COMP 306 – Mini Sahibinden

Çağan Dayıoğlu – 86863

Çağlar Dayıoğlu – 86518


```
spring.application.name=MiniSahibinden

# ========================================
# RELATIONAL SQL DATABASE CONFIGURATION (MySQL)
# ========================================
# Database Connection - Connect directly to MiniSahibinden database
spring.datasource.url=jdbc:mysql://localhost:3306/MiniSahibinden?useSSL=false&serverTimezone=UTC&allowPublicKeyRetrieval=true
spring.datasource.username=root
spring.datasource.password=Enter your mysql password here
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

# JPA/Hibernate Settings
# Explicitly set MySQL dialect for SQL generation
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLDialect
# Disable Hibernate DDL - we use schema.sql instead
spring.jpa.hibernate.ddl-auto=none
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.format_sql=true
# Use physical naming strategy to keep table/column names as declared
spring.jpa.hibernate.naming.physical-strategy=org.hibernate.boot.model.naming.PhysicalNamingStrategyStandardImpl

# Enable SQL initialization only on first run (embedded databases only)
# Change to 'always' if you need to reset the database
spring.sql.init.mode=always

server.error.whitelabel.enabled=false
```

Important note:

Before running the code enter your password here.
spring.datasource.password=Enter your mysql password here

After runnig the code for the first time database will be initialized.

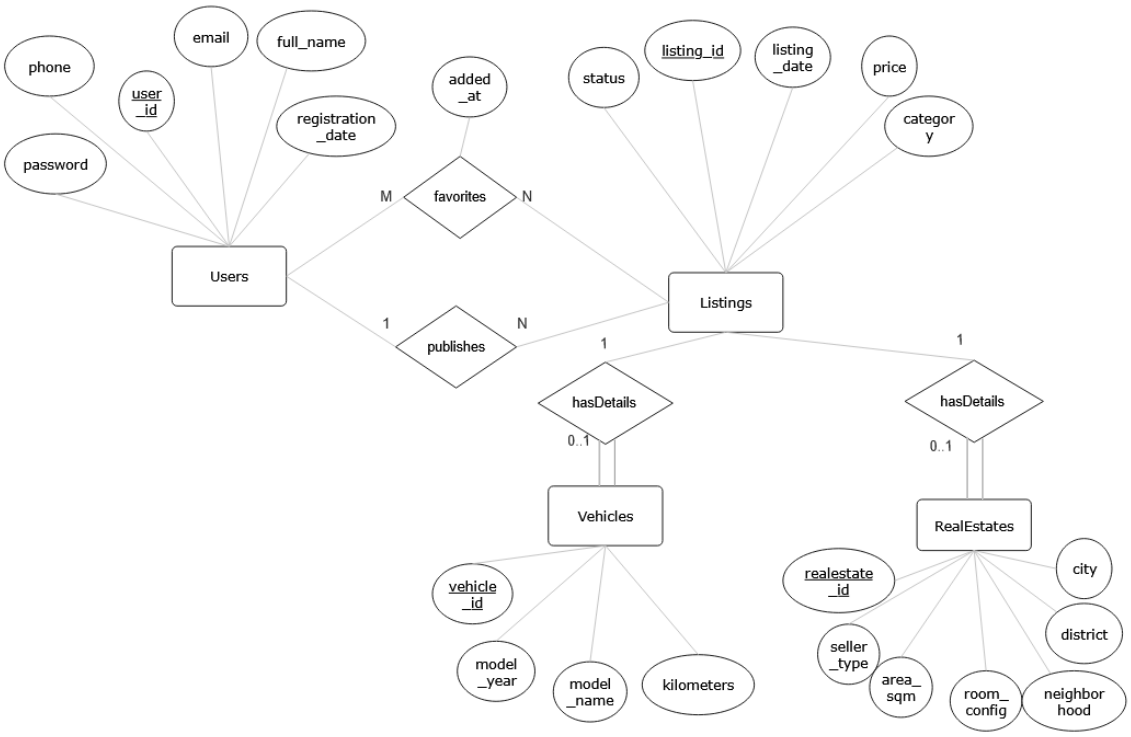
spring.sql.init.mode=always

Change ‘always’ to ‘never’

Project Description

We created a second-hand listing platform such that users can browse, save, and publish listings for vehicles and real estate.

