

Öğrenci Adı Soyadı : Furkan Köksalan

Öğrenci Numarası: 170423026

T.C. MARMARA ÜNİVERSİTESİ TEKNOLOJİ FAKÜLTESİ
2025-2026 EĞİTİM ÖĞRETİM YILI GÜZ
DÖNEMİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
VERİTABANI YÖNETİM SİSTEMLERİ DERSİ PROJE-FİNAL DÖKÜMANI

Tüm sorular soru metnini altındaki alanlara cevaplandırılacaktır. Bu belgenin formatını bozmayınız, sadece gerekli alanları doldurunuz.

- 1. Gerçekleştirdiğiniz veri tabanı projesi için grup arkadaşlarınızın isimlerini yazınız ve projenize ait veri tabanı/diğer yazılım bileşenleri hakkında bilgi veriniz. (5 p)**

Grup Üyeleri:

Ahmet Eren Başalı
Yusuf Durmuş
Furkan Köksalan

Kullanılan Veri Tabanı ve Yazılım Bileşenleri:

Veri Tabanı: PostgreSQL
Backend: Python – FastAPI
ORM: SQLAlchemy
Veri Doğrulama: Pydantic
Frontend: React

- 2. Gerçekleştirdiğiniz veri tabanı projesi için proje dokümanınızı ve dosyalarınızı içeren herkese açık Github bağlantılarınızı paylaşınız. (5 p)**

https://github.com/furkankoksalan/VTYS_Proje

- 3. Gerçekleştirdiğiniz projenin amacını detaylı bir şekilde açıklayınız. (10 p)**

Bu projenin temel amacı, **kayıp evcil hayvan vakalarında hayvan sahipleri ile hayvanı bulan kişileri hızlı, güvenli ve doğru şekilde bir araya getirebilen dijital bir platform geliştirmektir**. Günümüzde kayıp evcil hayvan ilanları çoğunlukla sosyal medya veya dağınık platformlar üzerinden paylaşılmasında, bu durum bilgi kirliliğine ve ilanların kısa sürede kaybolmasına neden olmaktadır. Projemiz, bu sorunu çözmek için merkezi, düzenli ve veri tabanı temelli bir yapı sunmayı hedeflemektedir.

Proje kapsamında kullanıcıların **kayıp veya bulunan evcil hayvanlar için detaylı ilanlar oluşturabilmesi** amaçlanmıştır. Bu ilanlarda hayvanın türü, cinsi, rengi, kaybolduğu veya bulunduğu konum, tarih bilgisi ve fotoğraflar gibi kritik veriler saklanmaktadır. Böylece kullanıcılar, aradıkları hayvana en uygun ilanlara hızlı şekilde ulaşabilmektedir. Veri tabanı tasarımlı bu ihtiyaçlar doğrultusunda normalleştirilmiş ve veri tutarlılığı ön planda tutulmuştur.

Projenin bir diğer önemli amacı, **kullanıcılar arasında güvenli ve kontrollü bir iletişim altyapısı oluşturmak**tır. Bu kapsamında mesajlaşma sistemi tasarlanmıştır; kullanıcıların ilanlar üzerinden doğrudan iletişime geçebilmesi, ancak iletişim sisteminde kayıt altına alınması sağlanmıştır. Bu yapı hem kullanıcı güvenliğini artırmakta hem de geçmiş konuşmaların izlenebilir olmasını sağlamaktadır.

Sonuç olarak bu proje, **toplumsal fayda sağlayan gerçek bir probleme çözüm üretiyor**.

4. Tasarladığınız veri tabanı mimarisinde hangi tablo ve ilişkileri kullanıldığınızı açıklayınız. (10 p)

Countries

Alan Adı	Veri Tipi
country_id	VARCHAR(16)
name	VARCHAR(100)
country_code	VARCHAR(10)

Cities

Alan Adı	Veri Tipi
city_id	VARCHAR(16)
country_id	VARCHAR(16)
name	VARCHAR(100)

Districts

Alan Adı	Veri Tipi
district_id	VARCHAR(16)
city_id	VARCHAR(16)
name	VARCHAR(100)

Neighborhoods

Alan Adı	Veri Tipi
neighborhood_id	VARCHAR(16)
district_id	VARCHAR(16)
name	VARCHAR(120)

Users

Alan Adı	Veri Tipi
user_id	UUID
full_name	VARCHAR(150)
email	VARCHAR(150)
phone	VARCHAR(30)
password_hash	VARCHAR(255)
primary_address_id	UUID
created_at	TIMESTAMP
updated_at	TIMESTAMP

Addresses

Alan Adı	Veri Tipi
address_id	UUID
country_id	VARCHAR(16)
city_id	VARCHAR(16)
district_id	VARCHAR(16)
neighborhood_id	VARCHAR(16)
address_label	VARCHAR(80)
street	VARCHAR(200)
building_no	VARCHAR(50)
apartment_no	VARCHAR(50)
postal_code	VARCHAR(30)
created_at	TIMESTAMP
updated_at	TIMESTAMP

User_addresses

Alan Adı	Veri Tipi

user_id	UUID
address_id	UUID
is_default	BOOLEAN
created_at	TIMESTAMP

Species

Alan Adı	Veri Tipi
species_id	VARCHAR(16)
name	VARCHAR(80)

Colors

Alan Adı	Veri Tipi
color_id	VARCHAR(16)
name	VARCHAR(50)

Pets

Alan Adı	Veri Tipi
pet_id	UUID
owner_user_id	UUID
species_id	VARCHAR(16)
name	VARCHAR(100)
sex	VARCHAR(20)
age_years	INT
microchip_no	VARCHAR(80)
distinctive_marks	VARCHAR(2000)
created_at	TIMESTAMP
updated_at	TIMESTAMP

Listings

Alan Adı	Veri Tipi
listing_id	UUID
created_by	UUID
pet_id	UUID
species_id	VARCHAR(16)
address_id	UUID
listing_type	VARCHAR(20)
status	VARCHAR(20)
title	VARCHAR(200)
description	VARCHAR(4000)
event_time	VARCHAR(64)
created_at	TIMESTAMP
updated_at	TIMESTAMP

listing_photos

Alan Adı	Veri Tipi
photo_id	UUID
listing_id	UUID
file_name	VARCHAR(255)
data_url	TEXT
sort_order	INT
created_at	TIMESTAMP

Conversations

Alan Adı	Veri Tipi
conversation_id	UUID
listing_id	UUID
created_at	TIMESTAMP

Conversation_participants

Alan Adı	Veri Tipi
conversation_id	UUID
user_id	UUID
role	VARCHAR(30)
created_at	TIMESTAMP

Messages

Alan Adı	Veri Tipi
message_id	UUID
conversation_id	UUID
sender_user_id	UUID
body	VARCHAR(2000)
created_at	TIMESTAMP

Lisiting_matches

Alan Adı	Veri Tipi
lost_listing_id	UUID
found_listing_id	UUID
score	INT
reasons	JSONB
created_at	TIMESTAMP

