

**T.C.
SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ**

BSM 498 BİTİRME ÇALIŞMASI

GERÇEK ZAMANLI VERİ İZLEME

G201210597 – Bilal RASLEN

**Fakülte Anabilim Dalı : BİLGİSAYAR MÜHENDİSLİĞİ
Tez Danışmanı : Prof.Dr. CELAL ÇEKEN**

2023-2024 Bahar Dönemi

**T.C.
SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ**

GERÇEK ZAMANLI VERİ İZLEME

BSM 498 - BİTİRME ÇALIŞMASI

Bilal RASLEN

Fakülte Anabilim Dalı : BİLGİSAYAR MÜHENDİSLİĞİ

Bu tez .. / .. / ... tarihinde aşağıdaki jüri tarafından oybirliği / oyçokluğu ile kabul edilmiştir.

.....
Jüri Başkanı

.....
Üye

.....
Üye

ÖNSÖZ

Bu proje, IoT ve veri mühendisliği alanlarındaki teknolojileri bir araya getirerek çevresel verilerin toplanması, işlenmesi ve görselleştirilmesi sürecini kapsar. ESP32 mikrodenetleyici ve çeşitli sensörler kullanılarak sıcaklık, nem ve titreşim verileri toplanmakta ve MQTT ile iletilmektedir. Veriler, PostgreSQL veritabanında saklanır, Node.js sunucusu ve Kafka ile işlenir. Son olarak, Firebase ve Server Sent Events sunucusu aracılığıyla gerçek zamanlı analiz edilerek React kullanıcı arayüzünde görselleştirilir. Bu proje, IoT cihazlarının veri toplama ve analiz kapasitelerini etkili bir şekilde göstermeyi amaçlamaktadır.

Prof. Dr. Celal Çeken Hocama son iki yıldır bana rehberlik ettiği ve beni bilinçlendirdiği için teşekkür ederim. Eğitim süresince maddi desteklerinden dolayı Mahmud Dahi Abu Mhdi'ye teşekkür ederim. Babamın vefatından bu yana bana destek olan anneme teşekkür ederim. Maddi ve manevi desteklerinden dolayı kardeşlerime teşekkür ederim. Eğitim yolculuğumu tamamlamamı beklediği için nişanlım Baraa Ali'ye teşekkür ederim. Bana en iyi üniversitelerinde öğrenim görme fırsatı verdiği için Türkiye'ye teşekkür ederim. Gurbet gününden bu yana her geçen gün daha fazla özlediğim Suriye'ye teşekkür ederim. Yıllarca süren eğitim süresince baskılara katlandığım için kendime teşekkür ederim.

İÇİNDEKİLER

ÖNSÖZ.....	iii
İÇİNDEKİLER.....	iv
SİMGELER VE KISALTMALAR LİSTESİ.....	vi
ŞEKİLLER LİSTESİ.....	vii
TABLolar LİSTESİ.....	viii
ÖZET.....	ix

SİMGELER VE KISALTMALAR LİSTESİ	vi
---------------------------------------	----

BÖLÜM 1.	GİRİŞ.....	1
1.1.	Projenin Amacı	1
1.2.	Kapsam ve Hedefler	2
BÖLÜM 2.	SİSTEMATİK YAKLAŞIM	3
2.1.	Formülasyon	3
2.1.1.	Veri Toplama ve İletim :	4
2.1.2.	Veri Depolama :	4
2.1.3.	Veri İşleme :	4
2.1.4.	Veri Aktarımı :	4
2.1.5.	Görselleştirme:.....	4
2.2.	Donanım Mimarisi.....	4
2.2.1.	Donanım Mimarisi.....	4
2.3.	Yazılım Mimarisi	5
2.3.1.	Veri Toplama ve İletim	5
2.3.2.	Veri Saklama.....	5
2.3.3.	Veri İşleme	6
2.3.4.	Veri Aktarımı	6
2.3.5.	Görselleştirme ve Kullanıcı Arayüzü	6
2.4.	Bulut Mimarisi.....	7

BÖLÜM 3.	DENEY DÜZENEGİ VE SANAL LABORATUVAR	
8		
3.1.	Deney Düzenegi	8
3.2.	Sanal Laboratuvar Karakteristiği	8
3.3.	Test Aşaması	9
BÖLÜM 4.	VERİ GÜVENLİĞİ DEĞERLENDİRMESİ	10
4.1.	Veri Bozulması ve Elektriksel Parametrelerin İlişkisi	
	10	
4.2.	Elektrik Dalgalanmasına Bağlı Güvenlik Kontrolünün	
	Haberleşme Kalitesine Etkileri	10
4.3.	Mobil Bozulma ve Veri Açığı İlişkisi	10
BÖLÜM 5.	SONUÇLAR VE ÖNERİLER.....	11
5.1.	Sonuçlar	11
5.2.	Öneriler	12
5.2.1.	Ek Sensörler ve Veri Kaynakları	12
5.2.2.	Veri Analitiği ve Yapay Zeka	13
5.2.3.	Mobil Uygulama Geliştirme	13
5.2.4.	Güvenlik Önlemleri	13
5.2.5.	Otomatik Uyarı ve Bildirim Sistemleri	13
5.2.6.	Enerji Verimliliği ve Güç Yönetimi	13
5.3.	Genel Değerlendirme	13
KAYNAKLAR.....		14
ÖZGEÇMİŞ.....		16
BSM 498 BİTİRME ÇALIŞMASI DEĞERLENDİRME VE SÖZLÜ SINAV		
TUTANAĞI.....		17