ER Diagram



Relational Database Design

```
CREATE DATABASE IF NOT EXISTS MiniSahibinden;
USE MiniSahibinden;

DROP TABLE IF EXISTS car;
DROP TABLE IF EXISTS house;
DROP TABLE IF EXISTS Favorites;
DROP TABLE IF EXISTS Vehicles;
DROP TABLE IF EXISTS RealEstate;
DROP TABLE IF EXISTS Listings;
DROP TABLE IF EXISTS Users;

CREATE TABLE Users (
    user_id INT AUTO_INCREMENT PRIMARY KEY,
```

```
    full_name VARCHAR(100) NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    phone VARCHAR(20),
    password VARCHAR(255) NOT NULL,
    registration_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE Listings (
    listing_id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT NOT NULL,
    price DECIMAL(15, 2) NOT NULL,
    listing_date DATE NOT NULL,
    category ENUM('Vehicle', 'RealEstate') NOT NULL,
    status ENUM('Active', 'Sold', 'Deleted') DEFAULT 'Active',
    FOREIGN KEY (user_id) REFERENCES Users(user_id) ON DELETE CASCADE
);


CREATE TABLE Vehicles (
    vehicle_id INT AUTO_INCREMENT PRIMARY KEY,
    listing_id INT UNIQUE NOT NULL,
    model_year INT NOT NULL,
    model_name VARCHAR(150) NOT NULL,
    kilometers INT NOT NULL,
    FOREIGN KEY (listing_id) REFERENCES Listings(listing_id) ON DELETE CASCADE
);

CREATE TABLE RealEstate (
    real_estate_id INT AUTO_INCREMENT PRIMARY KEY,
    listing_id INT UNIQUE NOT NULL,
    seller_type VARCHAR(50),
    area_sqm INT NOT NULL,
```

```
    room_config VARCHAR(20),
    city VARCHAR(50) NOT NULL,
    district VARCHAR(50),
    neighborhood VARCHAR(100),
    FOREIGN KEY (listing_id) REFERENCES Listings(listing_id) ON DELETE CASCADE
);

CREATE TABLE Favorites (
    user_id INT NOT NULL,
    listing_id INT NOT NULL,
    added_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY (user_id, listing_id),
    FOREIGN KEY (user_id) REFERENCES Users(user_id) ON DELETE CASCADE,
    FOREIGN KEY (listing_id) REFERENCES Listings(listing_id) ON DELETE CASCADE
);
```

Data Sources

Users: For users we used an external library called java faker to create users' names, email, phone numbers and passwords for their accounts. There are 50 tuples in our Users table.

Vehicles: https://www.kaggle.com/datasets/halimkaya/turkey-car-market

We used a dataset from Kaggle. There are 1000 tuples in our Vehicles table.

Real estates: https://www.kaggle.com/datasets/emrekaradag/real-estate-prices-in-turkey-2025

We used a dataset from Kaggle. There are 1000 tuples in our Real Estate table.

Listings: Listings table has 2000 tuples. 1000 tuples with vehicles, 1000 tuples with real estate.

Advanced SQL Queries

Get a good vehicle deal

```
SELECT v.*
    FROM Vehicles v
    JOIN Listings l ON l.listing_id = v.listing_id
    WHERE l.status = 'Active'
```

```
    AND l.price < (
        SELECT AVG(l2.price)
        FROM Vehicles v2
        JOIN Listings l2 ON l2.listing_id = v2.listing_id
        WHERE l2.status = 'Active'
          AND v2.model_year = v.model_year
```

Get a good real estate deal

```
SELECT r.*, l.price
FROM RealEstate r
JOIN Listings l ON r.listing_id = l.listing_id
WHERE l.status = 'Active'
  AND l.price = (
      SELECT MIN(l2.price)
      FROM RealEstate r2
      JOIN Listings l2 ON r2.listing_id = l2.listing_id
      WHERE r2.city = r.city
        AND l2.status = 'Active'
  );
```

