebilecek ve gereken eylemleri alabilecektir.
6. **Geliştirilebilirlik ve Genişletilebilirlik**: Proje, geliştirilebilir ve genişletilebilir bir yapıya sahip olacak şekilde tasarlanacaktır. Yeni IoT cihazları ve özelliklerin kolayca entegre edilebilmesi için uygun bir altyapı sağlanacaktır.
7. **Performans ve Verimlilik**: Proje, IoT cihazları ile uygulama arasındaki iletişimi hızlı ve verimli bir şekilde sağlayacak, böylece düşük gecikme süreleri ve yüksek performans elde edilecektir.

SmartIoT'un bu amaçlar ve hedefler doğrultusunda, kullanıcıların IoT cihazlarını etkili bir şekilde yönetmelerini ve verileri anlık olarak izlemelerini mümkün kılmak için çalışaktır.

**Projenin Hedef Kitlesi ve Kullanıcı Profili:**

1. **Ev Sahipleri**: Ev sahipleri, ev otomasyonu ve IoT cihazlarına ilgi duyan ve evlerinde farklı cihazları yönetmek ve izlemek isteyen kişilerdir. Örneğin, ısıtma ve soğutma sistemleri, aydınlatma, güvenlik kameraları gibi ev içi cihazları uzaktan kontrol etmek ve verilerini takip etmek isteyebilirler.
2. **Küçük İşletmeler**: Küçük işletme sahipleri, iş yerlerindeki IoT cihazlarıyla ilgilenen ve bu cihazları kolayca yönetmek ve verilerini izlemek isteyen kişilerdir. Örneğin, iş yerindeki akıllı sensörlerin verilerini izlemek ve enerji verimliliğini artırmak isteyebilirler.
3. **Maker ve IoT Geliştiricileri**: IoT projeleriyle uğraşan maker'lar ve IoT geliştiricileri, SmartIoT'u kendi projelerindeki IoT cihazlarını yönetmek ve verilerini izlemek için kullanabilirler. Ayrıca, SmartIoT'un API belgeleri ve dokümantasyonu sayesinde, kendi projelerine entegrasyon yapmak isteyen geliştiriciler de hedef kitlenin bir parçasıdır.
4. **Ev ve İş Yeri Otomasyon Firmaları**: Ev ve iş yeri otomasyonu konusunda hizmet veren firmalar, SmartIoT'u müşterilerine sunacakları bir hizmet olarak kullanabilirler. Bu sayede müşterileri, kendi evlerini veya iş yerlerini uzaktan yönetebilir ve verilerini takip edebilirler.
5. **Teknoloji Meraklıları**: Teknolojiye ilgi duyan ve yeni teknolojilere açık olan kişiler, SmartIoT'u keşfetmek ve IoT cihazlarıyla etkileşimde bulunmak için kullanabilirler.
6. **Endüstriyel ve Tarımsal Uygulamalar**: Endüstriyel ve tarımsal sektörlerdeki kullanıcılar, SmartIoT'u endüstriyel makineleri veya tarımsal sensörleri yönetmek ve izlemek için kullanabilirler. Bu sayede uzaktan takip ve kontrol sağlayarak verimliliği artırabilirler.

SmartIoT, geniş bir hedef kitlesine hitap eden kullanıcı dostu bir uygulama olacak şekilde tasarlanacak. Bu farklı kullanıcı profillerini anlamak ve ihtiyaçlarına uygun çözümler sunmak, projenin başarısı için önemlidir.

SmartIoT projesinin Web App tarafında kullanacağı bazı teknoloji ve platformlar aşağıda sıralanmıştır:

1. **Frontend Teknolojileri**:
   * HTML: Sayfaların yapılandırılması için kullanılacak temel işaretleme dili.
   * CSS: Stil ve düzenleme için kullanılacak. Kullanıcı arayüzünün tasarımını oluşturmak için gerekli.
   * JavaScript: Web App'in interaktif ve dinamik özelliklerini eklemek için kullanılacak.
   * Bootstrap: Responsive (duyarlı) ve modern kullanıcı arayüzleri oluşturmak için popüler bir CSS framework'ü.
2. **Sunucu ve API Teknolojileri**:
   * Node.js: Server-side JavaScript çalıştırmak için kullanılacak. Node.js, özellikle WebSocket ve HTTP bağlantıları sağlamak için kullanılacak.
   * Express.js: Node.js üzerinde çalışan hafif bir web uygulama çerçevesi. API'leri oluşturmak için kullanılacak.
   * MQTT Kitaplığı: MQTT protokolünü kullanarak IoT cihazları ile haberleşmek için Node.js’de bulunan MQTT kütüphanesi kullanılacak .
3. **Veritabanı Teknolojileri**:
   * MySQL: Yapılandırılmış veri saklamak ve ilişkisel veritabanı yönetimi için kullanılacak. Kullanıcı, Iot cihaz bilgilerini ve sensörlerden gelen bilgileri depolamak için kullanılacak.
4. **Grafik ve Veri Görselleştirme Kütüphaneleri**:
   * Chart.js: Grafikler oluşturmak ve IoT cihazlarının verilerini görselleştirmek için kullanılabilir.
   * D3.js: Gelişmiş grafikler ve veri görselleştirmeleri için kullanılabilir.
5. **Diğer Araçlar ve Teknolojiler**:
   * SmartConfig: ESP8266 Iot cihazlarının WiFi ağına bağlanmasını kolaylaştıran bir araçtır. Uygulama içinde WiFi bilgilerini cihaza gönderirken kullanılabilir.
   * MQTT Broker: MQTT protokolüne dayalı veri akışı sağlamak için uygun bir public MQTT broker kullanılacak.(HiveMQ, Mossquito, EMQX vs.)