SİMGELER VE KISALTMALAR LİSTESİ

MQTT	: MQ Telemetry Transport
js	: JavaScript
FCM	: Firebase Cloud Messaging
npm	: Node JS Package Management
IoT	: Internet of Things
DB	: Database
SQL	: Structured Query Language
SSE	: Server Sent Events

ŞEKİLLER LİSTESİ

Şekil 1 Real Time Data Monitoring and Notification System	3
Şekil 2 ESP32 mikrodnetleyici DHT11 ve titreşim sensörleri.....	5
Şekil 3 Veri Toplama ve İletim ESP32 ile MQTT.....	5
Şekil 4 Veri Saklama MQTT ile PostgreSQL Veritabanı	6
Şekil 5 Veri İşleme Nodejs ile Kafka.....	6
Şekil 6 Veri Aktarımı Kafka ile Server Sernt Events (SSE).....	6
Şekil 7 Görselleştirme ve Kullanıcı Arayüzü Nodejs ile Reactjs	7
Şekil 8 Bulut Mimarisi Nodejs ile Firebase Cloud Messaging (FCM) ile Reactjs	7
Şekil 9 Veritabanındaki veriler örneği ve tipleri.....	8
Şekil 10 Mockup React Uygulaması Bilgisayarda.....	11
Şekil 11 Mouckup React Uygulamsı Telefonda	12

ÖZET

Anahtar kelimeler: IoT, ESP32, MQTT, PostgreSQL, Kafka, Firebase Cloud Messaging, React, Büyük Veri, Çevresel Parametreler[1]

Bu proje, IoT ve büyük veri teknolojilerini entegre ederek çevresel verilerin toplanması, işlenmesi ve görselleştirilmesi sürecini kapsamaktadır. Donanım bölümünde, ESP32 mikrodenetleyici ve çeşitli sensörler kullanılarak sıcaklık, nem ve titreşim verileri toplanmakta ve MQTT protokolü aracılığıyla Mosquitto Broker'a iletilmektedir. Yazılım bölümünde ise, bu veriler PostgreSQL veritabanına kaydedilmekte ve Node.js sunucusu üzerinden Kafka'ya aktarılmaktadır. Kafka, verileri gerçek zamanlı olarak Firebase ve Server Sent Events sunucusuna iletmektedir. Bu veriler, son olarak React tabanlı bir kullanıcı arayüzünde görselleştirilerek kullanıcılara sunulmaktadır.

Projenin amacı, çevresel verilerin gerçek zamanlı olarak toplanması, işlenmesi ve kullanıcı dostu bir arayüzde görselleştirilmesidir. MQTT'nin hafif ve etkili veri iletim yetenekleri, PostgreSQL'in güçlü veri saklama çözümleri ve Kafka'nın yüksek hızlı veri işleme kabiliyetleri bu projede etkin bir şekilde kullanılmaktadır. Firebase ve Server Sent Events sunucusu, gerçek zamanlı veri işleme ve depolama görevlerini yerine getirmektedir.

Bu proje, IoT cihazlarının veri toplama kapasitelerini ve farklı teknolojilerin entegrasyonunu etkili bir şekilde göstermeyi amaçlamaktadır. IoT ve veri mühendisliği alanlarında sağlam bir temel oluşturarak, bu teknolojilerin daha da ileriye taşınmasına katkıda bulunmayı hedeflemektedir.

BÖLÜM 1. GİRİŞ

Bu proje, IoT ve büyük veri teknolojilerinin entegrasyonunu sağlayarak çevresel verilerin toplanması, işlenmesi ve görselleştirilmesi üzerine odaklanmaktadır. Sıcaklık, nem ve titreşim gibi parametreler ESP32 mikrodenetleyici ve çeşitli sensörler kullanılarak toplanmakta, MQTT aracılığıyla iletilmekte ve PostgreSQL veritabanında saklanmaktadır. Node.js, Kafka, Firebase ve React kullanılarak verilerin gerçek zamanlı olarak işlenmesi ve kullanıcı dostu bir arayüzde sunulması sağlanmaktadır.

1.1. Projenin Amacı

Bu projenin amacı, IoT ve büyük veri teknolojilerini kullanarak çevresel verilerin toplanması, işlenmesi ve görselleştirilmesini sağlayan entegre bir sistem geliştirmektir.

Proje kapsamında sıcaklık, nem ve titreşim gibi çevresel parametrelerin gerçek zamanlı olarak izlenmesi, bu verilerin güvenli ve etkili bir şekilde toplanması ve saklanması, ardından analiz edilerek anlamlı bilgilere dönüştürülmesi hedeflenmektedir. ESP32 mikrodenetleyici ve çeşitli sensörler kullanılarak toplanan veriler, MQTT protokolü aracılığıyla iletilmekte ve PostgreSQL veritabanında saklanmaktadır. Node.js sunucusu ve Kafka kullanılarak veriler işlenmekte ve gerçek zamanlı olarak Firebase ve Server Sent Events sunucusuna aktarılmaktadır. Bu sistem, kullanıcıların React tabanlı bir arayüz üzerinden verilere erişimini ve analiz yapabilmelerini sağlamaktadır.

