OSOP: Çoklu Veritabanı Yönetimi ve Analizi

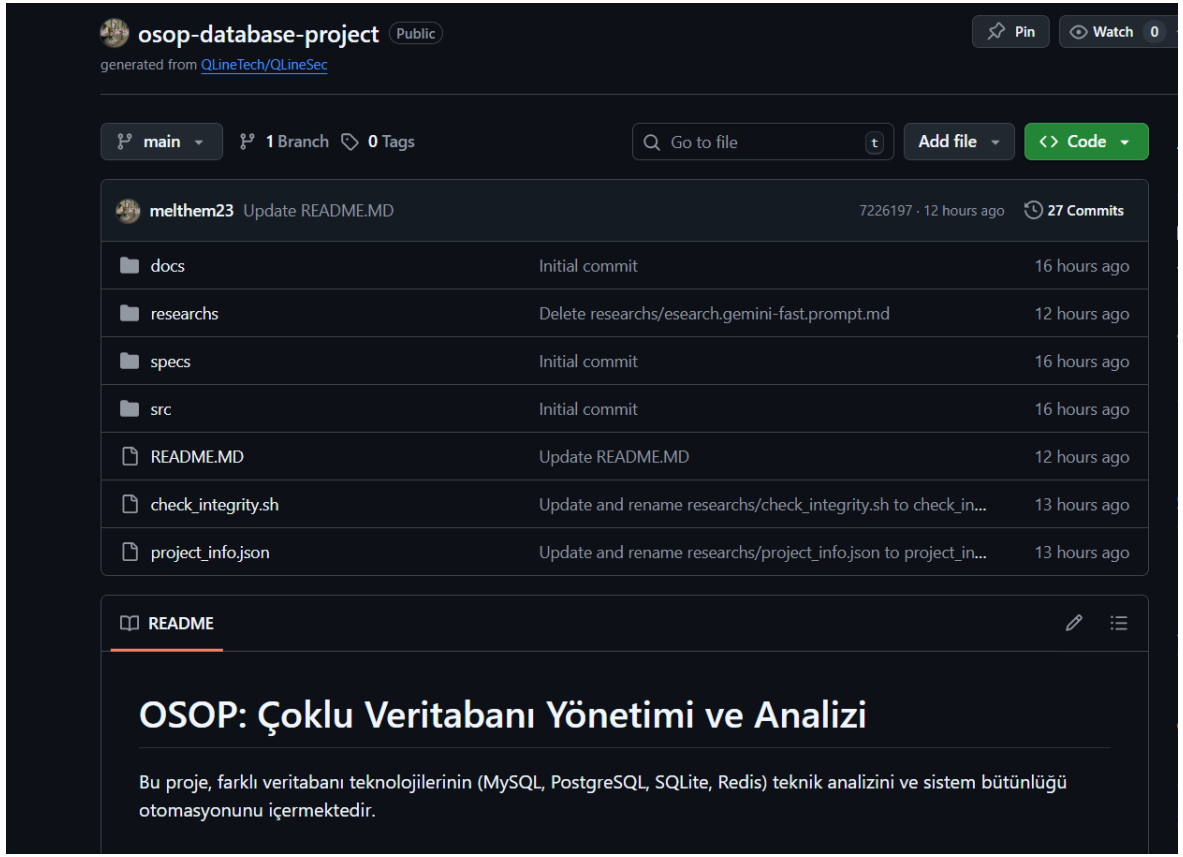
Modern Veri Sistemleri ve Otomasyon Süreçleri