Bu teknolojiler, SmartIoT projesinin Web App tarafında kullanılabilecek temel bileşenlerdir.

**Mimari**

SmartIoT uygulamasının genel mimarisi, kullanıcıların IoT cihazlarına bağlanmalarını, yönetmelerini ve verilerini izlemelerini sağlamak için tasarlanmıştır. Aşağıda uygulamanın genel mimarisi ve bileşenleri açıklanmaktadır:

1. **Kullanıcı Arayüzü (Frontend)**:

* Web veya mobil cihazlardan erişilebilen kullanıcı arayüzüdür.
  + Kullanıcılar buradan IoT cihazlarına bağlanabilir, verileri izleyebilir ve komutlar gönderebilir.
  + Arayüz, kullanıcıların IoT cihazlarını kolayca yönetebilmelerini sağlayacak şekilde tasarlanacaktır.

1. **Sunucu (Backend)**:
   * Kullanıcıların cihazlarına erişimini sağlamak ve veri alışverişini yönetmek için sunucu tarafı bulunur.
   * Sunucu, kullanıcıların verilerini saklamak ve işlemek için veritabanıyla iletişim kurar.
   * MQTT broker ile cihazlar arasındaki iletişimi de sunucu yönetir.
2. **Veritabanı**:
   * Kullanıcı verilerini ve cihazların konfigürasyon bilgilerini depolayan bir veritabanı kullanılır.
   * Kullanıcı bilgileri, cihaz tanımları, veri akışları ve diğer ilişkisel veriler burada saklanır.
   * Veritabanı, veri alımı ve gönderimi için sunucu ile etkileşimde bulunur.
3. **IoT Cihazları**:
   * ESP8266 gibi IoT cihazları, WiFi üzerinden ağa bağlanır ve MQTT protokolünü kullanarak sunucuyla iletişim kurar.
   * Cihazlar, belirlenen MQTT broker üzerinde yayın yapar ve konuları takip ederek komutları alır.
4. **MQTT Broker**:
   * MQTT protokolünü destekleyen bir broker, IoT cihazları ile sunucu arasında mesajlaşmayı sağlar.
   * IoT cihazları, belirlenen MQTT broker'a veri yayınlar ve komutları takip eder.
   * Sunucu, MQTT broker üzerinden IoT cihazlarına mesaj gönderir.
5. **Uyarı ve Bildirim Sistemi**:

* IoT cihazlarından alınan verileri izleyen ve belirli koşullar gerçekleştiğinde kullanıcılara bildirimler gönderen bir sistemdir.
* Kullanıcılar, önemli olaylar veya belirlenen durumlar için uyarılar alabilirler.