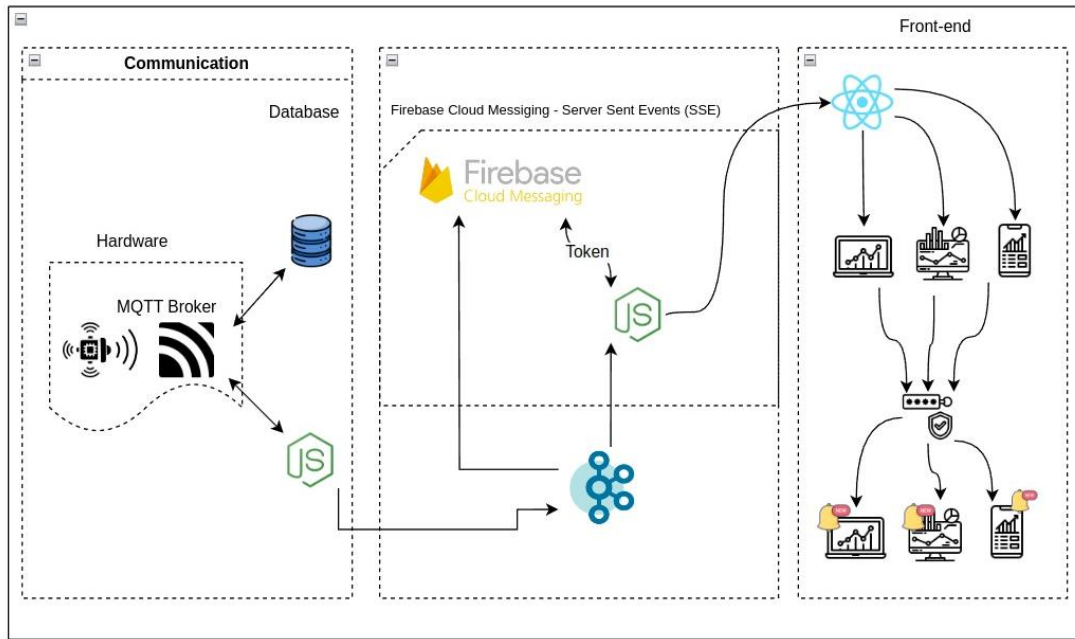
Projenin nihai amacı, IoT ve veri mühendisliği alanlarında bilgi ve deneyim kazanarak, bu teknolojilerin entegrasyonunu ve uygulamalarını göstermektir.

1.2. Kapsam ve Hedefler

Çevresel verilerin toplanması, iletilmesi, saklanması ve işlenmesi için kapsamlı bir sistem geliştiriyoruz. IoT cihazlarıyla toplanan veriler MQTT protokolüyle iletiliyor, PostgreSQL veritabanında depolanıyor ve Node.js ve Kafka kullanılarak işleniyor. Bu veriler daha sonra Firebase ve Server Sent Events sunucusuna aktarılıyor ve React kullanıcı arayüzünde görselleştiriliyor. Projenin hedefleri arasında veri toplama süreçlerinin otomatikleştirilmesi, gerçek zamanlı veri işleme yeteneklerinin sağlanması ve kullanıcıların verilere erişimini kolaylaştırmak yer alıyor. Ayrıca, makinaların arızasının önceden tespit edilmesini de sağlıyoruz.

BÖLÜM 2. SİSTEMATİK YAKLAŞIM

Bu bölümde, projenin sistematığı ve kullanılan yaklaşım detaylı olarak incelenmektedir. Proje, çevresel verilerin toplanması, iletilmesi, saklanması, işlenmesi ve görselleştirilmesi süreçlerini kapsamaktadır. Her adımda, belirlenen hedeflere ulaşmak için dikkatle planlanmış bir yaklaşım benimsenmiştir. Donanım ve yazılım bileşenleri arasındaki entegrasyon sağlanarak, verimli bir veri akışı sağlanmıştır. Proje sürecinde sistemli bir şekilde ilerlenerek, hedeflerin başarıyla gerçekleştirilmesi hedeflenmiştir.



Şekil 1 Real Time Data Monitoring and Notification System

2.1. Formülasyon

Projemizin temel amacı, çevresel verilerin IoT cihazları tarafından toplanması ve işlenmesini içerir. Bu bağlamda, aşağıdaki bileşenlerin entegrasyonunu hedefliyoruz:

2.1.1. Veri Toplama ve İletim :

IoT cihazları tarafından toplanan veriler, MQTT protokolü aracılığıyla iletilir.

2.1.2. Veri Depolama :

İletilen veriler PostgreSQL veritabanında saklanır.

2.1.3. Veri İşleme :

Node.js ve Kafka kullanılarak veriler işlenir ve analiz edilir.

2.1.4. Veri Aktarımı :

İşlenen veriler Firebase Cloud Messaging ve Server Sent Events sunucusuna aktarılır.