Ad SoyAd: Meltem Eser

Numara: 2420191036

*Ders kodu:***MYO062**

Standartlara Uygun Dosya Yapısı

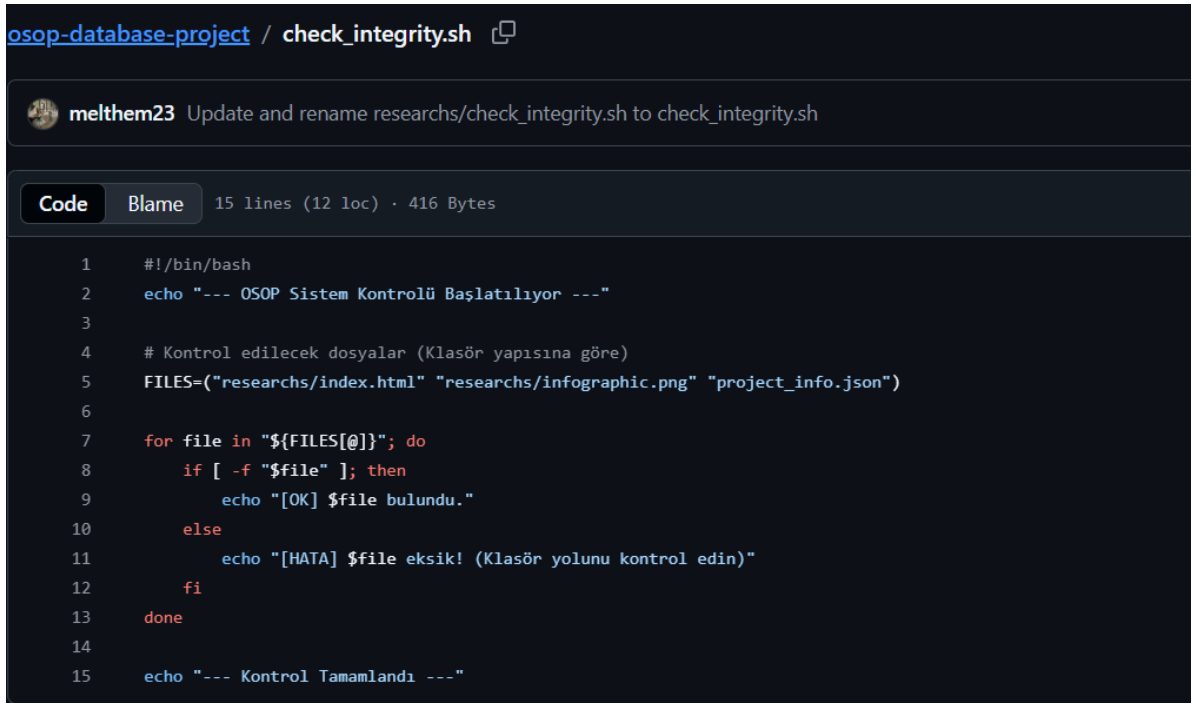


- Proje, profesyonel standartlara uygun olarak organize edilmiştir.
- **Researchs/**: Tüm raporlar ve infografikler bu klasörde.
- **README.MD**: Projenin ana dokümantasyonu.
- **Check_integrity.sh**: Sistem bütünlüğünü kontrol eden otomasyon scripti.
- **Dosya Organizasyonu**: OSOP standartlarına uygun klasörleme yapısı.
- **Merkezi Erişim**: Araştırma raporları ve otomasyon araçlarının stratejik yerleşimi.
- **Temiz Kod Politikası**: Gereksiz dosyalardan arındırılmış, teslimata hazır depo yapısı.

Teknik Analiz ve Veritabanı Karşılaştırması

- **Kapsamlı Analiz**: MySQL, PostgreSQL, SQLite ve Redis veritabanlarının mimari farkları incelenmiştir.
- **Kritik Metrikler**: Veritabanları; hız, güvenlik, ACID uyumluluğu ve ölçeklenebilirlik kriterlerine göre değerlendirilmiştir.
- **Prompt Mühendisliği**: Analizler, Gemini-Fast modeli kullanılarak optimize edilmiş promptlar ile derinleştirilmiştir.
- **Sonuç**: Proje ihtiyacına göre doğru veritabanı seçimi için teknik bir rehber oluşturulmuştur.

Otomasyon ve Sistem Güvenliği



osop-database-project / check_integrity.sh

melthem23 Update and rename researchs/check_integrity.sh to check_integrity.sh

Code Blame 15 lines (12 loc) · 416 Bytes

```
#!/bin/bash
echo "--- OSOP Sistem Kontrolü Başlatılıyor ---"

# Kontrol edilecek dosyalar (Klasör yapısına göre)
FILES=("researchs/index.html" "researchs/infographic.png" "project_info.json")

for file in "${FILES[@]}"; do
    if [ -f "$file" ]; then
        echo "[OK] $file bulundu."
    else
        echo "[HATA] $file eksik! (Klasör yolunu kontrol edin)"
    fi
done

echo "--- Kontrol Tamamlandı ---"
```

check_integrity.sh: Proje dosya yapısının doğruluğunu saniyeler içinde denetleyen özel otomasyon aracı.

Hata Önleme: Manuel kontrol süreçlerindeki insan kaynaklı hataları ortadan kaldırır.

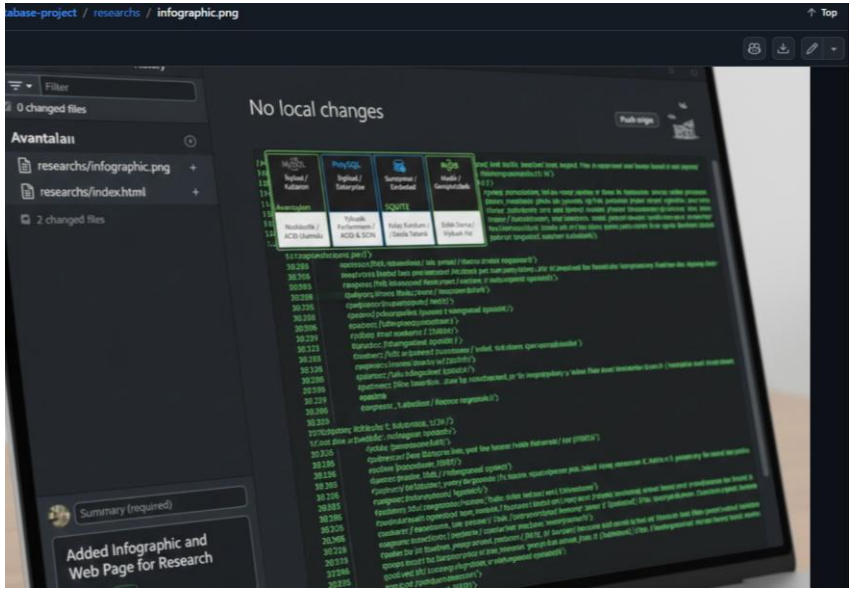
Süreklilik: Proje her güncellendiğinde, sistemin OSOP standartlarına uygunluğunu garanti eder.