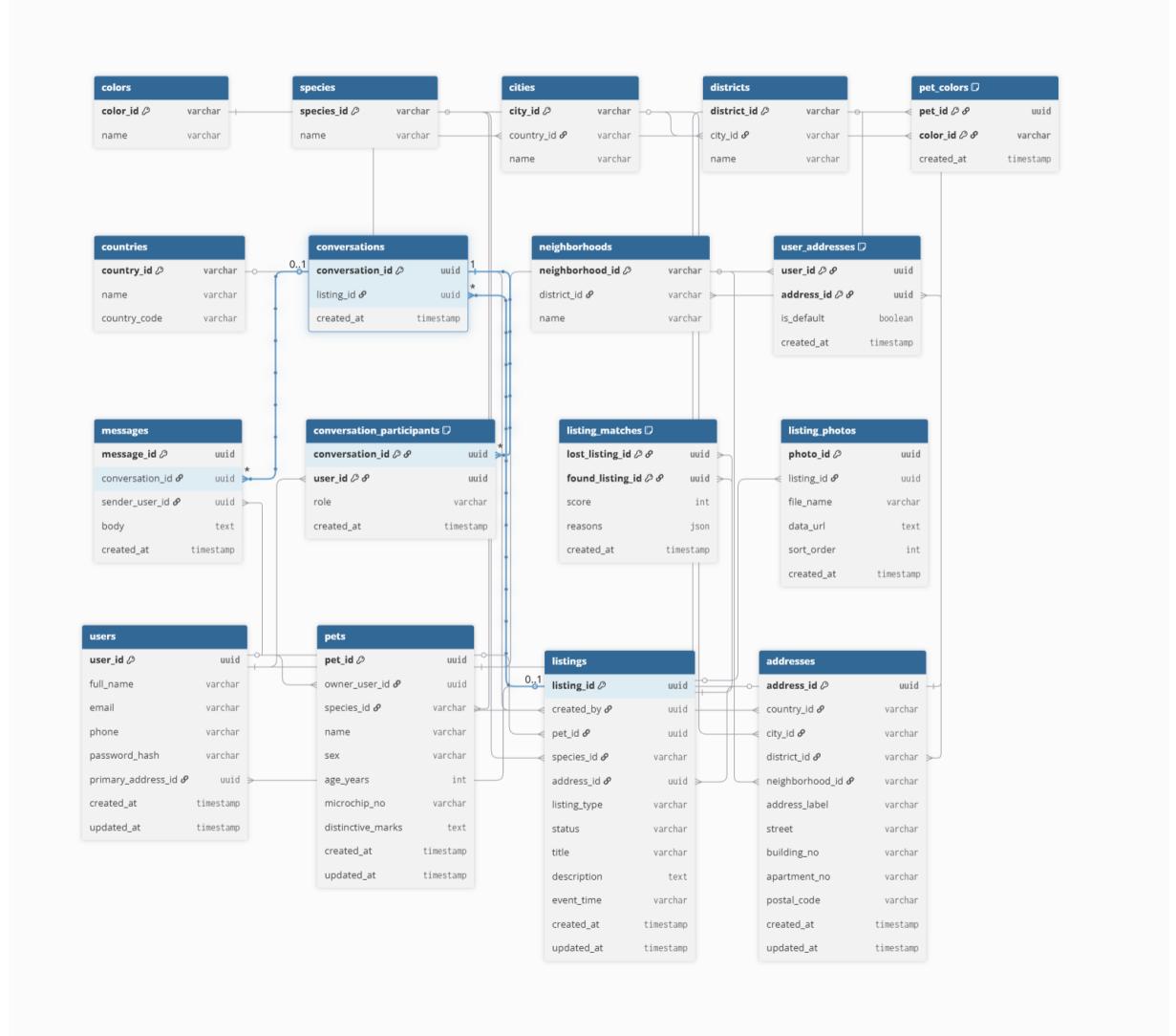
1) cities.country_id → countries.country_id (1 ülke → N şehir)

2) districts.city_id → cities.city_id (1 şehir → N ilçe)

3) neighborhoods.district_id → districts.district_id (1 ilçe → N mahalle)

- 4) addresses.country_id → countries.country_id (1 ülke → N adres)
 5) addresses.city_id → cities.city_id (1 şehir → N adres)
 6) addresses.district_id → districts.district_id (1 ilçe → N adres)
 7) addresses.neighborhood_id → neighborhoods.neighborhood_id (1 mahalle → N adres)
- 8) user_addresses.user_id → users.user_id (N-N için ara tablo)
 9) user_addresses.address_id → addresses.address_id (N-N için ara tablo)
- 10) pets.owner_user_id → users.user_id (1 kullanıcı → N pet)
 11) pets.species_id → species.species_id (1 tür → N pet)
- 12) pet_colors.pet_id → pets.pet_id (N-N için ara tablo)
 13) pet_colors.color_id → colors.color_id (N-N için ara tablo)
- 14) listings.created_by → users.user_id (1 kullanıcı → N ilan)
 15) listings.pet_id → pets.pet_id (1 pet → N ilan) [pet_id NULL olabilir]
 16) listings.species_id → species.species_id (1 tür → N ilan) [species_id NULL olabilir]
- 17) listings.address_id → addresses.address_id (1 adres → N ilan)
- 18) listing_photos.listing_id → listings.listing_id (1 ilan → N foto)
- 19) conversations.listing_id → listings.listing_id (1 ilan → N konuşma)
- 20) messages.conversation_id → conversations.conversation_id (1 konuşma → N mesaj)
 21) messages.sender_user_id → users.user_id (1 kullanıcı → N mesaj)

5. Veri tabanı ER (Entity Relationship) diagramının bilgisayar ortamında çizilmiş halini paylaşınız. (Ara raporda eksik kısımlar bu raporda giderilmelidir ve ER çizme programlarından faydalabilir. Elle çizim, çizip fotoğrafını çekme vb. kabul edilmeyecektir.) (10 p)



6. Herhangi iki tablonuz için DDL (create) kodları yazılmalıdır. (10 p)

DDL #1 : Kullanıcılar Tablosu

```
CREATE TABLE users (
    user_id UUID PRIMARY KEY,
    full_name VARCHAR(150) NOT NULL,
    email VARCHAR(150) NOT NULL UNIQUE,
    phone VARCHAR(30),
    password_hash VARCHAR(255) NOT NULL,
    primary_address_id UUID NULL,
```

```
    created_at TIMESTAMP NOT NULL,  
    updated_at TIMESTAMP NOT NULL  
);
```

DDL #2 : Adresler Tablosu

```
CREATE TABLE addresses (  
    address_id UUID PRIMARY KEY,  
    country_id VARCHAR(16) NOT NULL,  
    city_id VARCHAR(16) NOT NULL,  
    district_id VARCHAR(16) NOT NULL,  
    neighborhood_id VARCHAR(16) NOT NULL,  
    address_label VARCHAR(80) NOT NULL,  
    street VARCHAR(200),  
    building_no VARCHAR(50),  
    apartment_no VARCHAR(50),  
    postal_code VARCHAR(30),  
    created_at TIMESTAMP NOT NULL,  
    updated_at TIMESTAMP NOT NULL,  
    FOREIGN KEY (country_id) REFERENCES countries(country_id),  
    FOREIGN KEY (city_id) REFERENCES cities(city_id),  
    FOREIGN KEY (district_id) REFERENCES districts(district_id),  
    FOREIGN KEY (neighborhood_id) REFERENCES neighborhoods  
(neighborhood_id));
```

