

Veri Odaklı Banka Yönetim Sistemi

Ankara Üniversitesi Yapay Zeka Ve Veri Mühendisliği
Veri Tabanı Yönetim Sistemleri Dersi Final Projesi Sunumu

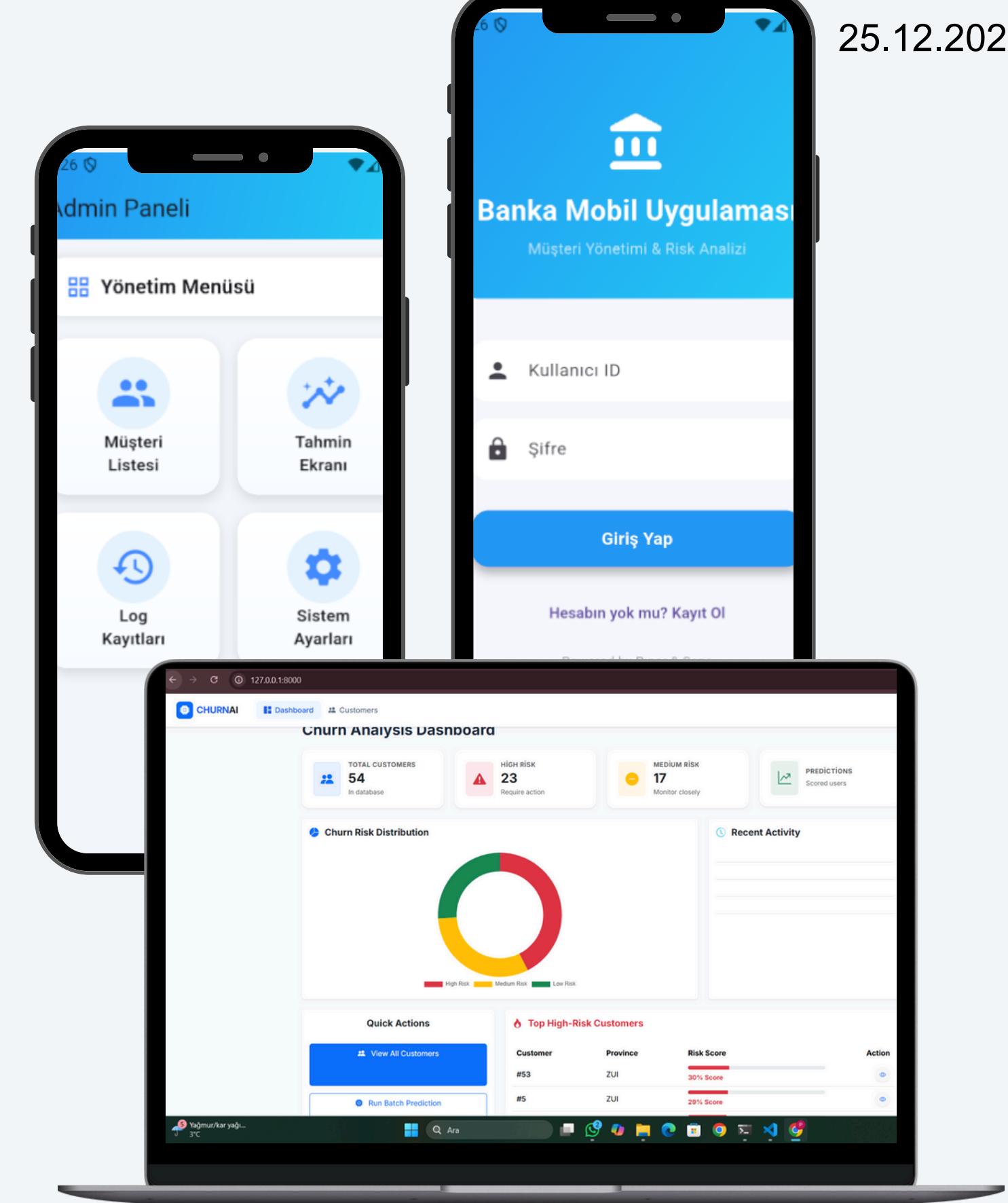
Geliştiren Öğrenciler:

Pınar Kocagöz (22290016)

Hasna Sena Kaymak (22290684)



Github Reposu



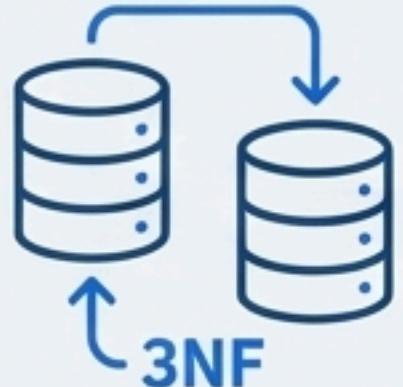
Banka Mobil Uygulaması: Müşteri Yönetimi & Risk Analizi

Veri Tabanı Yönetim Sistemleri dersi kapsamında geliştirilen, SQL ve Makine Öğrenmesi destekli bütünsel bir sistem.