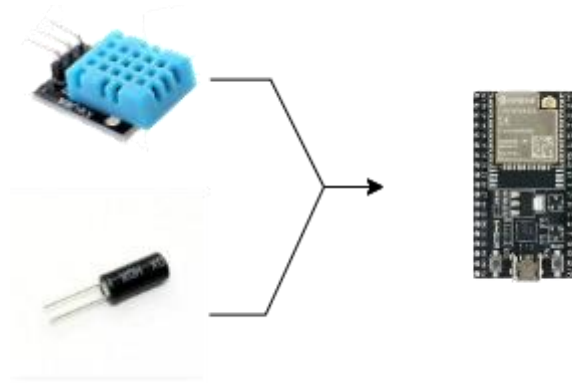
2.1.5. Görselleştirme:

Bu bileşenlerin entegrasyonu, projenin hedeflediği otomatik veri toplama, gerçek zamanlı veri işleme ve kullanıcı erişim kolaylığı sağlanacaktır. Ayrıca, makinaların arızasının önceden tespit edilmesini de amaçlıyoruz.

2.2. Donanım Mimarisi

2.2.1. Donanım Mimarisi

ESP32 mikrodnetleyici, çevresel koşulları izlemek için DHT11 sıcaklık ve nem sensörleri ile bir titreşim sensörünü (vibrasyon sensörü) bağlar. DHT11 sensörleri, ortam sıcaklığı ve nemini ölçerken, titreşim sensörü çevredeki titreşimleri algılar. ESP32, bu sensörlerden aldığı verileri işler ve MQTT protokolü kullanarak bir MQTT brokerına yayınlar. MQTT brokerı, gelen verileri alır ve bir veritabanına kaydederek saklar. Bu veritabanı, çevresel verilerin uzun süreli depolanması ve analiz için kullanılır[2].



Şekil 2 ESP32 mikrodeneleyici DHT11 ve titreşim sensörleri

Böylece, ESP32 tarafından toplanan veriler güvenli bir şekilde kaydedilir ve gerektiğinde erişilebilir durumda olur. Bu sistem, çevresel koşulların sürekli olarak izlenmesini ve verilerin uzaktan erişimle depolanmasını sağlayarak, çeşitli uygulamalarda kullanılabilir, örneğin tarım, hava durumu izleme veya endüstriyel otomasyon gibi alanlarda.

2.3. Yazılım Mimarisi

Projemizin yazılım mimarisi, çevresel verilerin toplanması, iletilmesi, saklanması, işlenmesi ve görselleştirilmesini içeren bir dizi yazılım bileşeninden oluşur. Aşağıdaki alt başlıklar, projenin yazılım mimarisini açıklar:

2.3.1. Veri Toplama ve İletim

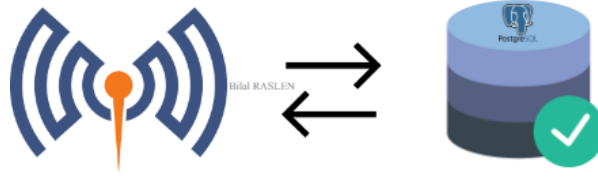
Bu bileşen, çevresel verilerin IoT cihazları tarafından toplanmasını ve MQTT protokolüyle iletilmesini sağlar.[3]



Şekil 3 Veri Toplama ve İletim ESP32 ile MQTT

2.3.2. Veri Saklama

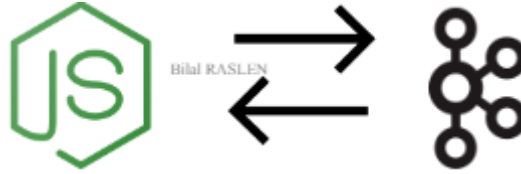
Alınan verilerin depolandığı PostgreSQL veritabanını içerir. Veritabanı, verilerin güvenli ve düzenli bir şekilde saklanmasını sağlar.[4]



Şekil 4 Veri Saklama MQTT ile PostgreSQL Veritabanı

2.3.3. Veri İşleme

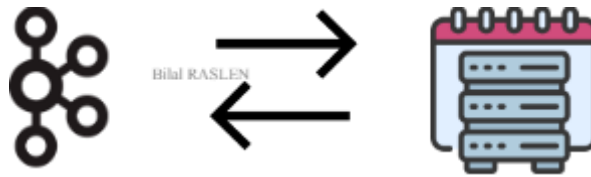
Node.js ve Kafka kullanılarak verilerin işlendiği ve analiz edildiği bileşen. Bu bileşen, verilerin gerçek zamanlı olarak işlenmesini ve anlamlı bilgilere dönüştürülmesini sağlar.[5]



Şekil 5 Veri İşleme Nodejs ile Kafka

2.3.4. Veri Aktarımı

İşlenen verilerin Firebase Cloud Messaging ve Server Sent Events Sourcing sunucusuna aktarıldığı bileşen. Bu adım, verilerin farklı platformlar arasında etkili bir şekilde aktarılmasını sağlar.[6]



Şekil 6 Veri Aktarımı Kafka ile Server Sernt Events (SSE)

2.3.5. Görselleştirme ve Kullanıcı Arayüzü

React kullanıcı arayüzünde verilerin görselleştirilmesini sağlayan bileşen. Kullanıcılar, bu arayüz aracılığıyla çevresel verilere erişebilir ve bunları anlamlı grafikler veya tablolar halinde görüntüleyebilir.[7]