Vehicles from active sellers (users with multiple listings)

```
SELECT v.* FROM Vehicles v
        INNER JOIN Listings l ON v.listing_id = l.listing_id
        INNER JOIN Users u ON l.user_id = u.user_id
        WHERE l.status = 'Active' AND u.user_id IN (
```

```
    SELECT user_id FROM Listings WHERE status = 'Active' GROUP BY user_id
HAVING COUNT(*) >= :minListingCount

    ) ORDER BY l.price DESC
```

Vehicles above average price

```
SELECT v.*

FROM Vehicles v

    INNER JOIN Listings l ON v.listing_id = l.listing_id

    WHERE l.status = 'Active' AND l.price > (

    SELECT AVG(price) FROM Listings WHERE category = 'Vehicle' AND status =
'Active'

    ) ORDER BY l.price DESC
```

Vehicles by price range and seller name

```
SELECT v.*

FROM Vehicles v

    INNER JOIN Listings l ON v.listing_id = l.listing_id

    INNER JOIN Users u ON l.user_id = u.user_id

    WHERE l.status = 'Active'

    AND l.price BETWEEN :minPrice AND :maxPrice

    AND u.full_name LIKE CONCAT('%', :userNamePattern, '%')

    ORDER BY l.price DESC
```

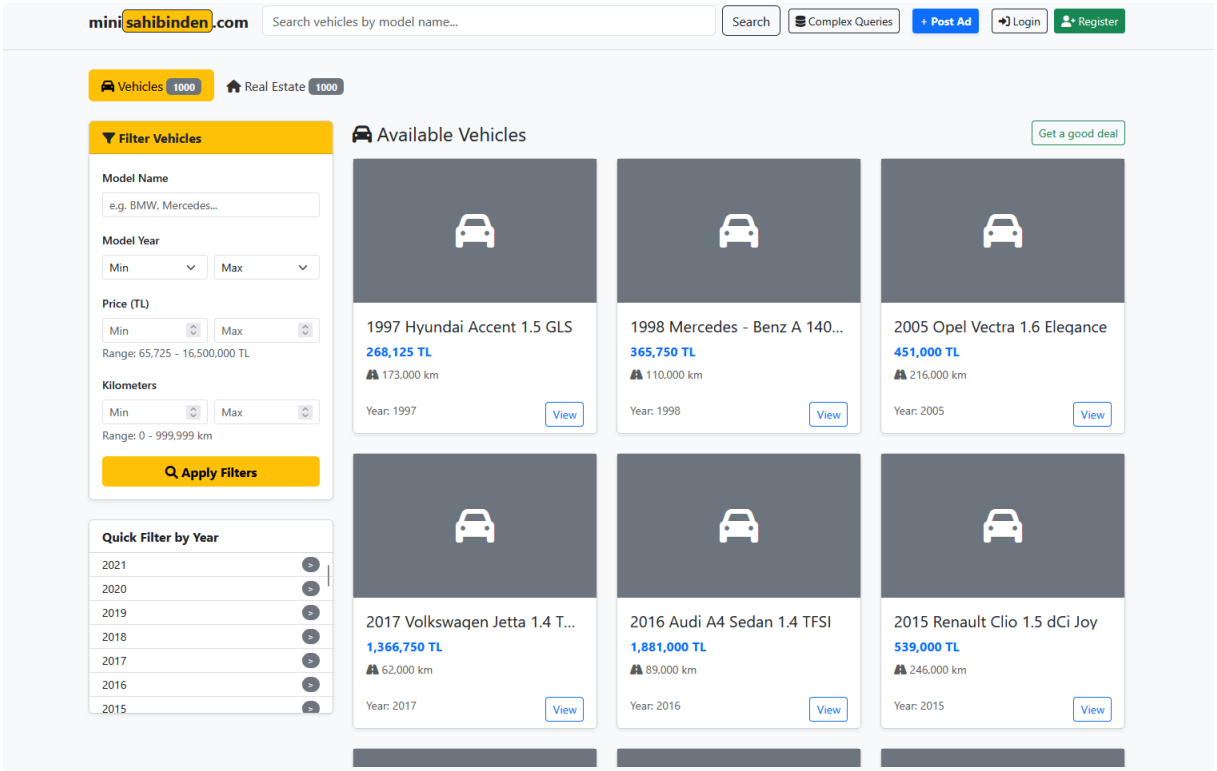
Top N vehicles per year by highest price

```
SELECT v.*

FROM (
```

```
SELECT v.*, l.price, l.status,  ROW_NUMBER() OVER (PARTITION BY
v.model_year ORDER BY l.price DESC) as rn
FROM Vehicles v
INNER JOIN Listings l ON v.listing_id = l.listing_id
WHERE l.status = 'Active'
) ranked
INNER JOIN Vehicles v ON ranked.listing_id = v.listing_id
WHERE ranked.rn <= :topN
ORDER BY ranked.model_year DESC, ranked.price DESC
```