The image displays two views of a banking application. On the left is a smartphone screen showing the login page of the mobile app. It features a blue header with a bank icon and the text 'Banka Mobil Uygulaması'. Below this is a subtitle 'Müşteri Yönetimi & Risk Analizi'. The main form has fields for 'Kullanıcı ID' and 'Şifre', both with placeholder text. A large blue 'Giriş Yap' button is at the bottom. Below it, a link 'Hesabın yok mu? Kayıt Ol' is visible. At the very bottom, it says 'Powered by Pınar & Sena'. On the right is a laptop screen showing the 'CHURN AI Dashboard'. The dashboard includes a summary card with 'TOTAL CUSTOMERS 54', 'HIGH RISK 23' (with a red upward arrow), 'MEDIUM RISK 17' (with a yellow circle), and 'PREDICTIONS Saved' (with a green right arrow). Below this is a 'Churn Risk Distribution' chart showing segments for High Risk (red), Medium Risk (yellow), and Low Risk (green). There are also sections for 'Recent Activity' (listing four items) and 'Top High-Risk Customers' (listing two customers with risk scores of 98% and 96%).

Teorik Bilgiyi Pratiğe Dönüşürtmek: Projenin Hedefleri

Ankara Üniversitesi Yapay Zeka ve Veri Mühendisliği Bölümü'nün 'Veri Tabanı Yönetim Sistemleri' dersi final projesi olarak geliştirilen bu çalışma, teorik DBMS kavramlarını gerçekçi bir senaryoda uygulamayı hefler.



İlişkisel Veritabanı Tasarımı

SQLite üzerinde normalize edilmiş (3NF), tutarlı ve ölçeklenebilir bir veritabanı şeması oluşturmak.



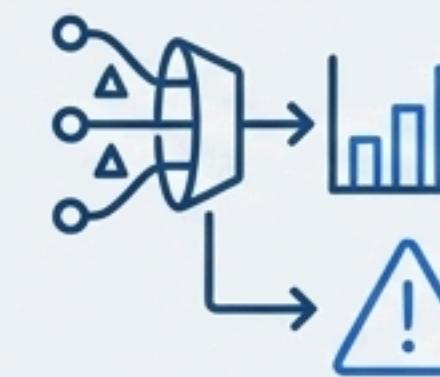
Uygulamalı SQL

'JOIN', 'GROUP BY', 'RANK()' gibi ileri düzey SQL sorgularını kullanarak veri analizi yapmak ve bu analizleri kullanıcı arayüzüne yansıtma.



Bütünleşik Sistem

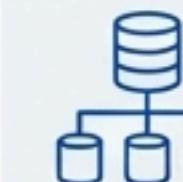
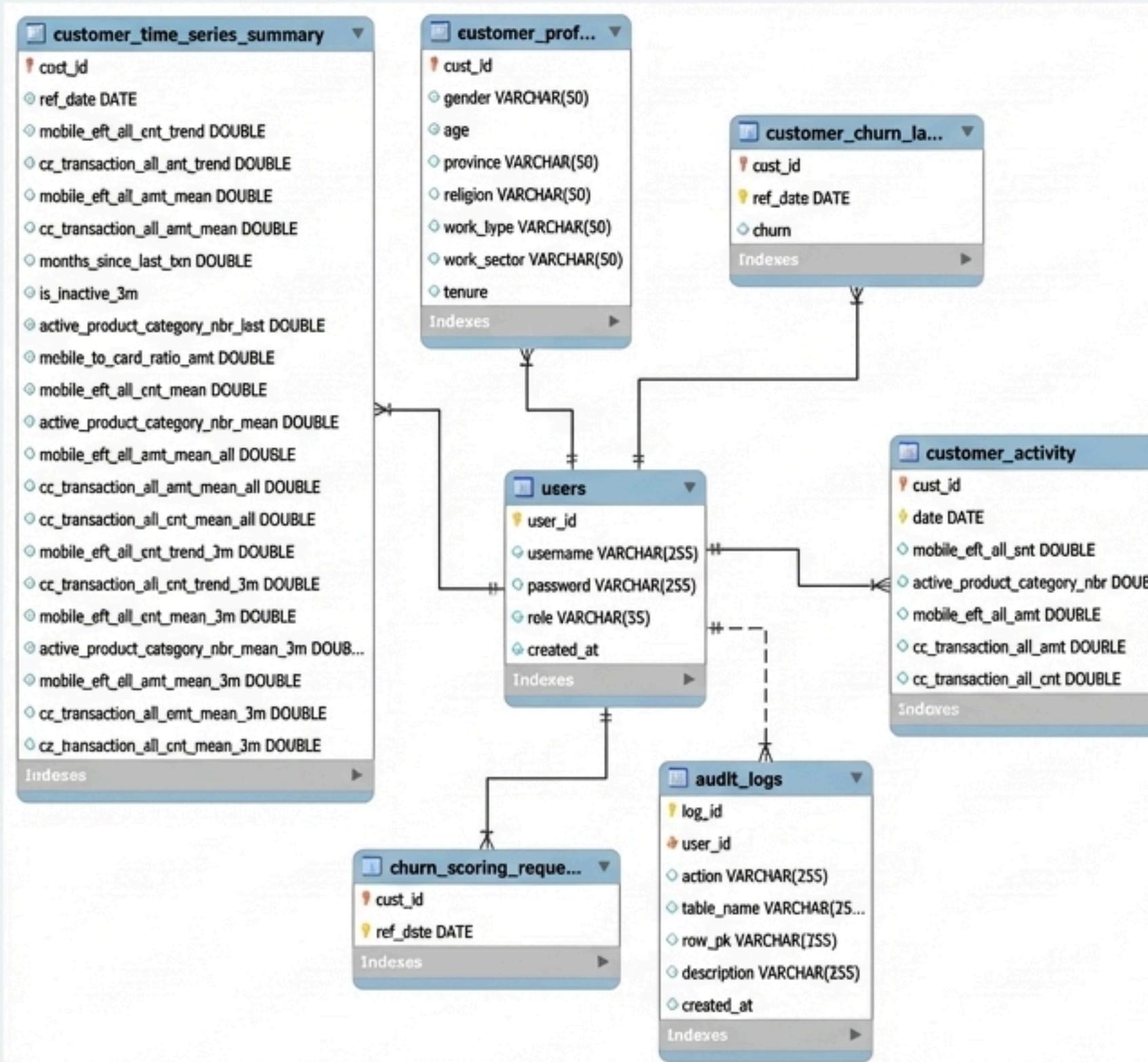
Tek bir SQL veritabanını paylaşan, iki farklı platformda (Flutter Mobil ve Django Web) çalışan, uçtan uca bir uygulama geliştirmek.