**İletişim Ağı**

1. **IoT Cihazları ile Cihaz İletişimi**
   1. **IoT Cihazlarının Konfigürasyonu**: IoT cihazları (örn. ESP8266), SmartConfig veya benzeri bir yöntemle WiFi ağına bağlanacaklar. Bu aşamada cihazlar, WiFi ağı için gerekli kimlik bilgilerini (SSID ve şifre) alarak bağlantı kuracaklardır.
   2. **MQTT Broker Bilgileri ve Konuların Tanımlanması**: IoT cihazları, WİFİ’ye bağlandıktan sonra kendisini Web Sunucusu olarak konumlandırır ve telefondan yada bigisayardan HTTP isteği geldiğinde response olarak yayın yaptığı ve dinlediği konuları cihazla paylaşır.
2. **IoT Cihazları ve MQTT Broker'ı Arasındaki İletişim Akışı**:
   1. **MQTT Broker ve IoT Cihazı Arasındaki Bağlantı**: IoT cihazları, sağlanan MQTT broker bilgilerini kullanarak belirtilen broker'a bağlanırlar. Bağlantı başarılı olduğunda, cihazlar belirtilen konuları takip etmeye başlarlar ve veri alışverişi için hazır hale gelirler.
   2. **Veri Yayını**: IoT cihazları, belirtilen konular üzerinde düzenli olarak veri yayınları yaparlar. Örneğin, sıcaklık, nem, ışık düzeyi gibi sensör verileri cihaz tarafından belirtilen konularda yayınlanır.
   3. **Komut Takibi**: IoT cihazları, belirtilen konuları takip eder ve gelen komutları algılarlar. Örneğin, kullanıcı tarafından gönderilen "Aydınlatmayı Kapat" komutu gibi belirli bir konuda yayın yapıldığında, cihaz komutu algılar ve ilgili aksiyonu gerçekleştirir.
3. **Gerçek Zamanlı Veri İzleme ve WebSocket Kullanımı**:
   1. **Sunucu Tarafında Veri İzleme**: Sunucu, IoT cihazlarından gelen verileri MQTT broker üzerinden düzenli olarak izler. Bu veriler, belirlenen konularda yayınlanır ve güncellenir.
   2. **Sunucu Tarafında WebSocket Bağlantısı Oluşturma**: Kullanıcı, SmartIoT Web veya Mobil uygulaması aracılığıyla sunucuyla WebSocket üzerinden bağlantı oluşturur. WebSocket bağlantısı, gerçek zamanlı veri alışverişini sağlamak için kullanılır.
   3. **Sunucu Tarafında Veri İletimi**: Sunucu, IoT cihazlarından alınan güncel verileri, kullanıcıların bağlı WebSocket bağlantılarına anlık olarak iletir. Bu sayede kullanıcılar, IoT cihazlarından gelen verilere gerçek zamanlı olarak erişebilirler.
   4. **Kullanıcının Veri Görselleştirmesi**: Kullanıcıların Web veya Mobil uygulamasında WebSocket bağlantısı üzerinden alınan veriler, grafikler, tablolar veya diğer görsel öğelerle gerçek zamanlı olarak görselleştirilir. Kullanıcılar, anlık verilere hızlı ve etkileşimli bir şekilde erişebilirler.
   5. **Veri Güncellemeleri ve Push Bildirimleri**: Sunucu, yeni veri güncellemelerini belirli aralıklarla veya anlık olarak bağlı WebSocket bağlantıları üzerinden kullanıcılara iletir. Bu sayede kullanıcılar, IoT cihazlarından gelen güncel verilere ve bildirimlere hızlı bir şekilde tepki verebilirler.
   6. **İstemci Tarafında Komut Gönderme**: Kullanıcılar, Web veya Mobil uygulama aracılığıyla sunucuya komutlar göndererek IoT cihazlarına etkileşimde bulunabilirler. Sunucu, bu komutları alır ve ilgili IoT cihazlarına iletir.
   7. **İstemci Tarafında Komut Onayı**: IoT cihazları, alınan komutları işler ve sonuçlarını sunucuya bildirir. Sunucu, bu sonuçları ilgili WebSocket bağlantıları üzerinden kullanıcılara iletir.
   8. **Gerçek Zamanlı Takip ve İnteraktif Kontrol**: Kullanıcılar, IoT cihazlarının anlık durumunu gerçek zamanlı olarak izleyebilir ve komut göndererek cihazları etkileyebilir. Sunucu aracılığıyla gerçek zamanlı veri izleme ve WebSocket kullanımı, kullanıcılara interaktif bir kontrol sağlar.

metin, ekran görüntüsü, diyagram, uzaktan kumanda içeren bir resim

Açıklama otomatik olarak oluşturulduekran görüntüsü, metin, diyagram, tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldu

Bu iletişim akışları sayesinde SmartIoT uygulaması, kullanıcıların IoT cihazlarını kolayca yönetmelerini, verileri izlemelerini ve güncel durumları hakkında bilgi alabilmelerini sağlar.

**Veri Tabanı Tasarımı:**

ekran görüntüsü, metin, tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Kullanıcı Arayüzü Temelleri**

1. **Giriş Ekranı**:
   * Kullanıcının uygulamaya giriş yapacağı ana ekran.
   * Kullanıcı maili ve şifre alanları bulunur.
   * "Giriş Yap" ve "Kayıt Ol" gibi seçenekler mevcuttur.
2. **Kayıt Ekranı**:
   * Yeni kullanıcıların uygulamaya kayıt olabileceği ekran.
   * Ad, soyad, e-posta ve şifre gibi bilgileri girmek için alanlar vardır.
   * "Kayıt Ol" düğmesiyle kayıt işlemi tamamlanır.
3. **Ana Ekran**:
   * Kullanıcının tüm IoT cihazlarını görüntüleyebileceği merkezi ekran.
   * Her cihaz için kartlar bulunur ve bu kartlarda cihazın adı, durumu ve son veri güncellemesi görüntülenir.
   * Kullanıcılar, cihazlara tıklayarak daha detaylı bilgilere erişebilir.
   * Kullanıcılar, yeni Iot cihazlarını bu ekran üzerinden ekleyebilir.
4. **Cihaz Detay Ekranı**:
   * Seçilen bir IoT cihazının detaylarının görüntülendiği ekran.
   * Cihazın anlık durumu, sensör verileri ve son alınan komutlar gibi bilgiler burada gösterilir.
   * Kullanıcılar, cihazın kontrolünü bu ekrandan sağlayabilir.
   * Not: Iot cihazlarının detay ekranları cihazdan cihaza farklılık göstercek. Led kontrolü ekranında “ledi kapat”/”ledi aç“ butonları bulunurken. Sıcaklık sensör ekranında sensörden gelen anlık ve geçmiş veriler görüncek.