Screenshots



My Posted Ads

Title	Price	Category	Date Posted	Actions
Bursa, Mudanya	8,480,000 TL	RealEstate	2025-05-24	View Delete
Diyarbakir, KayapÃ±nar	5,195,000 TL	RealEstate	2025-05-24	View Delete
Osmaniye, DÃ¼ziÃ§i	2,050,000 TL	RealEstate	2025-05-24	View Delete
Istanbul, BaÅŸakÅŸehir	1,750,000 TL	RealEstate	2025-05-23	View Delete
Samsun, Ã°lkadÃ±m	1,650,000 TL	RealEstate	2025-05-23	View Delete
Diyarbakir, KayapÃ±nar	9,250,000 TL	RealEstate	2025-05-21	View Delete
Ardahan, Merkez	2,400,000 TL	RealEstate	2025-05-21	View Delete
Van, Ã°pekyolu	5,250,000 TL	RealEstate	2025-05-20	View Delete
Usak, Merkez	3,750,000 TL	RealEstate	2025-05-19	View Delete
Usak, Merkez	4,300,000 TL	RealEstate	2025-05-19	View Delete
Afyonkarahisar, Merkez	1,950,000 TL	RealEstate	2025-05-17	View Delete
Van, Ã°pekyolu	6,650,000 TL	RealEstate	2025-05-17	View Delete
Diyarbakir, BaÄŸlar	3,225,000 TL	RealEstate	2025-05-17	View Delete
Usak, Merkez	2,800,000 TL	RealEstate	2025-05-17	View Delete
Erzurum, Yakutiye	3,300,000 TL	RealEstate	2025-05-16	View Delete
2016 Hyundai i20 1.4 MPI Elite	915,200 TL	Vehicle	2025-05-15	View Delete
2017 Renault Megane 1.5 dCi Touch	1,045,000 TL	Vehicle	2025-05-15	View Delete

My Favorite Ads

Title	Price	Category	Date Posted	Actions
Bingol, Solhan	2,300,000 TL	RealEstate	2025-05-25	View Remove
Samsun, Atakum	3,690,000 TL	RealEstate	2025-05-25	View Remove
1998 Mercedes - Benz A 140 Elegance	365,750 TL	Vehicle	2025-05-15	View Remove
2005 Opel Vectra 1.6 Elegance	451,000 TL	Vehicle	2025-05-15	View Remove
1997 Hyundai Accent 1.5 GLS	268,125 TL	Vehicle	2025-05-15	View Remove