7. 5 adet DML (update, insert, delete) içeren kodları yazılmalıdır. (10 p)

DML #1 (INSERT) : Yeni kullanıcı ekle

```
INSERT INTO users (full_name, email, phone, password_hash)  
VALUES ('Furkan Köksalan', 'furkan@example.com', '+90 555 111 2233',  
'hashed_pw_demo');
```

DML #2 (INSERT) : Kullanıcı için adres ekle + ilişkilendir (user_addresses)

```

WITH u AS (
    SELECT user_id FROM users WHERE email='furkan@example.com'
),
a AS (
    INSERT INTO addresses (
        country_id, city_id, district_id, neighborhood_id,
        address_label, street, building_no, apartment_no, postal_code
    )
    VALUES (
        '1','34','3401','340101',
        'Ev', 'Bahariye Cd.', '12', '5', '34710'
    )
    RETURNING address_id
)
INSERT INTO user_addresses (user_id, address_id, is_default)
SELECT u.user_id, a.address_id, TRUE
FROM u, a;

```

DML #3 (UPDATE) : Kullanıcının primary_address_id alanını default adresiyle güncelle

```

UPDATE users u
SET primary_address_id = ua.address_id,
    updated_at = now()
FROM user_addresses ua
WHERE ua.user_id = u.user_id
    AND ua.is_default = TRUE
    AND u.email = 'furkan@example.com';

```

DML #4 (INSERT + UPDATE) : Pet oluştur, sonra kayıp ilanı oluştur (listing species_id set)

```

WITH u AS (
    SELECT user_id FROM users WHERE email='furkan@example.com'
),
p AS (
    INSERT INTO pets (owner_user_id, species_id, name, sex, age_years,
distinctive_marks)
    SELECT u.user_id, '1', 'Maviş', 'dişi', 2, 'Sağ kulağında küçük kesik var'
)

```

```

    FROM u
    RETURNING pet_id, species_id
),
addr AS (
    SELECT ua.address_id
    FROM user_addresses ua
    JOIN users u2 ON u2.user_id = ua.user_id
    WHERE u2.email='furkan@example.com'
    ORDER BY ua.is_default DESC, ua.created_at DESC
    LIMIT 1
),
l AS (
    INSERT INTO listings (
        created_by, pet_id, species_id, address_id,
        listing_type, status, title, description, event_time
    )
    SELECT u.user_id, p.pet_id, p.species_id, addr.address_id,
        'kayıp', 'aktif',
        'Kadıköy Moda'da kayıp kedi',
        'Tekir gri, çok uysal. Gören olursa lütfen yazın.',
        '2025-12-16 21:30'
    FROM u, p, addr
    RETURNING listing_id
)
INSERT INTO listing_photos (listing_id, file_name, data_url, sort_order)
SELECT l.listing_id, 'kedi1.png', 'data:image/png;base64,DEMO_BASE64_DATA', 0
FROM l;

```

DML #5 (DELETE) : İlan sil (CASCADE ile photos, conversations, participants, messages, matches de temizlenir)

```

DELETE FROM listings
WHERE title = 'Kadıköy Moda'da kayıp kedi'
AND listing_type = 'kayıp';

```

**8. Projenize ait kendi belirlediğiniz toplam 5 adet SQL sorgusu yazınız.
Sorguların genel yapısı aşağıdaki ileri seviye kriterleri mutlaka sağlamalıdır:**

- **En az üç tabloyu birleştiren (JOIN) yapıya sahip olmalı,**
- **Kümeleme (Aggregate) fonksiyonlarını (COUNT, SUM, AVG, MIN, MAX vb.) içermeli,**
- **Varsa GROUP BY ve HAVING deyimlerini kapsamalıdır.**

Her bir madde için sorgunun amacını, SQL kodunu, cevabını ve sonuç çıktısına ait ekran görüntüsünü raporunuza ekleyiniz. Proje sunum anında bu sorgular SQL ortamında gösterilecek ve açıklanacaktır. (10 p)

1. SQL - İlçe bazında aktif kayıp ilan sayısı

Amaç: İstanbul/Kadıköy gibi ilçelerde **aktif kayıp ilanları** sayısını görmek; en az 2 ilan olan ilçeleri listelemek.

```
SELECT
    c.name AS city_name,
    d.name AS district_name,
    COUNT(*) AS active_lost_count,
    MAX(l.created_at) AS last_lost_created_at
FROM listings l
JOIN addresses a ON a.address_id = l.address_id
JOIN cities c ON c.city_id = a.city_id
JOIN districts d ON d.district_id = a.district_id
WHERE lower(trim(l.listing_type)) IN ('kayıp')
    AND lower(trim(l.status)) = 'aktif'
GROUP BY c.name, d.name
HAVING COUNT(*) >= 2
ORDER BY active_lost_count DESC, last_lost_created_at DESC;
```

	city_name character varying (100) 	district_name character varying (100) 	active_lost_count bigint 	last_lost_created_at timestamp without time zone 
1	Antalya	Muratpaşa	77	2025-12-17 17:56:21.145219
2	Kocaeli	İzmit	49	2025-12-17 17:56:21.145219
3	Bursa	Nilüfer	45	2025-12-17 17:56:21.145219
4	Ankara	Çankaya	42	2025-12-17 18:09:17.656042
5	Eskişehir	Odunpazarı	40	2025-12-17 17:56:21.145219
6	İstanbul	Maltepe	36	2025-12-17 17:56:21.145219
7	İzmir	Konak	34	2025-12-17 17:56:21.145219
8	İzmir	Karşıyaka	30	2025-12-17 17:56:21.145219
9	İzmir	Bornova	29	2025-12-17 17:56:21.145219
10	İstanbul	Kadıköy	28	2025-12-17 18:04:03.256202
11	İstanbul	Üsküdar	25	2025-12-17 17:56:21.145219
12	İstanbul	Ataşehir	10	2025-12-17 17:56:21.145219
13	İstanbul	Beşiktaş	8	2025-12-17 17:56:21.145219

2. SQL Tür bazında “kayıp vs bulundu” ilan sayıları + oran (sadece aktif ilanlar)

Amaç: Türlere göre (Kedi/Köpek) aktif kayıp ve aktif bulundu ilan sayısını ve kayıp oranını görmek.

```

SELECT
    s.name AS species_name,
    COUNT(*) FILTER (WHERE lower(trim(l.listing_type)) IN ('kayıp','kayıp') AND
lower(trim(l.status))='aktif') AS active_lost,
    COUNT(*) FILTER (WHERE lower(trim(l.listing_type))='bulundu' AND
lower(trim(l.status))='aktif') AS active_found,
    ROUND(
        100.0 * COUNT(*) FILTER (WHERE lower(trim(l.listing_type)) IN ('kayıp','kayıp') AND
lower(trim(l.status))='aktif')
        / NULLIF(COUNT(*) FILTER (WHERE lower(trim(l.status))='aktif'), 0),
        2
    ) AS lost_ratio_percent
FROM listings l
JOIN species s ON s.species_id = l.species_id
JOIN addresses a ON a.address_id = l.address_id
GROUP BY s.name
HAVING COUNT(*) FILTER (WHERE lower(trim(l.status))='aktif') >= 1
ORDER BY active_lost DESC, active_found DESC;