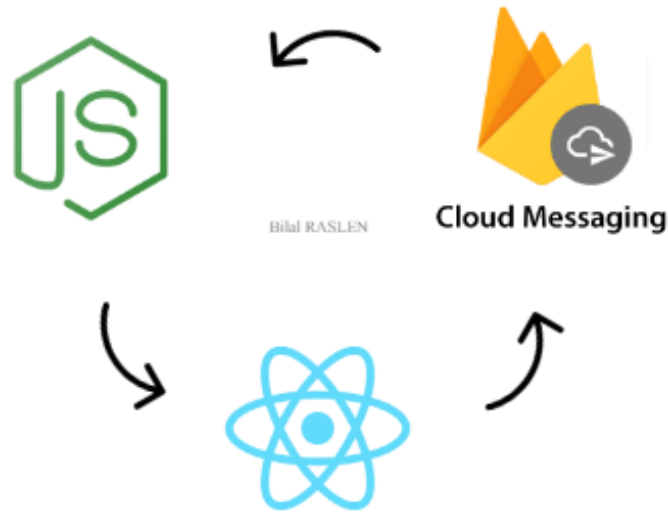
Bu alt başlıklar, projenin yazılım bileşenlerini ve bunların işlevlerini açıklar. Her bir bileşen, projenin bütününe katkıda bulunarak, çevresel verilerin etkili bir şekilde işlenmesini ve kullanılmasını sağlar.[8]



Şekil 7 Görselleştirme ve Kullanıcı Arayüzü Nodejs ile Reactjs

2.4. Bulut Mimarisi

Projemizin bulut mimarisi içerisinde Firebase Cloud Messaging (FCM), çevresel verilerin gerçek zamanlı olarak iletilmesini sağlamak için kullanılmaktadır. FCM, Android, iOS ve web uygulamalarına anlık bildirimler göndermek için kullanılan bir bulut tabanlı mesajlaşma çözümüdür. Bu sayede, projemizdeki kullanıcılara veri güncellemeleri ve uyarıları anında iletilir. [9]



Şekil 8 Bulut Mimarisi Nodejs ile Firebase Cloud Messaging (FCM) ile Reactjs

FCM aynı zamanda kullanıcı etkileşimlerine ve geri bildirimlere yanıt verebilir, böylece kullanıcı deneyimini iyileştirir. Bu bulut tabanlı çözüm, projemizin geniş kapsamlı ve gerçek zamanlı iletişim ihtiyaçlarını karşılamak için ideal bir seçenektir.

BÖLÜM 3. DENEY DÜZENEGİ VE SANAL LABORATUVAR

3.1. Deney Düzenegi

Deney düzenegi, ESP32 mikrodnetleyici, DHT11 sıcaklık ve nem sensörü ile titreşim sensörünün bağıladığı bir sistemden oluşmaktadır. ESP32, sensörlerden alınan verileri işleyerek MQTT protokolü kullanarak bir MQTT brokerına iletmektedir. Broker, gelen verileri bir veritabanına kaydederek depolamaktadır. Bu düzenek, çevresel parametrelerin gerçek zamanlı olarak izlenmesini ve verilerin uzaktan erişimle saklanması sağlar.

	id	topic	payload	detection_time
156	3590	publish/signal/push	{ "year": "1...	2024-04-19 16:40:44.000000
157	3589	publish/signal/push	{ "city": "D...	2024-04-19 16:40:44.000000
158	3580	publish/signal/ir	{ "city": "D...	2024-04-19 16:40:44.000000
159	3583	publish/signal/speed	{ "city": "T...	2024-04-19 16:40:44.000000
160	3581	publish/signal/show	{ "city": "T...	2024-04-19 16:40:44.000000
161	3582	publish/signal/slow	{ "city": "T...	2024-04-19 16:40:44.000000
162	3587	publish/signal/push	{ "city": "T...	2024-04-19 16:40:44.000000
163	3577	publish/signal/of	{ "city": "T...	2024-04-19 16:40:44.000000
164	3585	publish/signal/go	{ "id": "383...	2024-04-19 16:40:44.000000
165	3744	publish/signal/air	{ "id": "383...	2024-04-19 16:40:44.000000
166	3743	publish/signal/air	{ "id": "383...	2024-04-19 16:40:44.000000
167	3584	publish/signal/run	{ "id": "383...	2024-04-19 16:40:44.000000
168	3588	publish/signal/push	{ "id": "383...	2024-04-19 16:40:44.000000
169	3578	publish/signal/temp	{ "id": "383...	2024-04-19 16:40:44.000000

Şekil 9 Veritabanındaki veriler örneği ve tipleri

3.2. Sanal Laboratuvar Karakteristiğı

Sanal laboratuvar, gerçek dünyadaki deney düzeneginin simülasyonunu içerir. Bilgisayar tabanlı bir simülasyon ortamı kullanılarak, ESP32 mikrodnetleyici ve sensörlerin etkileşimi sanal olarak modellemiş ve uygulamıştır. Bu sayede, fiziksel bir laboratuvar ortamına ihtiyaç duymadan, deneyler sanal olarak gerçekleştirilebilir ve sonuçlar analiz edilebilir.

3.3. Test Aşaması

Test aşamasında, deney düzeneği ve sanal laboratuvar kullanılarak sistemin performansı değerlendirilmiştir. Sensörlerin doğruluğu, veri iletiminin güvenilirliği ve veritabanı kayıt süreçleri test edilmiştir. Ayrıca, sistemde olası hata durumları ve güvenlik önlemleri de göz önünde bulundurulmuştur. Test sonuçları, sistemin istikrarlı çalışması ve beklenen performansı göstermesiyle olumlu sonuçlanmıştır.