Back to Home

minisahibinden.com

← Back to Home

Post a New Ad

Post a Vehicle | Post Real Estate

Vehicle Model *

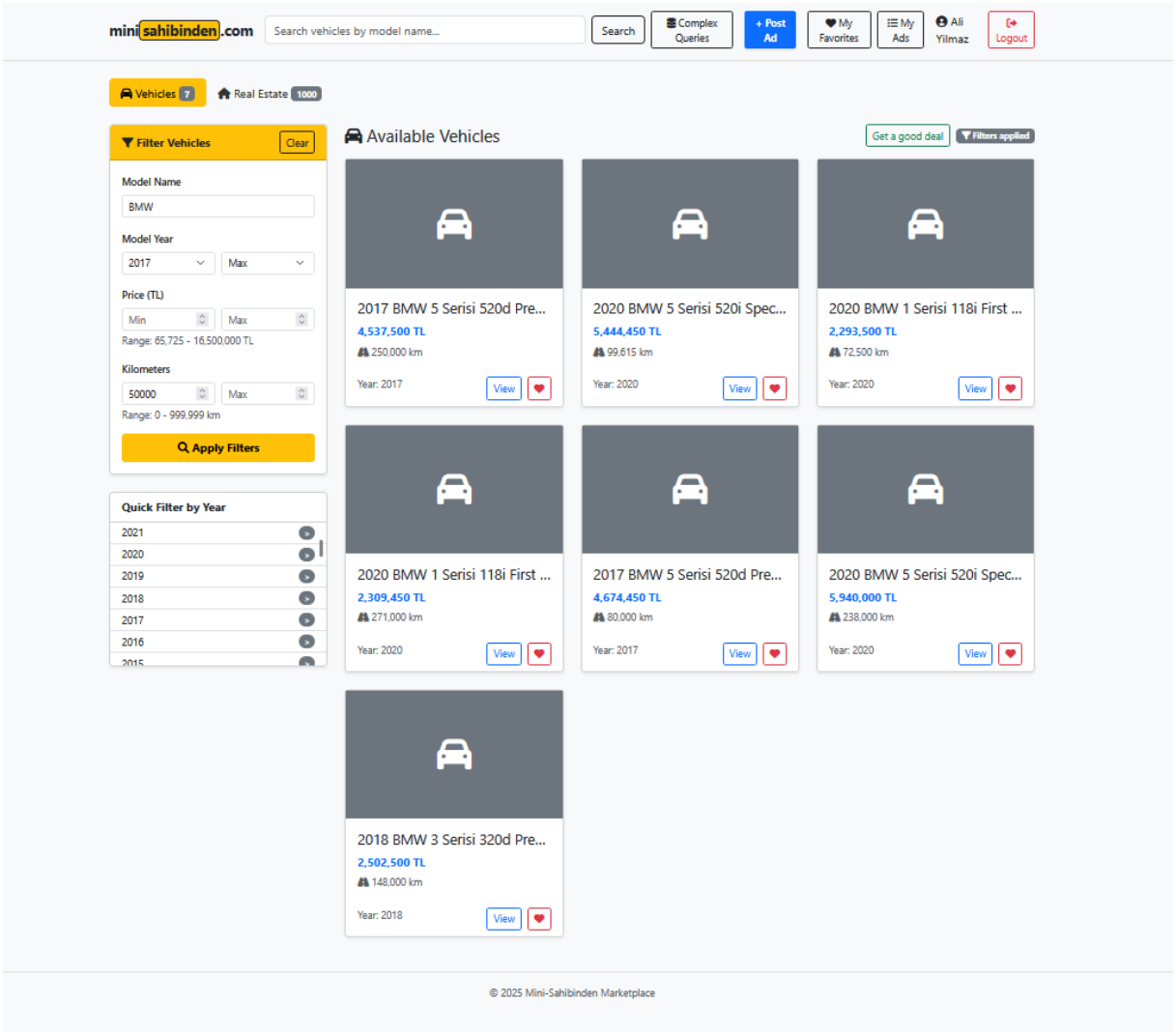
e.g., BMW 320i, Mercedes C200

Model Year * 2020

Price (TL) * 500000

Kilometers * 50000

Post Vehicle Ad


minisahibinden.com
Search vehicles by model name...
Search
Complex Queries
+ Post Ad
My Favorites
My Ads
Ali Yilmaz
Logout
Vehicles 7
Real Estate 1000
Filter Vehicles
Clear
Model Name
BMW
Model Year
2017
Max
Price (TL)
Min
Max
Range: 65,725 - 16,500,000 TL
Kilometers
50000
Max
Range: 0 - 999,999 km
Apply Filters
Quick Filter by Year
2021
2020
2019
2018
2017
2016
2015
Available Vehicles
Get a good deal
Filters applied
2017 BMW 5 Serisi 520d Pre...
4,537,500 TL
250,000 km
Year: 2017
View
2020 BMW 5 Serisi 520i Spec...
5,444,450 TL
99,615 km
Year: 2020
View
2020 BMW 1 Serisi 118i First ...
2,293,500 TL
72,500 km
Year: 2020
View
2020 BMW 1 Serisi 118i First ...
2,309,450 TL
271,000 km
Year: 2020
View
2017 BMW 5 Serisi 520d Pre...
4,674,450 TL
80,000 km
Year: 2017
View
2020 BMW 5 Serisi 520i Spec...
5,940,000 TL
238,000 km
Year: 2020
View
2018 BMW 3 Serisi 320d Pre...
2,502,500 TL
148,000 km
Year: 2018
View
© 2025 Mini-Sahibinden Marketplace