```

	species_name character varying (80)	active_lost bigint	active_found bigint	lost_ratio_percent numeric
1	Kedi	268	240	52.76
2	Köpek	176	170	50.87

3. SQL - Kullanıcı başına aktif ilan sayısı + ilanların konum dağılımı (en az 1 aktif ilan)

Amaç: Hangi kullanıcı kaç **aktif ilan** açmış, ilanları hangi şehirde yoğunlaşmış görmek.

```

SELECT
    u.full_name,
    u.email,
    c.name AS city_name,
    COUNT(l.listing_id) AS active_listing_count,
    MIN(l.created_at) AS first_listing_at,
    MAX(l.created_at) AS last_listing_at
FROM users u
JOIN listings l ON l.created_by = u.user_id
JOIN addresses a ON a.address_id = l.address_id
JOIN cities c ON c.city_id = a.city_id
WHERE lower(trim(l.status)) = 'aktif'
GROUP BY u.full_name, u.email, c.name
HAVING COUNT(l.listing_id) >= 1
ORDER BY active_listing_count DESC, last_listing_at DESC;

```

	full_name character varying (150)	email character varying (150)	city_name character varying (100)	active_listing_count bigint	first_listing_at timestamp without time zone	last_listing_at timestamp without time zone
1	Mert Taş	mert.tas8086@gmail.com	İstanbul	15	2025-12-17 17:56:21.145219	2025-12-17 17:56:21.145219
2	Onur Kurt	onur.kurt155@gmail.com	İzmir	14	2025-12-17 17:56:21.145219	2025-12-17 17:56:21.145219
3	Ceren Güneş	ceren.gunes9432@gmail.c...	Kocaeli	14	2025-12-17 17:56:21.145219	2025-12-17 17:56:21.145219
4	Emir Yıldız	emir.yildiz5079@gmail.com	Eskişehir	14	2025-12-17 17:55:09.731843	2025-12-17 17:55:09.731843
5	Ahmet Kurt	ahmet.kurt2069@gmail.com	Antalya	14	2025-12-17 17:55:09.731843	2025-12-17 17:55:09.731843
6	Elif Çelik	elif.celik2401@gmail.com	Ankara	13	2025-12-17 17:55:09.731843	2025-12-17 17:55:09.731843
7	Aslı Yıldız	asli.yildiz2627@gmail.com	Antalya	12	2025-12-17 17:56:21.145219	2025-12-17 17:56:21.145219
8	Yusuf Aydin	yusuf.aydin3145@gmail.co...	İzmir	12	2025-12-17 17:55:09.731843	2025-12-17 17:55:09.731843
9	Onur Arslan	onur.arslan5955@gmail.com	Ankara	12	2025-12-17 17:55:09.731843	2025-12-17 17:55:09.731843
10	Elif Arslan	elif.arslan9125@gmail.com	Antalya	11	2025-12-17 17:56:21.145219	2025-12-17 17:56:21.145219
11	Berk Aksoy	berk.aksoy7538@gmail.com	Antalya	11	2025-12-17 17:56:21.145219	2025-12-17 17:56:21.145219
12	Ece Sezer	ece.sezer3817@gmail.com	İstanbul	11	2025-12-17 17:56:21.145219	2025-12-17 17:56:21.145219
13	Berk Aksoy	berk.aksoy3590@gmail.com	Antalya	11	2025-12-17 17:55:09.731843	2025-12-17 17:55:09.731843
14	Ayşe Aydin	ayse/aydin6118@gmail.com	Bursa	11	2025-12-17 17:55:09.731843	2025-12-17 17:55:09.731843
15	Kaan Şahin	kaan.sahin2985@gmail.com	İstanbul	11	2025-12-17 17:55:09.731843	2025-12-17 17:55:09.731843
16	Ahmet Aksoy	ahmet.aksoy9367@gmail.c...	Antalya	11	2025-12-17 17:55:09.731843	2025-12-17 17:55:09.731843

Total rows: 141 | Query complete 00:00:00.035

4. SQL - Şehir + Tür bazında aktif kayıp ilan sayısı (en az 2 olanlar)

Amaç: Hangi şehirde hangi tür için kaç tane aktif kayıp ilanı var görmek (sadece 2 ve üzeri olanları listelemek).

```
SELECT
    c.name AS city_name,
    s.name AS species_name,
    COUNT(l.listing_id) AS active_lost_count,
    MAX(l.created_at) AS last_created_at
FROM listings l
JOIN addresses a ON a.address_id = l.address_id
JOIN cities c ON c.city_id = a.city_id
JOIN species s ON s.species_id = l.species_id
WHERE lower(trim(l.listing_type)) IN ('kayıp')
    AND lower(trim(l.status)) = 'aktif'
GROUP BY c.name, s.name
HAVING COUNT(l.listing_id) >= 2
ORDER BY active_lost_count DESC, last_created_at DESC;
```

city_name character varying (100) 	species_name character varying (80) 	active_lost_count bigint 	last_created_at timestamp without time zone 
İstanbul	Kedi	64	2025-12-17 17:56:21.145219
İzmir	Kedi	54	2025-12-17 17:56:21.145219
Antalya	Kedi	44	2025-12-17 17:56:21.145219
İzmir	Köpek	39	2025-12-17 17:56:21.145219
İstanbul	Köpek	38	2025-12-17 17:56:21.145219
Antalya	Köpek	33	2025-12-17 17:56:21.145219
Eskişehir	Kedi	30	2025-12-17 17:56:21.145219
Bursa	Kedi	30	2025-12-17 17:56:21.145219
Kocaeli	Kedi	28	2025-12-17 17:56:21.145219
Kocaeli	Köpek	21	2025-12-17 17:56:21.145219
Ankara	Köpek	20	2025-12-17 18:06:13.349309
Ankara	Kedi	18	2025-12-17 17:56:21.145219
Bursa	Köpek	15	2025-12-17 17:56:21.145219
Eskişehir	Köpek	10	2025-12-17 17:55:09.731843

5. SQL - Kayıp ilanlar için aynı şehirdeki aktif “bulundu” ilan sayısı (tür eşleşmeli) + HAVING ile filtre

Amaç: Bir kayıp ilanı için (aynı şehir + aynı tür) kaç tane potansiyel bulundu ilanı var görmek. (Sunumda “eşleşme mantığı” diye anlatırsın.)

```

SELECT
    lo.listing_id AS lost_listing_id,
    lo.title AS lost_title,
    c.name AS city_name,
    s.name AS species_name,
    COUNT(f.listing_id) AS candidate_found_count,
    MAX(f.created_at) AS last_found_created_at
FROM listings lo
JOIN addresses la ON la.address_id = lo.address_id
JOIN cities c ON c.city_id = la.city_id
JOIN species s ON s.species_id = lo.species_id
LEFT JOIN listings f
    ON lower(trim(f.listing_type))='bulundu'
    AND lower(trim(f.status))='aktif'
    AND f.species_id = lo.species_id
LEFT JOIN addresses fa
    ON fa.address_id = f.address_id
    AND fa.city_id = la.city_id
WHERE lower(trim(lo.listing_type)) IN ('kayıp')
    AND lower(trim(lo.status))='aktif'
GROUP BY lo.listing_id, lo.title, c.name, s.name
HAVING COUNT(f.listing_id) >= 1
ORDER BY candidate_found_count DESC, last_found_created_at DESC;