BÖLÜM 4. VERİ GÜVENLİĞİ DEĞERLENDİRMESİ

Veri güvenliği, projenin önemli bir unsuru olarak ele alınmış ve detaylı bir şekilde incelenmiştir. Sistemde kullanılan verilerin gizliliği ve bütünlüğü, güvenlik protokolleri ve algoritmalarıyla korunmuştur. Ayrıca, veri erişim yetkilendirmeleri ve izleme mekanizmalarıyla veriye sadece yetkili kullanıcıların erişimine izin verilmiştir.

4.1. Veri Bozulması ve Elektriksel Parametrelerin İlişkisi

Elektriksel dalgalanmaların, veri iletiminde bozulmalara neden olabileceği göz önünde bulundurulmuştur. Sistemdeki elektriksel parametrelerin stabilitesi ve güç kaynağının sağlamlığı, veri iletimi üzerinde önemli bir etkiye sahiptir. Bu nedenle, elektriksel parametrelerin sürekli olarak izlenmesi ve kontrol edilmesi gerekmektedir.

4.2. Elektrik Dalgalanmasına Bağlı Güvenlik Kontrolünün Haberleşme Kalitesine Etkileri

Elektrik dalgalanmalarına bağlı güvenlik kontrolleri, iletişim kalitesini etkileyebilir. WH etkisi ve güvenlik faktöründeki bozulmalar, iletişim hızını ve güvenilirliğini etkileyebilir. Bu nedenle, güvenlik kontrollerinin dengeli bir şekilde yapılandırılması ve güvenlik faktörünün optimize edilmesi önemlidir. Ayrıca, veri hızının bulut teknolojisine etkisi de göz önünde bulundurularak, iletişim altyapısının verimliliği değerlendirilmiştir.

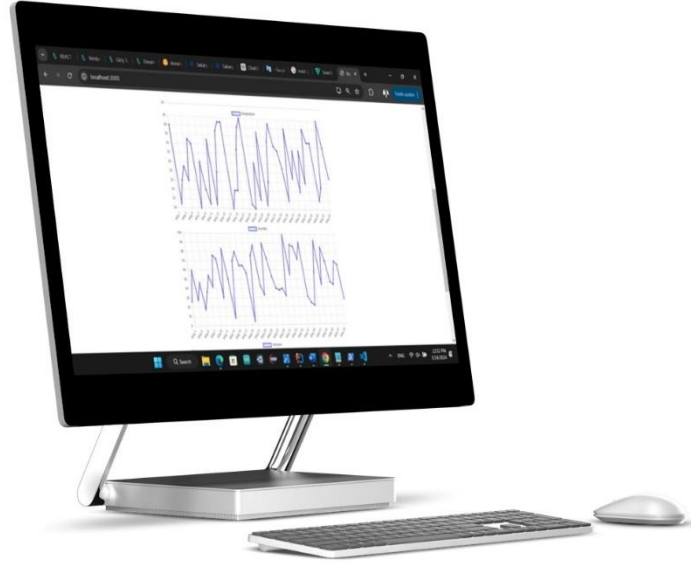
4.3. Mobil Bozulma ve Veri Açığı İlişkisi

Mobil bozulmaların, veri güvenliği açıklarına neden olabileceği ve veri bütünlüğünü tehlikeye atabileceği göz önünde bulundurulmuştur. Bu nedenle, mobil cihazlarla yapılan veri iletimi güvenlik açısından titizlikle incelenmiş ve uygun önlemler alınmıştır. Veri açıkları ve güvenlik zaafiyetlerine karşı etkin bir şekilde koruma sağlanmıştır.

BÖLÜM 5. SONUÇLAR VE ÖNERİLER

5.1. Sonuçlar

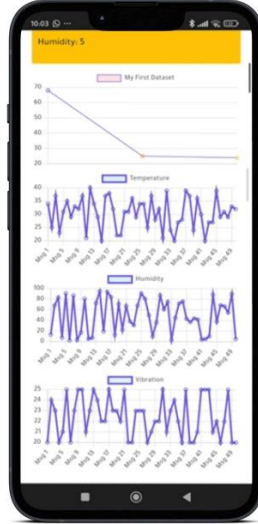
Bu projenin amacı, IoT ve büyük veri teknolojilerini kullanarak çevresel verilerin toplanması, işlenmesi ve görselleştirilmesi için entegre bir sistem geliştirmektir. ESP32 mikrodenetleyici, DHT11 sıcaklık ve nem sensörü ile titreşim sensörü kullanılarak veri toplama sağlanmıştır. MQTT protokolü aracılığıyla veriler, bir MQTT brokerına iletilmiş ve bu broker, verileri bir PostgreSQL veritabanına kaydetmiştir. Bu süreç, verilerin güvenli ve verimli bir şekilde toplanmasını ve saklanmasını sağlamıştır. Verilerin işlenmesi için Node.js sunucusu ve Kafka kullanılmış, bu sayede veriler gerçek zamanlı olarak Firebase ve Server Sent Events sunucusuna aktarılmıştır. Son olarak, veriler React tabanlı bir kullanıcı arayüzünde görselleştirilerek kullanıcılara sunulmuştur.