Veri Odaklı Değer

Ham veriyi, hem son kullanıcı (müşteri) hem de yönetici için anlamlı raporlara, grafiklere ve churn riski gibi eyleme geçirilebilir içgörülere dönüştürmek.

İlişkisel Veri Tabanı Tasarımı



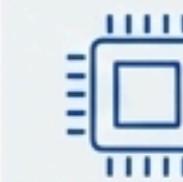
- **Yapı:** 7 ana tablodan oluşur: 'Users` (ana tablo), 'CustomerProfile`, 'CustomerActivity`, 'CustomerTimeSeriesSummary`, 'CustomerChurnLabel`, 'AuditLogs` ve 'churn_scoring_requests`.



- **Normalizasyon:** Tüm tablolar, veri tekrarını ve anormallilikleri önlemek amacıyla 3. Normal Form (3NF) kurallarına uygun olarak tasarlanmıştır.

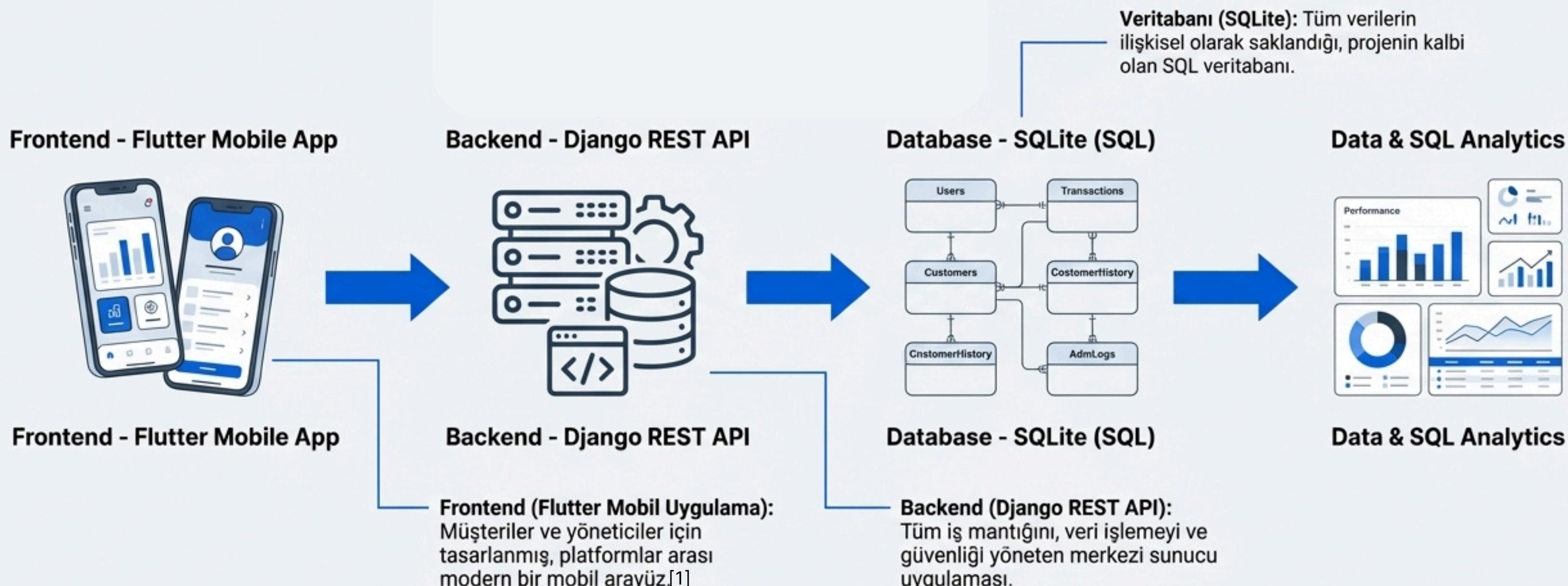


- **Veri Bütünlüğü:** 'Users` tablosu ile diğer tablolar arasındaki tüm ilişkiler, Foreign Key kısıtlamaları ile tanımlanarak veri tutarlılığı güvence altına alınmıştır.