```

lost_listing_id	lost_title	city_name	species_name	candidate_found_count	last_found_created_at
uuid	character varying (200)	character varying (100)	character varying (80)	bigint	timestamp without time zone
ffb09b22-9a98-4120-907f-d215fae1b8...	Kayıp evcil hayvan - yardım	İstanbul	Kedi	240	2025-12-17 18:03:20.940188
00be196b-516a-4cb8-b0ab-03b6389d...	Kayıp kedim aranıyor	İzmir	Kedi	240	2025-12-17 18:03:20.940188
42f3426d-3e8d-481e-9962-1f4904ba4...	Kayıp evcil hayvan - yardım	Ankara	Kedi	240	2025-12-17 18:03:20.940188
4367ffff-0b1cd-4a55-a26b-4f251b9f03...	Kayıp köpeğim kayboldu	Antalya	Kedi	240	2025-12-17 18:03:20.940188
442182fe-7b5b-415d-a7ea-5c8ad4965...	Evcil hayvanım kayıp	İstanbul	Kedi	240	2025-12-17 18:03:20.940188
4617d7a2-1494-4a52-8448-f35118b8e...	Kayıp evcil hayvan - yardım	Bursa	Kedi	240	2025-12-17 18:03:20.940188
469d34ce-61d2-4483-83af-eaa04def7...	Gören olursa lütfen iletişime ge...	Antalya	Kedi	240	2025-12-17 18:03:20.940188
47a3bf77-0024-4b00-9cc0-f1ca1f1b37...	Kayıp evcil hayvan - yardım	İstanbul	Kedi	240	2025-12-17 18:03:20.940188
47b45acd-15be-4077-85f3-25d44d465...	Kayıp kedim aranıyor	İzmir	Kedi	240	2025-12-17 18:03:20.940188
4810b44e-b446-4ba5-b41b-96be4a21...	Gören olursa lütfen iletişime ge...	İstanbul	Kedi	240	2025-12-17 18:03:20.940188
48ced64e-ebca-4858-8179-e5598b3e3...	Kayıp kedim aranıyor	Antalya	Kedi	240	2025-12-17 18:03:20.940188
49b70b3f-76c8-4258-98cf-4134e7cac...	Evcil hayvanım kayıp	Antalya	Kedi	240	2025-12-17 18:03:20.940188
49e26558-d4ee-4a53-a97c-c942efd11...	Gören olursa lütfen iletişime ge...	İstanbul	Kedi	240	2025-12-17 18:03:20.940188
4a88bc88-7721-491c-a78a-2c154923...	Gören olursa lütfen iletişime ge...	İstanbul	Kedi	240	2025-12-17 18:03:20.940188
4dff3af0-2ead-41fe-9b38-7610c505f7f	Kayıp evcil hayvan - yardım	Bursa	Kedi	240	2025-12-17 18:03:20.940188

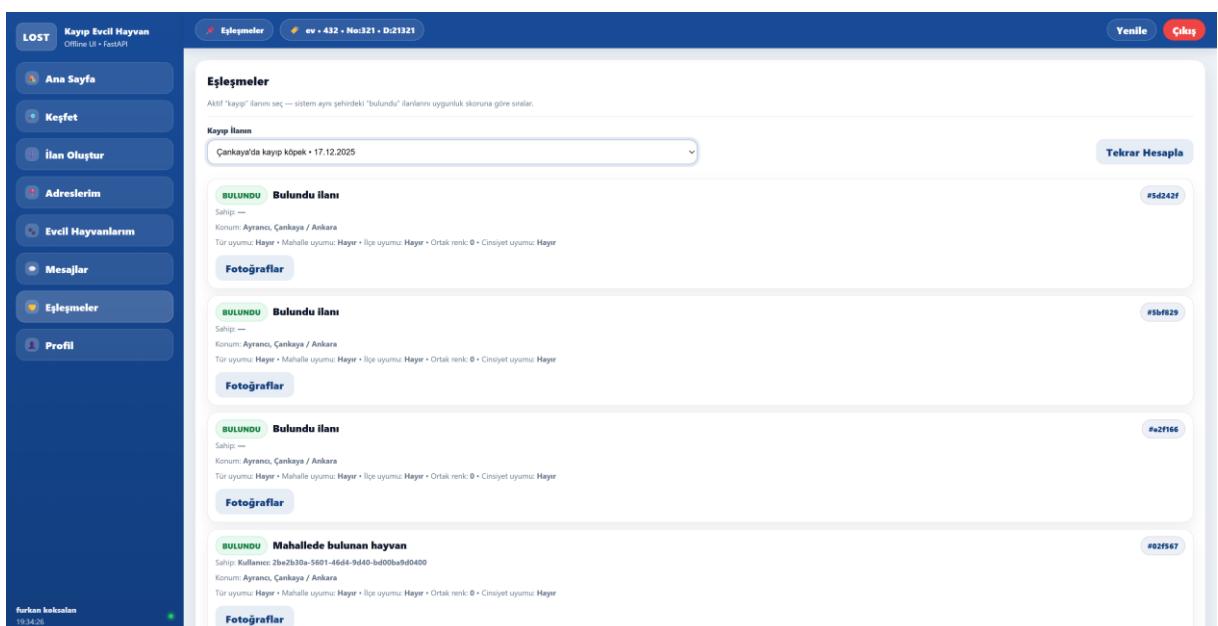
1 rows: 444 Query complete 00:00:00.168

9. Veri tabanı bağlama ve uygulama geliştirme aşamalarınızı kısaca açıklayarak, kullanıcı ara yüz ekranından bir örnek veriniz. Ve geliştirdiğiniz ara yüzü anlatınız. (10 p)

Projede öncelikle **PostgreSQL** kullanılarak ilişkisel veri tabanı tasarlanmış ve tablolar arasında **foreign key** ilişkileri kurulmuştur. Daha sonra **FastAPI** kullanılarak backend geliştirilmiştir, veri tabanı bağlantısı **SQLAlchemy** aracılığıyla sağlanmıştır. Backend tarafında oluşturulan REST API endpoint'leri ile veri ekleme, silme, listeleme ve güncelleme işlemleri gerçekleştirilmiştir.

Frontend tarafında **React** kullanılarak kullanıcı arayüzü geliştirilmiştir. Kullanıcıdan alınan veriler (adres ekleme, evcil hayvan ekleme, ilan oluşturma vb.) JSON formatında backend API'ye gönderilmiş, backend bu verileri veri tabanına kaydetmiş ve sonuçları tekrar arayüze ilemiştir. Böylece frontend–backend–veri tabanı entegrasyonu sağlanmıştır.

Eşleşme Ekranı



Evcil Hayvan Ekleme Ekranı

Evcil Hayvan Ekle

Evcil hayvanlar DİB de tutulur.

