**Konfigürasyon ve İzleme için Android Mobil Uygulama Geliştirilmesi**

1) İş Pakedi Faaliyetlerini Listesi

* Çift Kanallı Bağlantı Mimari Tasarımı: USB OTG ve Web API üzerinden hibrit iletişim kuran Native Android SDK ve Retrofit2 tabanlı bir katman.
* USB CDC Protokol Optimizasyonu: Android USB Host API ile veri formatını otomatik parse eden, hata kontrolü eklenmiş özel bir kütüphane geliştirilmesi.
* Dinamik UI/UX Tasarımı: Masaüstü arayüzle tutarlı, ancak mobile-responsive React Native değil, Jetpack Compose ile native bileşenler (sürükle-bırak grafik konfigürasyonu).
* Offline Loglama Modülü: CSV yerine SQLCipher ile şifrelenmiş lokal veritabanı ve kullanıcı tercihine göre CSV/Parquet export seçenekleri.
* Gerçek Zamanlı Grafik Motoru: MPAndroidChart tabanlı, ancak OpenGL ES ile çizim hızlandırmalı ve veri noktası interpolasyonu özellikli özelleştirilmiş bir chart kütüphanesi.
* Güvenli API Entegrasyonu: Django REST API ile JWT tabanlı auth ve SSL Pinning (SHA-256 fingerprint) ve obfuscated API Key kullanımı.
* Çapraz Cihaz Testleri: USB OTG desteği olan 10+ farklı Android cihazda (Xiaomi, Samsung, Huawei) bağlantı stabilite testi.
* Performans Optimizasyonu: Çeşitli hızlarda (akım/gerilim) RAM'de circular buffer kullanarak veri kaybı önleme.

2) Yöntemler ve Özgün Katkılar

* Yöntemler:
	+ Delta Configuration Update: Ayar değişikliklerinde tüm veriyi değil, sadece farkı göndererek bant genişliği optimizasyonu.
	+ Smart Log Compression: CSV/Parquet’e kayıt sırasında Zstandard sıkıştırma ve metadata tagging.
* Özgün Katkılar:
	+ USB OTG Auto-Reconnect: Bağlantı kesintilerinde exponential backoff algoritması ile 5 saniyeden 5 dakikaya kadar artan aralıklarla yeniden bağlanma.
	+ Battery-Optimized Real-Time Graphing: Arka planda veri işleme için Android WorkManager ve GPU’da veri interpolasyonu ile pil tüketimini %40 azaltma.
* İncelenecek Parametreler:
	+ Örnekleme esnasında GPU kullanımı minimumda olacak.
	+ Eşzamanlı parametre değişiklikleri durumunda API’nin yanıt süresi hızlı gerçekleşecek.

3) Deney/Test/Analiz Adı

Hibrit Bağlantı Fonksiyon Testi

4) Testin Yapılma Nedeni:

Bu test, hibrit bağlantı modelinde kesintisiz veri akışı ve enerji verimliliğini doğrulamak amacıyla yapılmaktadır.

**Konfigürasyon ve İzleme için Masaüstü Uygulama Geliştirilmesi**

1) İş Pakedi Faaliyetlerini Listesi

* İhtiyaç Analizi ve Kapsam Belirleme: Cihaz konfigürasyon parametreleri , akım seçenekleri , hata logları ve gerçek zamanlı izleme gereksinimleri detaylandırılacak.
* UI/UX Tasarımı: React tabanlı, kullanıcı dostu arayüz tasarımı (Cihaz Bağlantı Durumu, Okuma/Loglama/Ayar/Durum İzleme/Grafik sayfaları).
* Backend Altyapı Geliştirme: Node.js/Express.js ile USB iletişim modülü, veri işleme ve PostgreSQL/SQLite entegrasyonu.
* Cihaz İletişim Modülü: Platformlar arası USB protokolü , akım ve parametre okuma/yazma API’ları.
* Gerçek Zamanlı Grafik Entegrasyonu: Chart.js/D3.js ile sıcaklık, akım, gerilim verilerinin dinamik görselleştirmesi.
* Veri Yönetimi ve Loglama: Ölçüm verilerinin SQLite’da lokal depolanması, hata loglarının PostgreSQL’e senkronizasyonu.
* Çapraz Platform Testleri: Electron uygulamasının Windows/Linux/macOS’ta USB bağlantı stabilitesi ve performans validasyonu.
* Kullanıcı Kabul Testleri (UAT): Fabrika/saha senaryoları ile ayar değişikliği, log okuma ve gerçek zamanlı izleme testleri.
* Dokümantasyon: Teknik kullanım kılavuzu, API dokümantasyonu ve geliştirici rehberi oluşturulması.