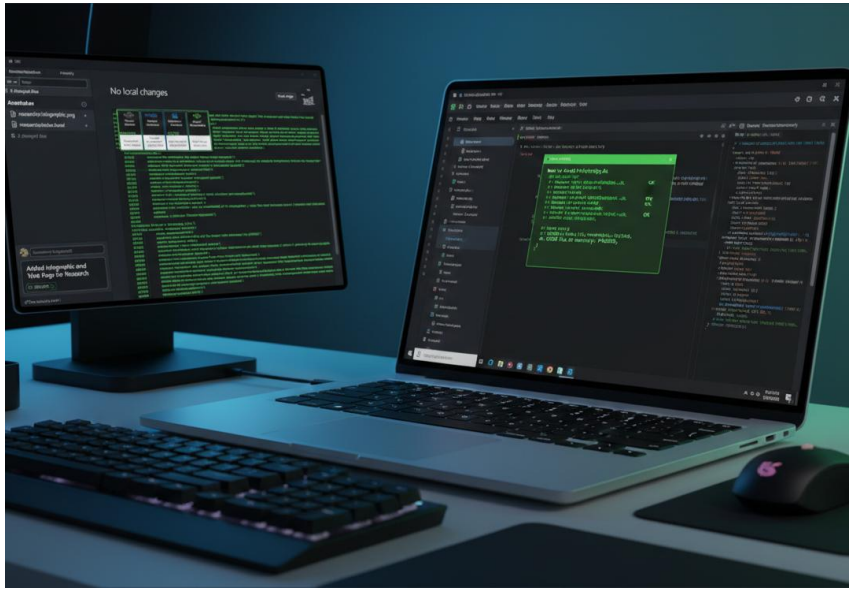
Kullanıcı Dostu: Tek bir komutla (bash check_integrity.sh) tüm sistemin sağlık taramasını gerçekleştirir.

Veri Görselleştirme (Infographic)

- **Teknikten Görsele**: Karmaşık veritabanı analizleri, herkesin anlayabileceği bir infografiğe dönüştürülmüştür.

- **Hızlı Karar Destek**: Veritabanı seçim süreçlerini hızlandıran görsel rehber.





Sonuç ve Proje Teslimi

- **Tam Uyumluluk**: Proje, OSOP (Açık Standartlı Organizasyon Projesi) standartlarına %100 uyumlu şekilde yapılandırılmıştır.
- **Otomasyon Başarısı**: check_integrity.sh betiği ile sistemin her zaman kararlı çalışması garanti altına alınmıştır.
- **Veri Odaklı Karar**: Yapılan teknik analizler, modern web mimarileri için en optimize veritabanı seçimlerini (MySQL, PostgreSQL, Redis) raporlamıştır.
- **Gelecek Vizyonu**: Proje, AI araçlarının (Gemini, Kimi) yazılım geliştirme süreçlerine entegrasyonu için başarılı bir örnek teşkil etmektedir.

Hangi Veritabanı Nerede Kullanılır?

Hız Gerekiyorsa (Caching): **Redis** seçilmelidir; çünkü veriyi RAM üzerinde tutar.

Karmaşık İlişkiler Varsa: **PostgreSQL** en iyi tercihtir; veri bütünlüğü ve standartlara uyumu çok yüksektir.

Standart Web Uygulamaları: **MySQL** kullanılmalıdır; geniş topluluk desteği ve okuma hızı avantajlıdır.

Basit ve Yerel Depolama: **SQLite** tercih edilir; sunucu gerektirmez, tek bir dosya olarak çalışır.

Dinlediğiniz İçin Teşekkürler!

- **Proje Durumu**: OSOP standartlarına uygun olarak başarıyla tamamlanmıştır.
- **Açık Kaynak**: Projenin tüm teknik detaylarına ve kodlarına GitHub üzerinden erişilebilir.
- **İletişim & Portfolyo**:
- **GitHub**: https://github.com/melthem23
- **Proje Deposu**: osop-database-project