Tür: Pamuk

Cinsiyet: Erkek

Büyümeye: 2

Mikroçip No (ops.): Ayrt Edici Özellik (ops.): Boynunda benek...

Renkler (ops.): Alacalı, Beyaz, Beyaz-Kahverengi, Beyaz-Siyah, Çikolata, Fawn, Gri, Kahverengi, Krem, Sarı, Siyah, Tekir, Tekir Gri, Tekir Kahverengi, Tekir Sarı, Turuncu

Evcili Hayvani Kaydet!

Adres Ekleme Ekranı

Adres Ekle

Adres Etiketi: Ev / İş / Anne ev...

Ülke: Seç

Şehir: Seç

İlçe: Seç

Mahalle: Seç

Sokak: Seç

Bina No (ops.): 12

Daire No (ops.): 5

Posta Kodu (ops.): 34000

Birinci adres yap!

Adreslerim

Birinci adres ist bardak etiket olarak görünür.

sokak
Ülke: Türkiye • Şehir: --- • İlçe: --- • Mahalle: ---
Birincil Yap **Sil**

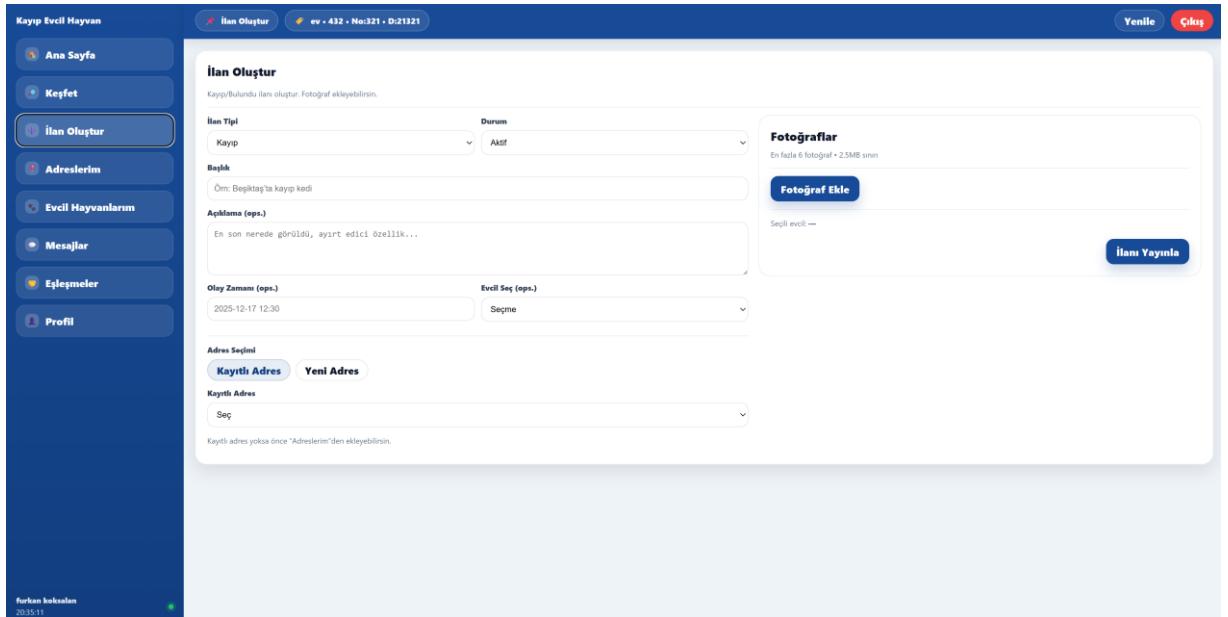
Ev
Ülke: Türkiye • Şehir: --- • İlçe: --- • Mahalle: ---
Birincil Yap **Sil**

Kedi
Ülke: Türkiye • Şehir: --- • İlçe: --- • Mahalle: ---
Birincil Yap **Sil**

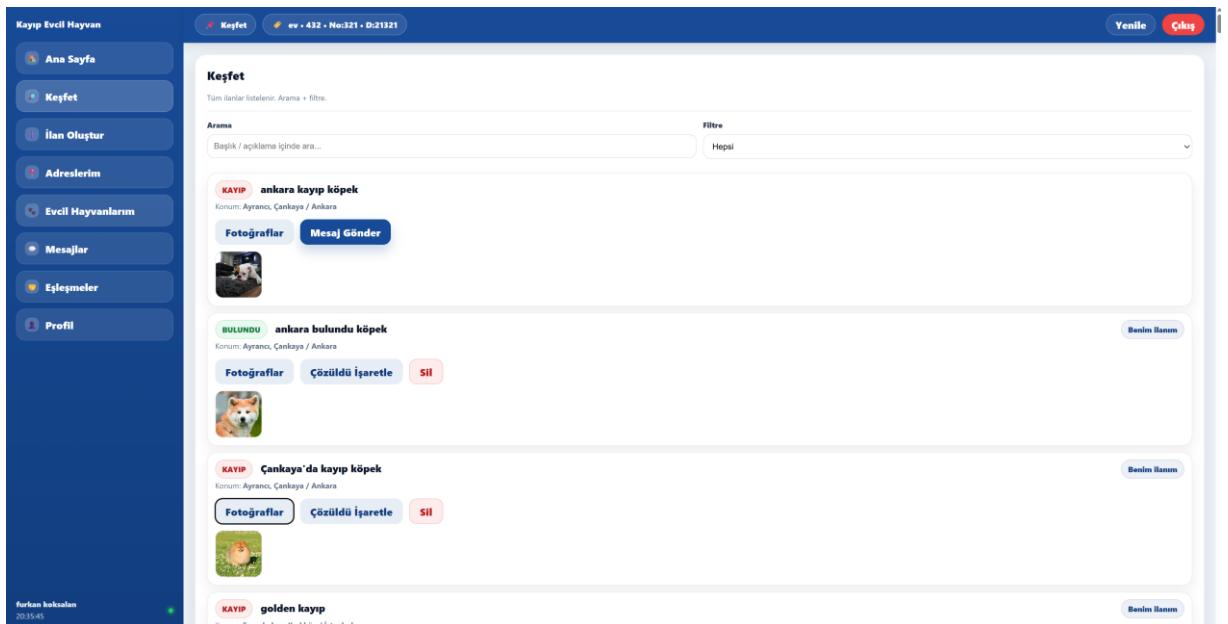
ev
Ülke: Türkiye • Şehir: --- • İlçe: --- • Mahalle: ---
Birincil Yap **Sil**

ev - 432 - No:321 - D:21321 **Birincil Yap** **Sil**

İlan oluşturma ekranı



Keşfet Ekranı



10. VTYs sistemlerinde Transaction nedir açıklayınız ve çalışmanızdan bir Transaction örneği veriniz. (10 p)

Transaction, veritabanında bir bütün olarak ele alınan ve tamamı başarılı olursa kalıcı hale gelen, aksi durumda tamamen geri alınan (rollback) işlemler grubudur.