2) Yöntemler ve Özgün Katkılar

* Yöntemler:
	+ Modüler Mimari: Cihaz iletişimi, veri işleme ve UI katmanları birbirinden bağımsız geliştirilecek.
	+ WebSocket ile Gerçek Zamanlı İletişim: Veri gecikmesini düşürmek için özel bir WebSocket kanalı tasarlanacak.
* Özgün Katkılar:
	+ Akıllı Veri Buffering: Yüksek veri akışında SQLite’a yazma işlemini batch’leyerek disk yükü azaltılacak.
	+ USB Connection Manager: Bağlantı kopmalarında otomatik yeniden bağlanma ve son veriyi koruma özelliği eklenecek.
	+ Custom React Hooks: Cihaz durumu ve hata logları için global state yönetimi (Redux yerine Context API optimizasyonu).
* İncelenecek Parametreler:
	+ USB baud rate uyumu
	+ Grafiklerde veri örnekleme frekansı,
	+ Aynı anda 10+ parametrenin izlenmesinde CPU kullanımı.

3) Deney/Test/Analiz Adı

Çoklu Platform USB Fonksiyon ve Performans Testi

4) Testin Yapılma Nedeni:

Farklı işletim sistemlerinde USB iletişiminin sorunsuz çalıştığını ve yoğun veri aktarımı sırasında sistem performansının stabil kaldığını doğrulamak için yapılır.

**Uzaktan İzleme Sistemi**

1) İş Pakedi Faaliyetlerini Listesi

* Donanım Protokol Entegrasyonu: CAN-Ethernet Gateway/CAN-WiFi modülleri için CANbus ↔ TCP/IP çevirici middleware geliştirilmesi (SocketCAN kütüphanesi tabanlı).
* Güvenlik Altyapısı: SSL/TLS desteğine bağlı olarak hibrit şifreleme modeli.
* MQTT & RESTful API Hibrit Mimari: MQTT broker’ı (Mosquitto) ve Django REST API’nin tek bir WebSocket kanalı üzerinden entegrasyonu (Django Channels + Celery kombini).
* Dinamik Kullanıcı Arayüzü: React’te cihaz başına özelleştirilebilir dashboard.
* Zaman Serisi Veri Optimizasyonu: PostgreSQL’e TimescaleDB eklentisi ile otomatik veri partisyonlama ve arka planda veri sıkıştırma.
* Asenkron Log Arşivleme: Celery ile günlük verilerin S3/Glacier’a periyodik yedeklenmesi ve Cold Storage’a taşınan veriler için metadata indeksleme.
* Çapraz Donanım Testleri: Farklı CAN modülleri ile iletişim testi.
* Yüksek Yük Testleri: 10.000+ cihaz simülasyonu (Locust.io ile) altında API ve MQTT broker’ın stabilite analizi.

2) Yöntemler ve Özgün Katkılar

* Yöntemler:
	+ Protokol Abstraction Layer (PAL): CANbus verilerini JSON Schema’ya dönüştüren ve MQTT/REST’e uyarlayan bir ara katman.
	+ Adaptive Security Model: Cihaz TLS desteklemiyorsa, proxy’de ephemeral key exchange ile AES-GCM şifreleme.
	+ Time-Series Data Tiering: TimescaleDB’de sıcak/soğuk veri katmanları oluşturma.
* Özgün Katkılar:
	+ CANbus Message Prioritization: Kritik hata mesajlarını MQTT QoS 2 ile, diğer verileri QoS 0 ile işleme.
	+ Dynamic Topic Subscription: Kullanıcıların cihazlara özel MQTT topic’leri RegEx ile filtreleyebilmesi.
	+ Django ORM Optimizasyonu: TimescaleDB hiper tabloları için batch insert işlemlerinde COPY komutu kullanımı .
	+ React Virtualized Log Viewer: Fazla satırdaki log’u WebWorker ile render etme ve client-side’da fuzzy search desteği.
* İncelenecek Parametreler:
	+ CANbus baud rate ↔ TCP/IP gecikme ilişkisi,
	+ TLS handshake süresi ,
	+ Birden fazla cihazda WebSocket bağlantı sayısı ,
	+ TimescaleDB’de veri üzerinde sorgu performansı .

3) Deney/Test/Analiz Adı

Çoklu Protokol Entegrasyonu ve Zaman Serisi Veri Yönetimi Fonksiyon ve Performans Testi

4) Testin Yapılma Nedeni:

Farklı veri formatlarının tek API üzerinden uyumlu çalışıp, yoğun veri akışı altında sistemin stabil performans sergilediğini kontrol etmek için yapılır.