

STAJDA YAPILAN UYGULAMALAR [ŞİKAYET KAYIT VE İZLEME MODÜLÜ]

AMAÇ

Müşterilerden gelen şikâyetlerin sistematik olarak kaydedilmesi ve çözüm sürecinin takip edilebilmesi.

GELİŞTİRME

Python ile masaüstü uygulamada form tabanlı arayüz oluşturuldu. MySQL veritabanına şikâyet bilgilerini kaydeden parametrik INSERT sorguları hazırlandı. Şikâyetlere durum (açık, çözümde, kapalı) ve öncelik alanları eklendi.

SONUÇ

Tüm müşteri şikâyetleri tek bir merkezi veritabanında toplandı. Kullanıcılar, şikâyetleri hızlıca ekleyip güncelleyebildi. Çözüm süresi takibi mümkün hale geldi ve raporlama kolaylaştı.

[ŞİKAYET ÇÖZÜM SÜRECİ RAPORLAMA]

AMAÇ

Müşteri şikâyetlerinin çözülme sürelerini ölçmek ve performans göstergelerini analiz etmek.

GELİŞTİRME

MySQL’de şikâyet başlangıç ve bitiş tarihlerini kullanarak VIEW ve sorgular oluşturuldu. Python tarafında Pandas ve Matplotlib ile grafiksel raporlar üretildi. Kullanıcı arayüzüne rapor görüntüleme ekranı eklendi.

SONUÇ

Ortalama çözüm süresi ve yoğun şikâyet dönemleri görülebilir hale geldi. Yönetim, çözüm süresini iyileştirmek için veri temelli kararlar aldı.

ŞİKAYET ÖNCELİKLENDİRME ALGORİTMASI

AMAÇ

Acil veya önemli şikâyetlerin öncelikli olarak ele alınmasını sağlamak.

GELİŞTİRME

Şikâyet türü, müşteri önemi ve aciliyet seviyesine göre puanlama sistemi tasarlandı. Python arayüzünde sıralama ve filtreleme fonksiyonları entegre edildi. MySQL tarafında gerekli sütunlar ve indeksler oluşturuldu.

SONUÇ

Yöneticiler en kritik şikâyetleri hızlıca tespit edebildi. Müşteri memnuniyeti arttı, çözüm süreci daha

A LOG VE OTOMATİK BİLDİRİM SİSTEMİ

AMAÇ

Uygulama ve veritabanında oluşan hataların anlık olarak takip edilmesi ve ilgili personele bildirilmesi.

GELİŞTİRME

MySQL sorgu hataları ve Python uygulama hataları log dosyasına kaydedildi. Python ile günlük kritik hata raporlarını e-posta ile yöneticilere gönderme modülü oluşturuldu.

SONUÇ

Hatalar hızlı şekilde tespit edilip müdahale edilebildi. Sistemde kesinti süresi ve veri kaybı minimize edildi.