Bir transaction şu özellikleri sağlamak zorundadır :

1. Atomicity (Atomiklik)

Transaction içindeki işlemler ya tamamı gerçekleşir ya da hiçbirini gerçekleşmez.

2. Consistency (Tutarlılık)

Transaction sonunda veritabanı kurallara (PK, FK, CHECK, UNIQUE) uygun ve tutarlı durumda kalır.

3. Isolation (İzolasyon)

Aynı anda çalışan transaction'lar birbirinin ara sonuçlarını görmez.

4. Durability (Kalıcılık)

Transaction commit edildikten sonra yapılan değişiklikler sistem çökse bile kaybolmaz.

Çalışmamızdan Transaction Örneği

Projemizde kullanıcıya ait adres ekleme ve bu adresin varsayılan (default) adres olarak atanması işlemi bir transaction olarak ele alınmalıdır. Çünkü adres eklenip kullanıcıya bağlanamazsa veritabanı tutarsız hale gelir.

Senaryo

1. Yeni adres eklenir

2. Adres kullanıcıya bağlanır (user_addresses)

3. Kullanıcının primary_address_id alanı güncellenir → Bu işlemler tek bir transaction içinde yapılır.

```
BEGIN;
```

```
-- 1) Yeni adres ekle
```

```
INSERT INTO addresses (
    country_id, city_id, district_id, neighborhood_id,
    address_label, street, building_no, apartment_no, postal_code
)
VALUES (
    '1','34','3401','340101',
    'Ev', 'Bahariye Cd.', '12', '5', '34710'
);
```

```
-- 2) Adresi kullanıcıya bağla
```

```

INSERT INTO user_addresses (user_id, address_id, is_default)
SELECT u.user_id, a.address_id, TRUE
FROM users u
JOIN addresses a ON a.address_label='Ev'
WHERE u.email='furkan@example.com';

-- 3) Kullanıcının varsayılan adresini güncelle
UPDATE users
SET primary_address_id = (
    SELECT address_id
    FROM user_addresses ua
    WHERE ua.user_id = users.user_id
        AND ua.is_default = TRUE
    LIMIT 1
),
updated_at = now()
WHERE email='furkan@example.com';

COMMIT;

```

11. View nedir açıklayınız ve bir adet view, bir adet saklı yordam (Stored Procedure) ifadesine ait SQL deyimlerinin sorgusunu ve cevabını yazınız. (10 p)

View, veritabanında fiziksel olarak veri tutmayan, bir veya birden fazla tablodan elde edilen sonuçların sanal bir tablo gibi sunulmasını sağlayan yapılardır.

View'lerin temel amaçları:

- Karmaşık sorguları basitleştirmek
- Kullanıcıya sadece gerekli alanları göstermek
- Güvenlik sağlamak (tablonun tamamını açmadan veri sunmak)
- Sık kullanılan sorguları standart hale getirmek

View'ler çağrıldığında, arka planda tanımlandıkları SELECT sorgusu çalışır.

Çalışmamızdan View Örneği

Amaç

Kayıp evcil hayvan ilanlarını, adres ve tür bilgisiyile birlikte listelemek

CREATE OR REPLACE VIEW vw_active_lost_listings AS

SELECT

```

l.listing_id,
l.title,
l.description,
l.event_time,
l.created_at,
s.name AS species_name,
c.name AS city_name,
d.name AS district_name,
n.name AS neighborhood_name

```

FROM listings l

JOIN species s ON s.species_id = l.species_id

JOIN addresses a ON a.address_id = l.address_id

JOIN cities c ON c.city_id = a.city_id

JOIN districts d ON d.district_id = a.district_id

JOIN neighborhoods n ON n.neighborhood_id = a.neighborhood_id

WHERE l.listing_type = 'kayıp'

AND l.status = 'aktif';

listing_id	title character varying (200)	description character varying (4000)	event_time character varying (60)	created_at timestamp without time zone	species_name character varying (60)	city_name character varying (100)	district_name character varying (100)	neighborhood_name character varying (120)
3e3a4d9-41d7-4ec4-b484-93ca04800...	sadsa	sadsa	2025-12-17 16:13:24.119463	2025-12-17 16:13:24.119463	Köpek	Ankara	Cankaya	Ayancı
c2c04309-41d5-4b84-93c5-98d988449...	asdsda	sadsa	[null]	2025-12-17 16:13:51.444191	Köpek	Ankara	Cankaya	Ayancı
f51a5f9e-4242-45fe-b7f4-901b6101ba...	adws	asdsda	[null]	2025-12-17 16:17:45.559915	Köpek	Ankara	Cankaya	Ayancı
e2244ff0-4667-4bcb-946d-e291961...	asdd	[null]	[null]	2025-12-17 16:19:41.800703	Köpek	Ankara	Cankaya	Ayancı
e23999fb-55de-42d0-b75b-77979467...	Kayıp evcil hayvan - yardım	Fotoğrafı ile hayvana benzırsız iletişime geçin.	2025-11-28 19:06	2025-12-17 17:51:18.563061	Köpek	İstanbul	Ataşehir	İçerenköy
e3bc484b-4b1b-4bcb-946d-e291961...	Kayıp Kayıp İani	İstanbullu kişi hizmeti, konakları. Yalnızken dikkat olun. Lütfen yanlış haber paylaşın.	2025-12-08 08:31	2025-12-17 17:51:18.563061	Kedi	İstanbul	Kadıköy	Caddebostan
9d79eb9b-186a-441b-a309-c0371199...	Kayıp Kickeyim kaybındı	En son Alışveriş sırasında görüldü. Gören okurta lütfen haber versin. Lütfen yanlış haber paylaşın.	2025-12-12 21:41	2025-12-17 17:51:18.563061	Kedi	Izmir	Konak	Altıncık
3ef7e2b-2824-4eeb-b22b-1798b4092...	Açılıp Kayıp İani	İstanbullu kişi hizmeti, konakları. Yalnızken dikkat olun. Sahibi tarafından çok sevilen bir hayvan. Kapılı stanlara girmem olabilir.	2025-12-06 02:51	2025-12-17 17:51:18.563061	Köpek	Kocaeli	İzmit	Yahya Kaplan
69993ab5-7327-402c-9bdc-4519c53b...	Kayıp evcil hayvan - yardım	İstanbullu kişi oyuncu, konakları. Lütfen yanlış haber paylaşın.	2025-11-28 07:21	2025-12-17 17:51:18.563061	Kedi	Eskişehir	Ödemişçayır	Alanoğlu
b9732649-47c0-412a-94d7-2201c61...	Evcil hayvanın Kayıp	Fotoğrafı ile hayvana benzırsız iletişime geçin.	2025-12-14 07:41	2025-12-17 17:51:18.563061	Kedi	İstanbul	Ataşehir	Atatürk Mahallesı
5094baef-8c09-412c-95b5-45b6617b...	Açılıp Kayıp İani	Gören ya da bulan okurta lütfen mesaj atın. Arf geri dönüşü bekleyin.	2025-12-01 09:31	2025-12-17 17:51:18.563061	Kedi	İstanbul	Osaköyler	Çengelköy
c1bd484b-4ba9-4423-9907-1f94bedc1...	Kayıp Kickeyim kaybındı	Fotoğrafı ile hayvana benzırsız iletişime geçin.	2025-12-02 08:21	2025-12-17 17:51:18.563061	Kedi	Eskişehir	Ödemişçayır	Alanoğlu
a5ca8255-54d7-4995-9319-ab949107...	Açılıp Kayıp İani	Etrafa dolanınan seyyar ama eve geri dönmeli. Cesurcalar çok üzgün.	2025-12-01 08:41	2025-12-17 17:51:18.563061	Kedi	Izmir	Bornova	Kazaklık
ec409910-1a60-4e48-9d82-3d8a040...	Açılıp Kayıp İani	Etrafa dolanınan seyyar ama eve geri dönmeli. Odul verilecektir.	2025-11-29 06:06	2025-12-17 17:51:18.563061	Kedi	Izmir	Bornova	Kazaklık
29657d6e-fcde-4e7c-8fce-830116fbuff...	Gören okurta lütfen iletişime ge...	Seslenince ligili veya şerpiye olabilir. Teme Padi. Çocuklar çok üzgün.	2025-12-11 10:36	2025-12-17 17:51:18.563061	Köpek	Kocaeli	İzmit	Allâhiye
6af1466-5007-4e56-9096-24594243...	Evcil hayvanın Kayıp	İstanbullu kişi çekingen, konakları. Yaklaşıkken dikkat olun.	2025-12-12 12:31	2025-12-17 17:51:18.563061	Köpek	İzmir	Bornova	Ezzone
18629636-83d4-4638-9870-22118882...	Gören okurta lütfen iletişime ge...	İstanbullu kişi çekingen, konakları. Yaklaşıkken dikkat olun.	2025-11-10 01:41	2025-12-17 17:51:18.563061	Kedi	İstanbul	Kadıköy	Caddebostan
27237330-83d4-4638-9870-22118882...	Kayıp evcil hayvan - yardım	Seslenince ligili veya şerpiye olabilir. Teme Padi. Çocuklar çok üzgün.	2025-11-10 01:36	2025-12-17 17:51:18.563061	Kedi	Izmir	Konak	Altıncık
02240949-8029-4454-91f1-86014429...	Açılıp Kayıp İani	Seçkinlikle yürüyor. Oyun oynuyor. Çırılçıplak. Lütfen yanlış haber paylaşın.	2025-11-10 01:41	2025-12-17 17:51:18.563061	Kedi	İstanbul	Ataşehir	Atatürk Mahallesı
65647802-c17-a404-419-2074820...	Açılıp Kayıp İani	Çok sızlı, yılançca yatağındır. Oda yerinde. Lütfen yanlış haber paylaşın.	2025-11-10 15:12:51	2025-12-17 17:51:18.563061	Kedi	Kocaeli	Ümraniye	Allâhiye
7f104f5a-2a9f-432e-9f96-cb509999...	Kayıp Kickeyim kaybındı	Seslenmesi ligili veya şerpiye olabilir. Gece saatlerinde data aktif olabilir.	2025-11-10 01:41	2025-12-17 17:51:18.563061	Kedi	İstanbul	Kadıköy	Caddebostan
45051913-306-4420-9210-55ea1183...	Kayıp evcil hayvan - yardım	Mahallede 1-12 eseriinde görtüğün olsun. Lütfen yanlış haber paylaşın.	2025-12-04 04:41	2025-12-17 17:51:18.563061	Kedi	Izmir	Bornova	Ezzone
3008220-052-4314-9f8a-444178a...	Gören okurta lütfen iletişime ge...	Fotoğrafı ile hayvana benzırsız iletişime geçin. Gece saatlerinde data aktif olabilir.	2025-11-09 01:38	2025-12-17 17:51:18.563061	Kedi	Eskişehir	Ödemişçayır	Alanoğlu
4bc49191-01a6-4234-9441-8441780...	Eçili hayvanın Kayıp	Etrafa dolanınan seyyar ama eve geri dönmeli. Lütfen yanlış haber paylaşın.	2025-11-18 08:06	2025-12-17 17:51:18.563061	Köpek	Kocaeli	Ümraniye	Altıncık
e54326c3-48f9-4023-9fcf-803e48073...	Kayıp evcil hayvan - yardım	Etrafa dolanınan seyyar ama eve geri dönmeli. Lütfen yanlış haber paylaşın. Gece saatlerinde data aktif olabilir.	2025-11-09 21:45	2025-12-17 17:53:19.781943	Köpek	Ankara	Cankaya	Kozluk

Stored Procedure, veritabanında saklanan ve parametre alabilen, tekrar tekrar çalıştırılabilen SQL komut bloklarıdır.

Stored Procedure'lerin avantajları:

- İş mantığını veritabanına taşıır
- Tekrarlı kodu azaltır
- Performans ve güvenlik sağlar

- Transaction yönetimi yapılabilir

Çalışmamızdan Stored Procedure Örneği

Amaç

Kullanıcıya ait tüm aktif ilanları listelemek

```
CREATE OR REPLACE FUNCTION get_user_active_listings(p_user_id UUID)
RETURNS TABLE (
    listing_id UUID,
    title VARCHAR,
    listing_type VARCHAR,
    status VARCHAR,
    created_at TIMESTAMP
)
LANGUAGE plpgsql
AS $$

BEGIN
    RETURN QUERY
    SELECT
        l.listing_id,
        l.title,
        l.listing_type,
        l.status,
        l.created_at
    FROM listings l
    WHERE l.created_by = p_user_id
        AND l.status = 'aktif';
END;
$$;

SELECT *
FROM get_user_active_listings(
    (SELECT user_id FROM users where email = 'furkankoksalan@marun.edu.tr' )
);
```

```
80 |     SELECT *  
81 |     FROM get_user_active_listings(  
82 |         (SELECT user_id FROM users WHERE email = 'furkankoksalan@marun.edu.tr' )  
83 |     );  
84 |
```

Data Output Messages Notifications

The screenshot shows a PostgreSQL database interface with a SQL query editor at the top and a table viewer below. The table contains six rows of data with columns: listing_id, title, listing_type, status, and created_at.

listing_id	title	listing_type	status	created_at
uuid	character varying	character varying	character varying	timestamp without time zone
c2c04308-4195-4d84-98c5-98ad86449...	aasdsa	kayıp	aktif	2025-12-17 16:13:51.443191
f51ad3fe-42d2-454e-b7f4-601bb11dba...	adwa	kayıp	aktif	2025-12-17 16:17:46.559915
78deb0f1-9112-4719-8d2c-a65efa84b...	kedi	bulundu	aktif	2025-12-17 17:30:41.224412
6791257d-515d-47e5-953a-873898b12...	Kedi bulundu	bulundu	aktif	2025-12-17 18:03:20.940188
9f2ecba0-b237-4f1e-b14b-b983cc44c4...	golden kayıp	kayıp	aktif	2025-12-17 18:04:03.256202
9db002e8-c33f-42f7-ab8b-170e7b5b8...	Çankaya'da kayıp köpek	kayıp	aktif	2025-12-17 18:06:13.349309
f190dacb-3aba-4fe7-a7a9-24b81e0c68...	ankara bulundu köpek	bulundu	aktif	2025-12-17 18:07:05.919589