- **Veri Kaynağı:** Prototip, Kaggle'in 'ING Datathon' veri setinden alınan 50 kullanıcılık bir örneklem üzerinde çalışmaktadır.[2]

Sistemin Anatomisi: Parçalar Nasıl Birleşiyor?



Flutter Arayüzü → Django REST API → SQLite Veritabanı (SQL Sorguları) → Analiz ve Raporlar

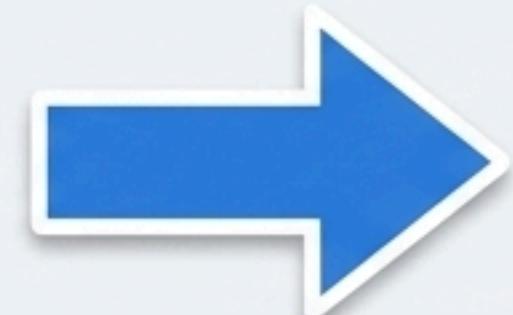
Uygulamadaki tüm analitik özellikler, grafikler ve raporlar, arka planda çalışan güçlü SQL sorguları tarafından anlık olarak üretilmektedir.

Kullanılan SQL Yetenekleri ve Uygulama Alanları

SQL Anahtar Kelimesi / Kavramı	Projede Kullanım Alanı (Pratik Uygulama)
SELECT, WHERE	Kullanıcı doğrulama, belirli bir müşterinin profilini getirme.
INNER JOIN	'users' ve 'customer_profile' tablolarını birleştirerek tam müşteri bilgisi oluşturma.
GROUP BY, SUM, AVG	Aylık EFT/Kart işlem özetlerini ve toplam harcamaları hesaplama.
ORDER BY	Müşteri listelerini tarihe veya isme göre sıralama.
RANK() Pencere Fonksiyonu	Müşterileri toplam harcama gibi metriklere göre sıralayarak 'Kampanyalar' özelliğini güçlendirme.

Sorgudan Arayüze: Müşteri Profili Bilgileri

```
SELECT  
    u.username,  
    u.created_at,  
    p.gender,  
    p.age,  
    p.province,  
    p.work_type,  
    p.work_sector,  
    p.tenure  
FROM users u  
JOIN customer_profile p ON u.user_id = p.cust_id  
WHERE u.user_id = 7;
```



Kullanıcının demografik bilgileri (`customer_profile`) ve temel hesap bilgileri (`users`) iki ayrı tabloda tutulmaktadır. Bu ekran, iki tabloyu `JOIN` ile birleştirerek kullanıcıya bütünsel bir görünüm sunan SQL sorgusunun sonucudur.

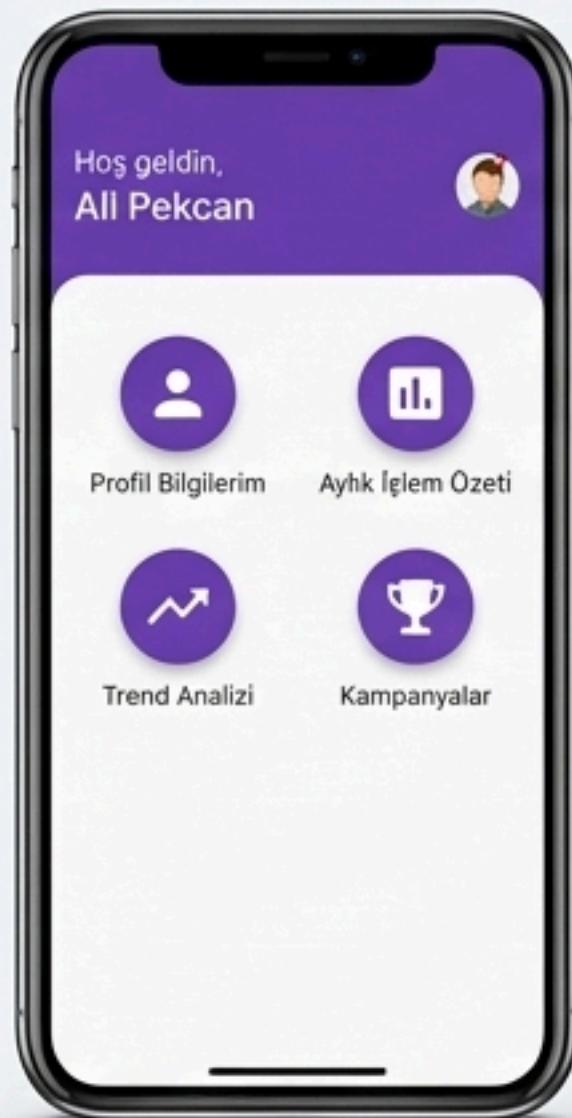
Sorgudan Arayüze: Kampanya Sıralamaları

```
SELECT rank  
FROM (  
    SELECT  
        cust_id,  
        RANK() OVER (ORDER BY SUM(mobile_eft_all_amt) DESC) AS rank  
    FROM customer_activity  
    GROUP BY cust_id  
) ranked  
WHERE cust_id = 7;
```