mini sahibinden .com

Home

Complex SQL Queries - Vehicle Database

Explore advanced SQL queries with different filtering and aggregation capabilities for our vehicle marketplace.

Query 1: Active Sellers

Find vehicles from sellers who have posted multiple listings (uses subquery with GROUP BY and HAVING).

Minimum Listings

2

Execute Query

Query 2: Above Average Price

Find vehicles priced above the overall average (uses correlated subquery).

No parameters needed

Execute Query

Query 3: Price & Seller Filter

Filter by price range and seller name pattern (uses JOIN with multiple WHERE conditions).

Min Price (TL)

0

Max Price (TL)

5000000

Seller Name Pattern

e.g., Ali

Execute Query

Query 4: Top Vehicles Per Year

Get top N vehicles per model year ranked by price (uses window function ROW_NUMBER).

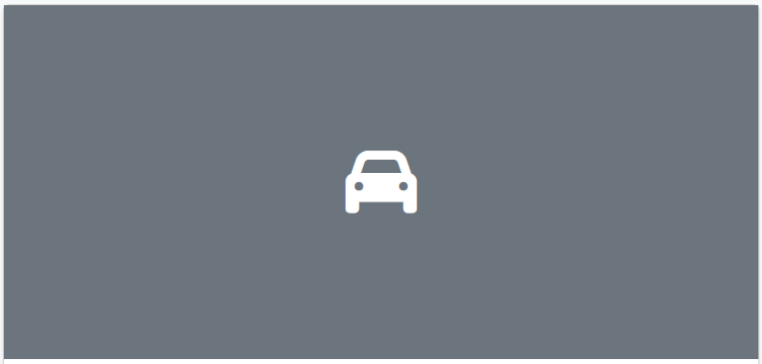
Top N Vehicles Per Year

3

Execute Query

mini sahibinden .com

Back to Home



451,000 TL

Seller Info

Name: Merve Ozdemir
Email: merve.ozdemir@email.com
Phone: 555-883-8822

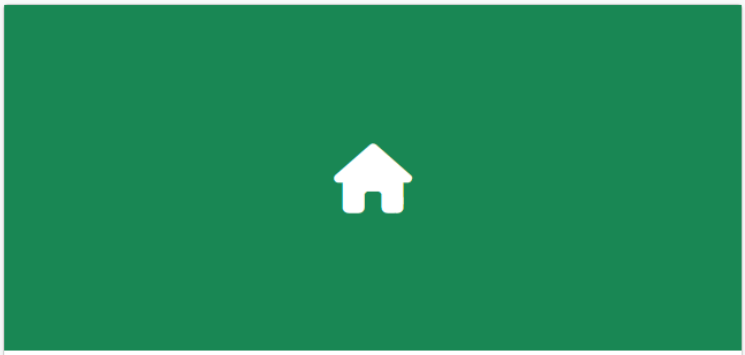
2005 Opel Vectra 1.6 Elegance

Active

Model Year	Kilometers	Listed
2005	216,000 km	2025-05-15

Vehicle Details

Model	Opel Vectra 1.6 Elegance
Year	2005
Kilometers	216,000 km
Listing Date	2025-05-15



Mugla, Marmaris

Armutalan

Active

Size	Rooms	Listed
135 m²	3+1	2025-05-25

Property Details

City	Mugla
District	Marmaris
Neighborhood	Armutalan
Area	135 m²
Room Config	3+1
Seller Type	Real Estate Agency
Listing Date	2025-05-25

9,800,000 TL

Seller Info

Type: Real Estate Agency
Name: Burcu Ucar
Email: burcu.ucar@email.com
Phone: 555-822-9158