Şekil 10 Mockup React Uygulaması Bilgisayarda

Proje süresince, farklı teknolojilerin entegrasyonu başarılı bir şekilde gerçekleştirilmiştir. ESP32'nin sensörlerden veri toplama ve bu verileri MQTT ile yayınlama kapasitesi doğrulanmıştır. MQTT brokerının, gelen verileri güvenli bir

şekilde alıp PostgreSQL veritabanına kaydetme performansı test edilmiştir. Node.js ve Kafka, verilerin hızlı ve etkili bir şekilde işlenmesini sağlamış, Firebase ve Server Sent Events sunucusu ise verilerin gerçek zamanlı olarak izlenmesini mümkün kılmıştır.



Şekil 11 Mouckup React Uygulamsı Telefonda

Kullanıcı arayüzü olarak React kullanılması, kullanıcıların verileri anlamlı grafikler ve görseller aracılığıyla kolayca takip edebilmesini sağlamıştır. Bu görseller, çevresel verilerin daha iyi anlaşılmasına ve analiz edilmesine yardımcı olmuştur. Ayrıca, sistemin modüler yapısı, gelecekte ek sensörler ve yeni veri kaynaklarının kolayca entegre edilmesine olanak tanımaktadır.

5.2. Öneriler

5.2.1. Ek Sensörler ve Veri Kaynakları

Sisteme daha fazla sensör eklenerek, izlenen çevresel parametrelerin çeşitliliği artırılabilir. Örneğin, hava kalitesi sensörleri, ışık sensörleri veya toprak nem sensörleri eklenerek daha kapsamlı veri toplama sağlanabilir.

5.2.2. Veri Analitiği ve Yapay Zeka

Toplanan verilerin analiz edilmesi ve anlamlı sonuçlar elde edilmesi için veri analitiği ve yapay zeka algoritmaları entegre edilebilir. Makine öğrenimi algoritmaları, toplanan verilerden gelecekteki çevresel koşulların tahmin edilmesine yardımcı olabilir.

5.2.3. Mobil Uygulama Geliştirme

Kullanıcıların verileri sadece web arayüzünden değil, aynı zamanda mobil cihazlar üzerinden de takip edebilmesi için bir mobil uygulama geliştirilebilir. Bu, kullanıcı deneyimini artıracak ve verilerin daha kolay erişilebilir olmasını sağlayacaktır.

5.2.4. Güvenlik Önlemleri

Veri güvenliği ve gizliliği konusunda ek önlemler alınabilir. Verilerin şifrelenmesi, güvenli veri iletimi protokolleri ve kullanıcı doğrulama mekanizmaları gibi güvenlik önlemleri sistemin daha güvenli hale gelmesine yardımcı olacaktır.

5.2.5. Otomatik Uyarı ve Bildirim Sistemleri

Belirli eşik değerlerinin aşılması durumunda otomatik uyarı ve bildirim sistemleri entegre edilebilir. Örneğin, sıcaklık veya nem belirli bir seviyenin üzerine çıktığında kullanıcılara anında bildirim gönderilebilir.

5.2.6. Enerji Verimliliği ve Güç Yönetimi

Özellikle bataryalı sistemlerde, enerji verimliliği ve güç yönetimi önemli bir konudur. Enerji tasarrufu sağlayacak yöntemler ve düşük güç tüketimli bileşenler kullanılarak sistemin çalışma süresi uzatılabilir.

5.3. Genel Değerlendirme

Bu proje, IoT ve büyük veri teknolojilerinin entegrasyonunun, çevresel verilerin toplanması ve işlenmesinde nasıl kullanılabileceğini göstermiştir. ESP32 mikrodenetleyici ve sensörlerin etkin kullanımı, verilerin MQTT protokolü aracılığıyla iletilmesi ve bu verilerin PostgreSQL veritabanında saklanması başarılı bir şekilde gerçekleştirilmiştir. Node.js ve Kafka'nın kullanımı, verilerin gerçek zamanlı olarak işlenmesini sağlamış, Firebase ve Server Sent Events Sourcing sunucusu ise verilerin sürekli izlenmesine olanak tanımıştır. React tabanlı kullanıcı arayüzü,

kullanıcıların verileri anlamlı grafikler ve görseller aracılığıyla takip edebilmesini sağlamıştır.

Sonuç olarak, bu proje, çevresel verilerin toplanması, işlenmesi ve görselleştirilmesi konusunda önemli bir başarı elde etmiştir. Gelecekte yapılacak geliştirmeler ve eklemelerle, sistemin yetenekleri ve kullanım alanları daha da genişletilebilir. Bu proje, IoT ve veri mühendisliği alanlarında bilgi ve deneyim kazanmak isteyenler için değerli bir kaynak ve örnek teşkil etmektedir.