‘RANK()’ pencere fonksiyonu, tüm müşteriler arasında dinamik bir sıralama oluşturmamızı sağlar. Bu sorgu, her kullanıcının en yüksek harcama yapanlar listesindeki yerini görmesini sağlayarak etkileşimi artırır.

Müşteri Gözünden: Flutter Mobil Uygulama Deneyimi



Profil Yönetimi



Aylık İşlem Özeti



Trend Analizi



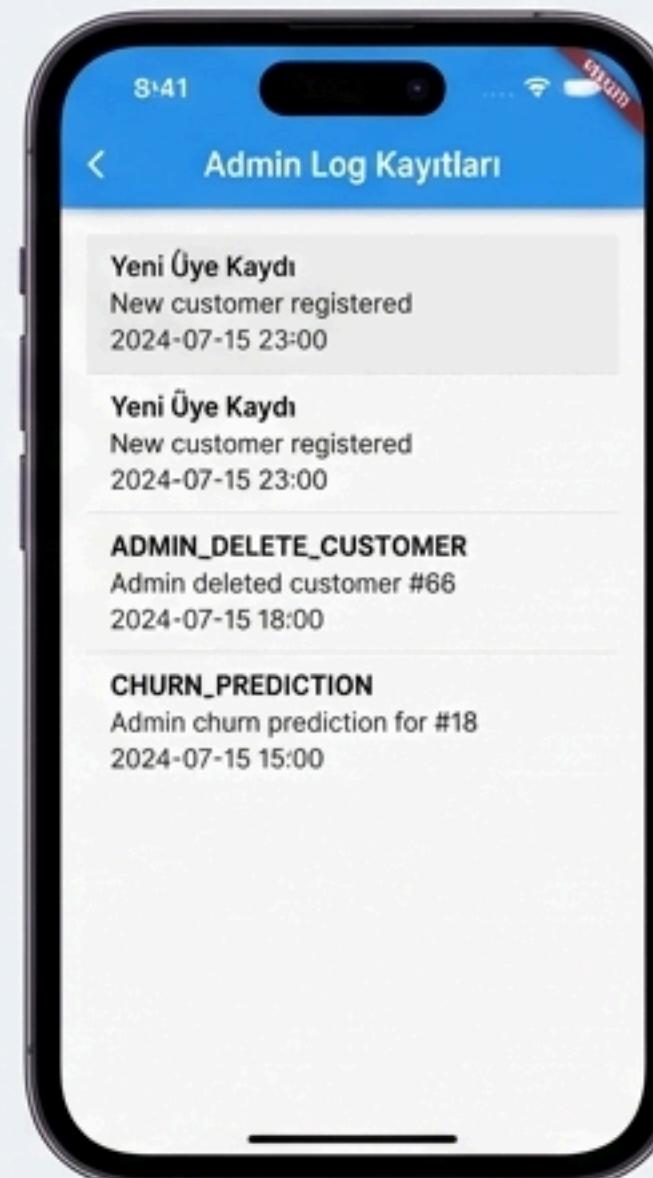
Kampanyalar

Yönetici Kontrolü: Flutter Mobil Admin Paneli

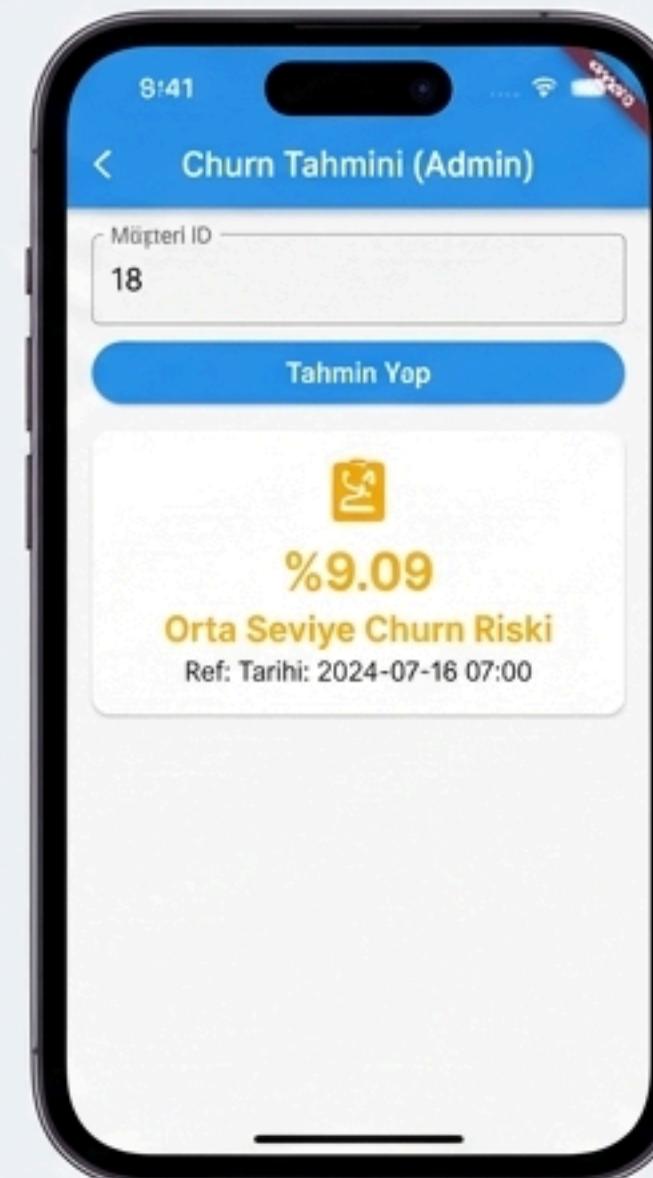
Müşteri Yönetimi



Denetim Kayıtları (Audit Logs)



Anlık Churn Tahmini



Sistem İstatistikleri



Her Adım Kayıt Altında: Güvenlik için Audit Log Mekanizması

`audit_logs` tablosu, sistemde gerçekleşen tüm kritik yönetici ve sistem işlemlerini geri dönük olarak izlemek, güvenlik ve hesap verebilirliği sağlamak için tasarlanmıştır.

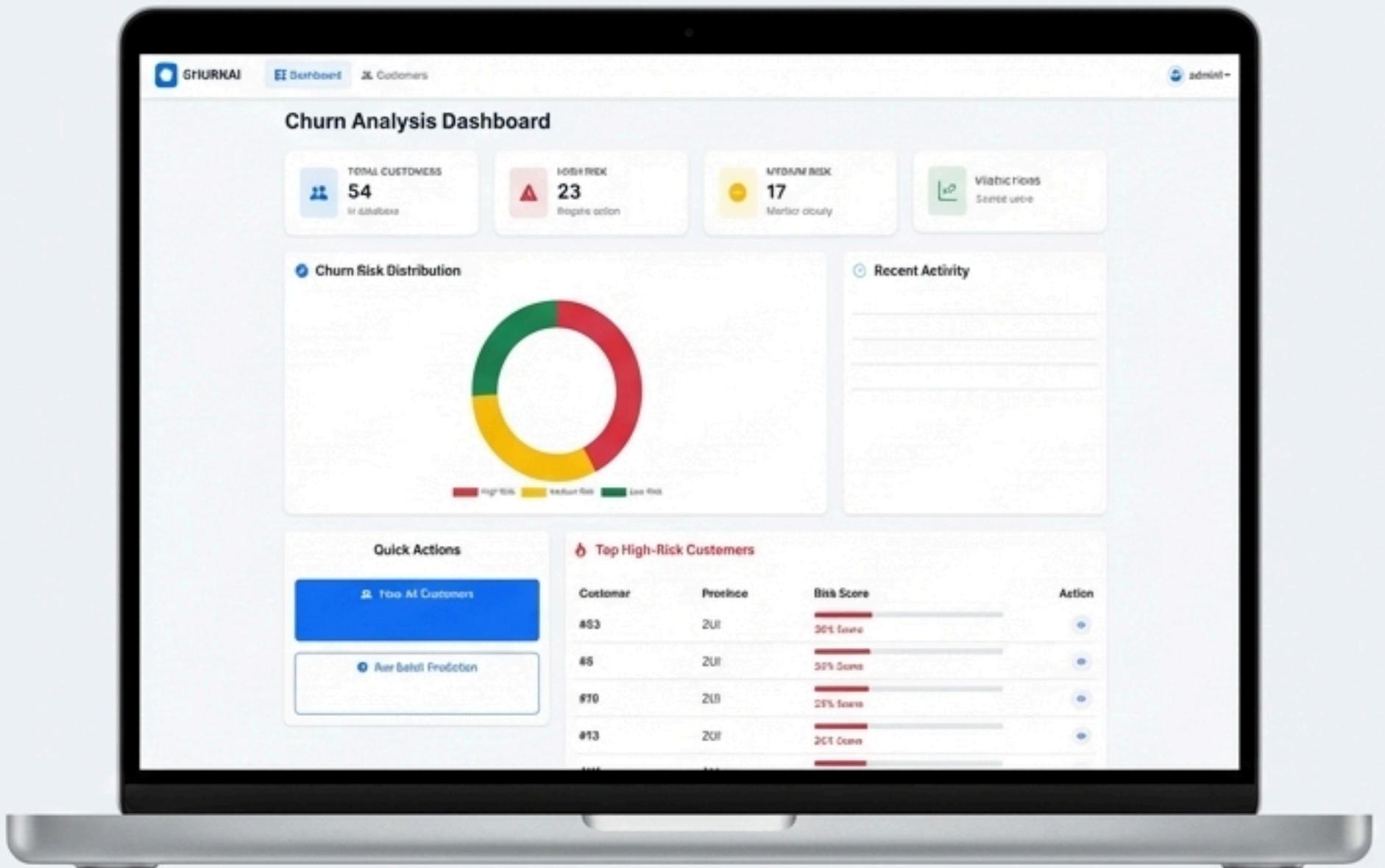
Sol Taraf (SQL)

```
INSERT INTO audit_logs (user_id, action, table_name, row_pk,  
description, created_at)  
VALUES (1, 'ADMIN_DELETE_CUSTOMER', 'users', '7', 'Admin  
deleted customer with user_id = 7', NOW());  
  
INSERT INTO audit_logs (user_id, action, table_name, row_pk,  
description, created_at)  
VALUES (1, 'CHURN_PREDICTION', 'customer_churn_label', '7',  
'Admin churn prediction for customer #7', NOW());
```

Sağ Taraf (Arayüz)



Django Web Yönetim Paneli



Dashboard'un Temel Yetenekleri

- Risk Dağılımı:** Müşterileri Yüksek, Orta ve Düşük risk gruplarına ayıran interaktif donut grafiği.
- Önceliklendirme:** 'En Yüksek Riskli Müşteriler' listesi ile acil müdahale gerektiren hesapları öne çıkarma.
- Hızlı Aksiyonlar:** Tüm müşterileri listeleme veya tüm portföy için toplu risk tahmini (batch prediction) başlatma.
- Akıllı Filtreleme:** Müşteri portföyünü lokasyon, sektör veya risk seviyesine göre anlık filtreleyerek detaylı analiz yapma imkanı.

Detaylı Analiz: Müşteri Portföy Yönetimi

CHURNAI Dashboard Customers

Home / Customers Customer Directory

Akıllı Filtreleme: ID, lokasyon, sektör ve risk kategorisine göre anlık sorgulama.

SEARCH RECORDS ID, Province, or Sector... RISK CATEGORY All Risk Levels Apply Filters

All Customers 24

Customer ID	Demographics	Location	Tenure	Sector	Risk Status	Action
#1 #1	56 yrs + F	ZUI	65 Months	Finance	Medium	🔗
#2 #2	62 yrs + F	NOB	224 Months	Healthcare	Medium	🔗
#3 #3	22 yrs + F	ZUI	47 Months		Medium	🔗
#4 #4	18 yrs + F	ANK	55 Months	none	Low	🔗
#5 #5	27 yrs + M	ZUI	108 Months	Finance	High	🔗
#6 #6	40 yrs + F	NOH	187 Months		High	🔗
#7 #7	64 yrs + F	GEL	218 Months	Public Sector	Medium	🔗
#8 #8	35 yrs + F	ZUI	184 Months	Retail	High	🔗
#9 #9	52 yrs + M	GRO	216 Months	Healthcare	High	🔗
#10 #10	31 yrs + F	FLE	156 Months	Manufacturing	Low	🔗

Merkezi Takip: Tüm müşteri portföyünün tek ekrandan yönetimi.

Akıllı Filtreleme: ID, lokasyon, sektör ve risk kategorisine göre anlık sorgulama.

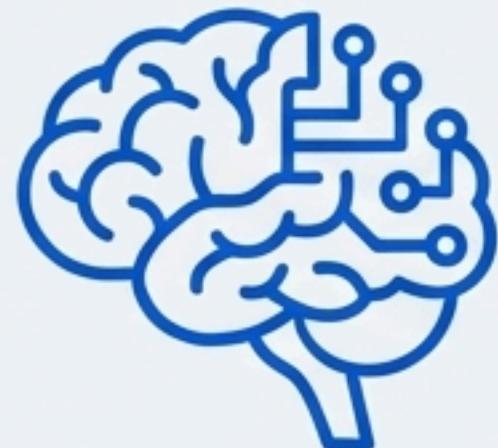
Hızlı Raporlama: Seçilen verilerin analiz için tek tıkla CSV formatında dışa aktarılması.

Önceliklendirme: Kritik müşterileri hızla bulmayı sağlayan renkli risk etiketleri (High, Medium, Low).

ML Destekli Churn Risk Analizi

Müşteri terk (churn) tahmini için olasılık temelli bir **CatBoost Sınıflandırma Modeli** kullanılmıştır.

Adım 1: Olasılık Hesaplama



Adım 2: Eşik Değer Kontrolü

```
churn_prob = churn_model.predict_proba(X)[0][1]
churn_pct = round(churn_prob * 100, 2)

if churn_pct >= 15:
    risk = "HIGH"
    churn_label = 1
elif churn_pct >= 5:
    risk = "MEDIUM"
    churn_label = 1
else:
    risk = "LOW"
    churn_label = 0
```



Adım 3: Risk Etiketleme



Model, her müşteri için 0 ile 1 arasında bir churn olasılık değeri (`churn_prob`) üretir.

Bu olasılık, önceden tanımlanmış iş kuralları (eşik değerler: >15%, >5%) ile karşılaştırılır.

Sonuç, yönetici için anlaşılır bir risk seviyesine ('Yüksek', 'Orta', 'Düşük') dönüştürülür.

Kazanımlar ve Gelecek Potansiyeli

Projenin Sağladığı Temel Kazanımlar

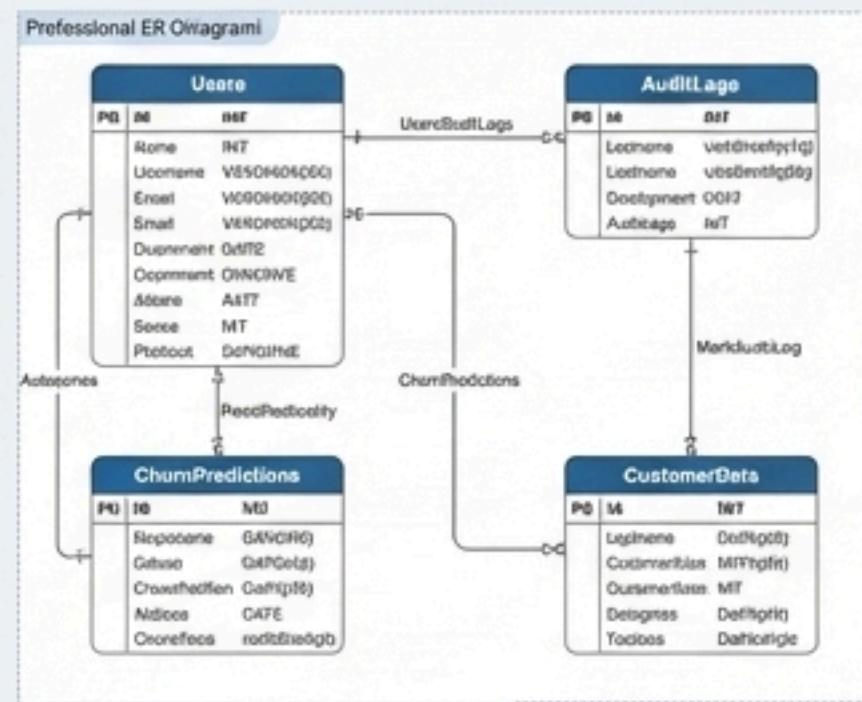
- ✓ Gerçek hayata uygun, **ilişkisel veritabanı tasarıımı** pratiği.
- ✓ **SQL sorgularının** bir uygulamanın iş mantığının merkezine nasıl yerleştirileceğinin anlaşılması.[4]
- ✓ **Backend - Frontend veri akışının** REST API üzerinden başarıyla yönetilmesi.
- ✓ **ORDER BY, GROUP BY** gibi **DBMS konularının** pratik karşılığının deneyimlenmesi.
- ✓ Veri analizi ve görselleştirme **integrasyonu** yetkinliği.

Gelecek Geliştirme Alanları

- ⚙️ Sistemin daha büyük veri setleriyle **performans testlerinin** yapılması.
- ⚙️ Web paneline daha gelişmiş **raporlama ve veri görselleştirme** özelliklerinin eklenmesi.
- ⚙️ Churn tahmin modelinin periyodik olarak yeni verilerle **otomatik olarak yeniden eğitilmesi**.

Kavramdan Koda, Koddan Anlamlı Bilgiye

Bu proje, **Veri Tabanı Yönetim Sistemleri** dersinde öğrenilen teorik bilgilerin; **SQL + Backend + Mobil uygulama** bütünlüğü içinde, veri odaklı ve gerçekçi bir senaryo ile nasıl hayatı geçirilebileceğinin somut bir örneğidir.



KAVRAM (ER DİYAGRAMI)

```
1  SELECT
2      usersName,
3          auditName,
4          auditAppSnt,
5          ChurnPredictions.stex
6  FROM
7      Veri.ia.Tabani.Yonetin.Sistemleri
8      JGIN ChurnPredictions.as
9      JGIN ChurnPredictions.table.test(oratbiheltoesoe)
10     MIRE collect.AuditiensData
11     AND (ChurnPrediction
12         ) AND ChurePredictien.s AS 1)
13     MIRE CustomerData.a.songiks = Churn, nsentfic = null;
14     GROUP BY group by
15     ORDER BY new_Users;
```

KOD (SQL SORGUSU)



ANLAMLI BİLGİ
(WEB DASHBOARD)

Kaynakça

- [1] Flutter Team, “Flutter Documentation,” Google. [Online]. Available: <https://docs.flutter.dev>. Accessed: Dec. 2025.
- [2] Kaggle, “Bank Customer Churn Dataset,” Kaggle Inc. [Online]. Available: <https://www.kaggle.com/competitions/ing-hubs-turkiye-datathon>
- [3] Django Software Foundation, “Django REST Framework Documentation.” [Online]. Available: <https://www.django-rest-framework.org>. Accessed: Dec. 2025.
- [4] SQLite Consortium, “SQLite Documentation.” [Online]. Available: <https://www.sqlite.org/docs.html>. Accessed: Dec. 2025.