KAYNAKLAR

- [1] D. Dinculeană and X. Cheng, “Vulnerabilities and Limitations of MQTT Protocol Used between IoT Devices,” *Applied Sciences* 2019, Vol. 9, Page 848, vol. 9, no. 5, p. 848, Feb. 2019, doi: 10.3390/APP9050848.
- [2] Institute of Electrical and Electronics Engineers and IOT Asia (Conference) (2014 : Singapore), *2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP) : 21-24 April 2014, Singapore*.
- [3] “Applied Sciences | Free Full-Text | Vulnerabilities and Limitations of MQTT Protocol Used between IoT Devices.” Accessed: May 14, 2024. [Online]. Available: <https://www.mdpi.com/2076-3417/9/5/848>
- [4] L. Fröhlich, “PostgreSQL,” in *PostgreSQL*, 2022. doi: 10.3139/9783446473157.fm.
- [5] M. J. A. Baig, M. T. Iqbal, M. Jamil, and J. Khan, “Design and implementation of an open-Source IoT and blockchain-based peer-to-peer energy trading platform using ESP32-S2, Node-Red and, MQTT protocol,” *Energy Reports*, vol. 7, 2021, doi: 10.1016/j.egyr.2021.08.190.
- [6] A. Hugo, B. Morin, and K. Svantorp, “Bridging MQTT and Kafka to support C-ITS: A feasibility study,” in *Proceedings - IEEE International Conference on Mobile Data Management*, 2020. doi: 10.1109/MDM48529.2020.00080.
- [7] B. A. Jaso, N. I. Kraus, and A. S. Heller, “Identification of Careless Responding in Ecological Momentary Assessment Research: From Posthoc Analyses to Real-Time Data Monitoring,” *Psychol Methods*, vol. 27, no. 6, 2021, doi: 10.1037/met0000312.
- [8] J. S. Murthy, G. M. Siddesh, and K. G. Srinivasa, “TwitSenti: A Real-Time Twitter Sentiment Analysis and Visualization Framework,” *Journal of*

Information and Knowledge Management, vol. 18, no. 2, 2019, doi: 10.1142/S0219649219500138.

- [9] G. Albertengo, F. G. Debele, W. Hassan, and D. Stramandino, “On the performance of web services, google cloud messaging and firebase cloud messaging,” *Digital Communications and Networks*, vol. 6, no. 1, 2020, doi: 10.1016/j.dcan.2019.02.002.

ÖZGEÇMİŞ

Bilal RASLEN, 01.01.2002 de Suriye’de doğdum. İlk ve orta eğitimini Suriye’de tamamladım lise ise Türkiyede. 2019 yılında Açık Öğretim Lisesinden mezun oldum. 2020 yılında Sakarya Üniversitesi Bilgisayar Mühendisliği Bölümü’nü kazandım. 2023 yılında Sakarya Ünivesitesinde iotrlab. Donanım stajını ve 2024 yılında da iotrlab’te Yazılım stajını yaptım. SAÜ Bilgisayar Mühendisliği Bölümünden.

BSM 498 BİTİRME ÇALIŞMASI DEĞERLENDİRME VE SÖZLÜ SINAV TUTANAĞI

KONU : REAL Time Data Monitoring

ÖĞRENCİLER (Öğrenci No/AD/SOYAD): G201210597 – Bilal RASLEN

Değerlendirme Konusu	İstenenler	Not Aralığı	Not
Yazılı Çalışma			
Çalışma klavuza uygun olarak hazırlanmış mı?	x	0-5	
Teknik Yönden			
Problemin tanımı yapılmış mı?	x	0-5	
Geliştirilecek yazılımın/donanımın mimarisini içeren blok şeması (yazılımlar için veri akış şeması (dfd) da olabilir) çizilerek açıklanmış mı?			
Blok şemadaki birimler arasındaki bilgi akışına ait model/gösterim var mı?			
Yazılımın gereksinim listesi oluşturulmuş mu?			
Kullanılan/kullanılması düşünülen araçlar/teknolojiler anlatılmış mı?			
Donanımların programlanması/konfigürasyonu için yazılım gereksinimleri belirtilmiş mi?			
UML ile modelleme yapılmış mı?			
Veritabanları kullanılmış ise kavramsal model çıkarılmış mı? (Varlık ilişki modeli, noSQL kavramsal modelleri v.b.)			
Projeye yönelik iş-zaman çizelgesi çıkarılarak maliyet analizi yapılmış mı?			
Donanım bileşenlerinin maliyet analizi (prototip-adetli seri üretim vb.) çıkarılmış mı?			
Donanım için gerekli enerji analizi (minimum-uyku-aktif-maksimum) yapılmış mı?			
Grup çalışmalarında grup üyelerinin görev tanımları verilmiş mi (iş-zaman çizelgesinde belirtilebilir)?			
Sürüm denetim sistemi (Version Control System; Git, Subversion v.s.) kullanılmış mı?			
Sistemin genel testi için uygulanan metotlar ve iyileştirme süreçlerinin dökümü verilmiş mi?			
Yazılımın sızma testi yapılmış mı?			
Performans testi yapılmış mı?			
Tasarımın uygulamasında ortaya çıkan uyumsuzluklar ve aksaklıklar belirtilerek çözüm yöntemleri tartışılmış mı?			
Yapılan işlerin zorluk derecesi?	x	0-25	
Sözlü Sınav			
Yapılan sunum başarılı mı?	x	0-5	
Soruları yanıtlama yetkinliği?	x	0-20	
Devam Durumu			
Öğrenci dönem içerisindeki raporlarını düzenli olarak hazırladı mı?	x	0-5	
Diğer Maddeler			
Toplam			

DANIŞMAN (JÜRİ ADINA): CELAL ÇEKEN

DANIŞMAN İMZASI: