



YILDIZ TEKNİK ÜNİVERSİTESİ
ELEKTRİK-ELEKTRONİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ
BÖLÜMÜ

BLM3041
VERİTABANI YÖNETİMİ DERSİ
DÖNEM PROJESİ

GRUP NUMARASI: 19

KONU: Hobi Pazarı - El Emeği Ürünlerin Sergi & Satış Sistemi

GRUP ÜYELERİ: Elif İNCE - Hüsamettin IŞIKTAŞ - Sümeyye GÜLDEMİR - Yusuf Taha TÜTEN

ÖĞRENCİ NUMARALARI: 20011098 – 21011014 – 21011009 – 21011027

DERS YÜRÜTÜCÜSÜ: M. Utku KALAY

1. PROJE KONUSU:

Hobi Pazarı - El Emeği Ürünlerin Sergi & Satış Sistemi, el yapımı ürünleri sergilemek ve satmak isteyen kullanıcılar için bir platform sunmaktadır. Bu sistem ile birlikte kullanıcılar kişisel profillerini oluşturabilmekte ve el yapımı ürünlerini satışa sunabilmektedir. Aynı zamanda kullanıcılar sisteme yüklenen ürünleri satın alabilmektedir. Kullanıcılar, sisteme ürünlerini kategorilere ayırarak yüklemekte, açıklama ve fiyat bilgileri ekleyebilmekte, stok ve satış miktarlarını takip edebilmektedir. Ürün satın alan kullanıcılar satın aldıkları ürünler hakkında yorum yapabilmektedir.

2. SİSTEM ÖZELLİKLERİ:

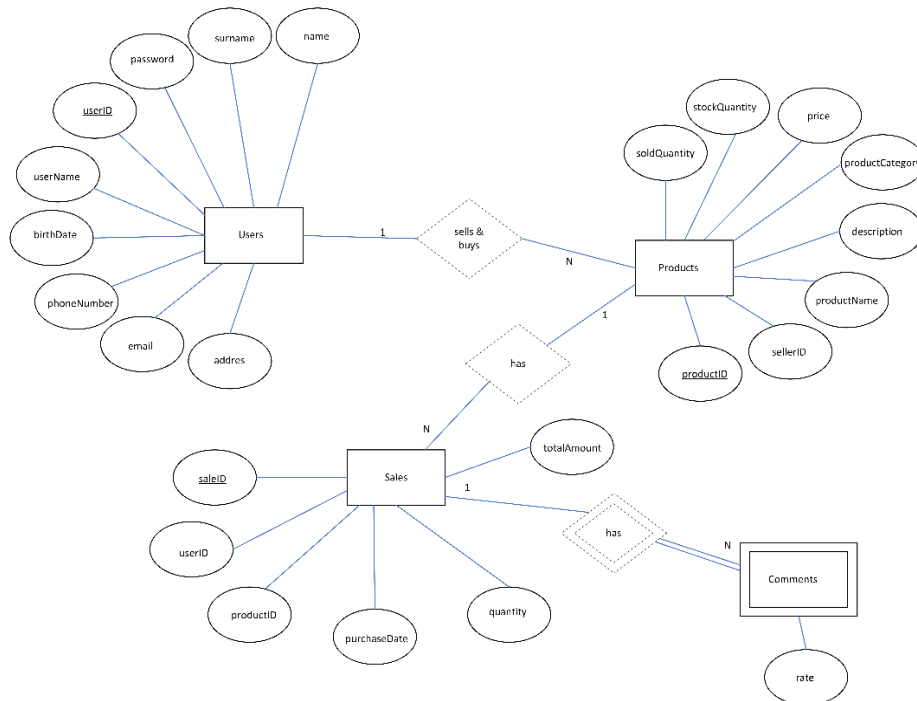
Kullanıcı Yönetimi: Kullanıcılara ait kişisel bilgiler, iletişim bilgileri ve satış geçmişlerini sisteme kaydedilmektedir. Sistemde kullanıcı yaşının 18'den büyük olma kısıtlaması bulunmaktadır.

Ürün Yönetimi: Kullanıcılar el yapımı ürünlerini ekleyebilmekte, açıklama ve fiyat bilgileriyle birlikte kategorisini belirleyebilmektedir.

Satış İşlemleri: Kullanıcılar satışa konan ürünlerden stok miktarınca satın alabilmektedir. Satın alınan ürünün kim tarafından ne zaman satın alındığı ve stok bilgisi depolanmaktadır.

Yorum ve Değerlendirme: Kullanıcılar satın aldıkları ürün ile ilgili yorum yapabilmektedir.

3. E-R DİYAGRAMI:



4. TABLOLAR:

4.1. Kullanıcılar Tablosu (Users):

- Kullanıcıları temsil etmektedir.
- Kullanıcıların kimlik bilgilerini (isim, kullanıcı adı, şifre), iletişim bilgilerini (e-posta, telefon numarası, adres) ve doğum tarihlerine ilişkin bilgileri saklar.
- Telefon numarası uzantısı ve yaş sınırı doğrulaması gibi kısıtlamalar içermektedir.
- Her kullanıcı primary key olarak bir userID'ye sahiptir.

	userid [PK] integer	username character varying (20)	password character varying (20)	name character varying (20)	surname character varying (20)	birthdate date	email character varying (50)	phonenumber character varying (13)	address character varying (255)
1	1	bhains0	wJ4Fuj#M	Bentley	Hains	1990-04-14	bhains0@gmail.com	+905052196176	4 Arizona Center Suite 53
2	2	gepdell1	gU0a*abc	Gerry	Epdell	1982-06-05	gepdell1@hotmail.com	+905479538603	02133 Schmedeman Place Room 1440
3	3	jprobate2	wI9&2.Oe3	Jenn	Probate	1998-05-14	jprobate2@gmail.com	+905949670535	357 Laurel Terrace Suite 32
4	4	gvuitte3	sD99AfctkPV	Gale	Vuitte	1997-09-24	gvuitte3@hotmail.com	+905039061650	6190 Welch Avenue Suite 47
5	5	aluberti4	yG6h4n9J	Anson	Luberti	1979-04-13	aluberti4@gmail.com	+905128827985	461 Golf Pass Suite 76
6	6	jcrudgington5	sX4Y2dLj	Johnnie	Crudgington	1979-08-11	jcrudgington5@gmail.com	+905274783013	4 Lakewood Lane Room 76
7	7	bfrith6	wX3TV7Q8yf0	Bea	Frith	1991-01-13	bfrith6@hotmail.com	+905373170290	86133 Rldney Lane Room 1344
8	8	sfloode7	gB7DrIxW	Sloan	Flode	2003-12-17	sfloode7@hotmail.com	+905238403886	68 Redwing Plaza 10th Floor
9	9	thayller8	gASXsX4K	Tirrell	Hayller	1981-02-21	thayller8@hotmail.com	+905823977143	16 Butterfield Road Room 1350
10	10	lcheney9	hU4suf65u	Luther	Cheney	2000-10-03	lcheney9@gmail.com	+905117348173	3 Birchwood Court Apt 924
11	11	rbellaya	vS5U&OVVvunn8	Rossie	Bellay	1980-05-27	rbellaya@gmail.com	+905654289080	25 Pierstorff Court 19th Floor
12	12	alivockb	dV4C_4r_cy83	Alfonse	Livock	1976-08-19	alivockb@gmail.com	+905665061508	9 Memorial Junction Suite 39
13	13	lwyantc	pY9cKYiC!	Loutitia	Wyant	1989-06-10	lwyantc@gmail.com	+905388860463	63621 American Avenue 8th Floor
14	14	aclohissyd	sD0HXGHH7N	Alane	Clohissy	2001-08-25	aclohissyd@gmail.com	+905411824287	898 Hudson Lane Room 1140
15	15	sspareye	uW9T1234wd	Stormy	Sparey	1998-03-11	sspareye@gmail.com	+905301894676	4 Dapin Alley Apt 1490

4.2. Ürünler Tablosu (Products):

- El yapımı ürünleri temsil etmektedir.
- Ürünlerin adını, açıklamasını, kategorisini, fiyatını, stok miktarını, satış miktarını ve satıcı bilgisini içermektedir.
- Her bir ürün primary key olarak bir productID'ye sahiptir.
- Satıcı bilgisi, kullanıcılar tablosundaki ilgili kullanıcının userID'sine referansla ilişkilendirilmektedir.

	productid [PK] integer	sellerid integer	productname character varying (50)	description text	productcategory character varying (30)	price numeric	stockquantity integer	soldquantity integer
1	1	1	macrame wall hanging	30x50 dimensions, beige color, suitable for long-term use	wall decoration	250.00	2	0
2	2	2	landscape oil painting	50x50 dimensions, soft colors	wall decoration	570.00	1	0
3	3	3	friendship bracelet	adjustable size, colors may vary depending on stock availabili...	jewelry	20.00	12	5
4	4	4	quartz beaded necklace	adjustable size, symbolizes power	jewelry	160.00	11	3
5	5	5	brass ring	fully handmade, tarnish free	jewelry	240.00	20	2
6	6	5	koi fish brass brooch	tarnish free, koi fish symbolize perseverance	jewelry	350.00	7	2
7	7	6	denim backpack	suitable for a 14" laptop	bag	400.00	5	2
8	8	1	macrame phone pouch	20x14 dimensions, available in black and blue colors	bag	50.00	9	1
9	9	6	crossbody leather bag	25x15 dimensions, adjustable size, unisex	bag	450.00	5	3
10	10	7	ocean-scented candle	12 hours burning time	candle	25.00	27	4
11	11	2	natural soap	perfume-free	soap	30.00	25	3
12	12	7	sulfur soap	good for eczema, not suitable for the face	soap	35.00	43	7

4.3. Satışlar Tablosu (Sales):

- Gerçekleşen satışları temsil etmektedir.
- Kullanıcılar tarafından yapılan satın alımları, ürünleri, ürünlerin satın alınma tarihini, miktarını ve toplam tutarları içermektedir.
- Her satış primary key olarak bir saleID'ye sahiptir.
- Kullanıcı ve ürün bilgileri, sırasıyla kullanıcılar ve ürünler tablosundaki userID ve productID ile ilişkilendirilmektedir.

	saleid [PK] integer	userid integer	productid integer	purchasedate timestamp without time zone	quantity integer	totalamount numeric
1	1	11	3	2023-12-01 12:55:55	2	40.00
2	2	13	3	2023-11-25 14:30:12	3	60.00
3	3	15	4	2023-11-26 12:55:55	1	160.00
4	4	15	6	2023-12-16 18:23:20	1	350.00
5	5	10	4	2023-12-17 16:27:12	1	160.00
6	6	7	4	2023-11-12 22:50:55	1	160.00
7	7	6	5	2023-10-01 23:13:12	2	480.00
8	8	8	6	2023-10-25 21:45:15	1	350.00
9	9	4	7	2023-10-14 16:24:34	1	400.00
10	10	5	7	2023-10-18 19:25:34	1	400.00
11	11	2	8	2023-10-18 17:45:34	1	50.00
12	12	5	9	2023-10-18 15:25:34	2	900.00
13	13	1	9	2023-10-15 19:25:34	1	450.00
14	14	3	10	2023-10-19 19:25:34	2	50.00
15	15	5	10	2023-11-18 22:25:34	2	50.00
16	16	8	11	2023-12-18 19:25:34	3	90.00
17	17	7	12	2023-11-14 20:25:34	1	35.00
18	18	8	12	2023-10-20 12:25:34	5	175.00
19	19	5	12	2023-10-05 14:25:34	1	35.00

4.4. Yorumlar Tablosu (Comments):

- Ürün ve satışlarla ilişkili kullanıcı yorumlarını temsil etmektedir.
- Sales tablosuna saleID üzerinden zayıf varlık olarak bağlıdır.
- Yorumlar puan olarak yapılmaktadır ve bu puanlar 0 ile 5 arasında değer almaktadır.
- Satış tablosundaki bir satışın saleID'sine referansla ilişkilendirilmektedir.

	saleid [PK] integer	rate integer
1	1	4
2	2	5
3	3	3
4	4	5
5	5	5
6	6	4
7	7	4
8	8	5
9	9	3
10	10	4
11	11	3
12	12	5
13	13	5
14	14	3
15	15	4
16	16	3
17	17	2
18	18	3
19	19	2

5. ARAYÜZ TANITIMI VE İLGİLİ KODLAR:

5.1. Giriş Ekranı

The screenshot shows the login interface for 'Hobi Pazarı'. It features a dark background with a purple header. The title 'Hobi Pazarı' is displayed in white. Below the title, there are two input fields: 'Kullanıcı Adı' (Username) and 'Şifre' (Password). A blue 'Giriş Yap' (Login) button is positioned below the password field, and a grey 'Kaydol' (Register) button is below it. A red error message at the bottom reads: '(- -) Geçersiz Kullanıcı Adı veya Şifre (- -)'.

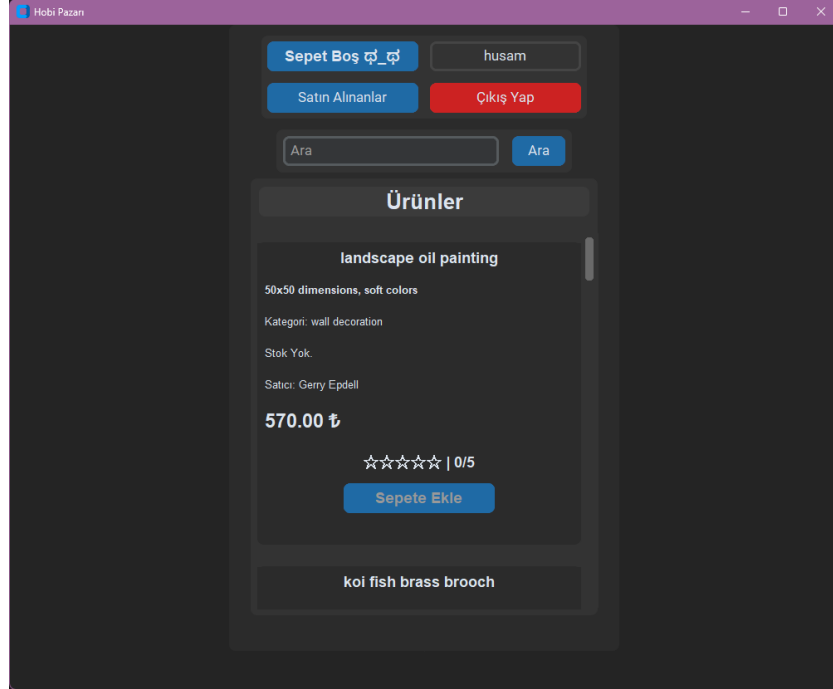
Kullanıcı uygulamayı açtığında karşısına ilk olarak giriş ekranı çıkmaktadır. Veri tabanında kayıtlı olan kullanıcılar ilgili kullanıcı adı ve şifre bölümlerini doldurarak doğrudan giriş yapabilmektedir. Bu aşamada kullanıcının hatalı giriş yapması durumunda arayüzde hata mesajı görüntülenmektedir.

5.2. Kayıt Olma Ekranı

The screenshot shows the registration interface for 'Hobi Pazarı'. It features a dark background with a purple header. The title 'Kaydol' (Register) is displayed in white. Below the title, there are several input fields: 'Kullanıcı Adınız' (Username), 'Şifreniz' (Password), 'Adınız' (First Name), 'Soyadınız' (Last Name), 'E-posta Adresiniz' (Email Address), 'Telefon Numaranız' (Phone Number), and 'Adresiniz' (Address). Below these fields, there is a section for selecting the birth date, labeled 'Doğum Tarihinizi Seçiniz'. This section includes a calendar widget showing the month of January for the year 2006. The calendar has a grid with days of the week (Mon, Tue, Wed, Thu, Fri, Sat, Sun) and dates (1-31). The date 16 is highlighted in blue. At the bottom of the screen, there is a blue 'Kayıt Ol' (Register) button.

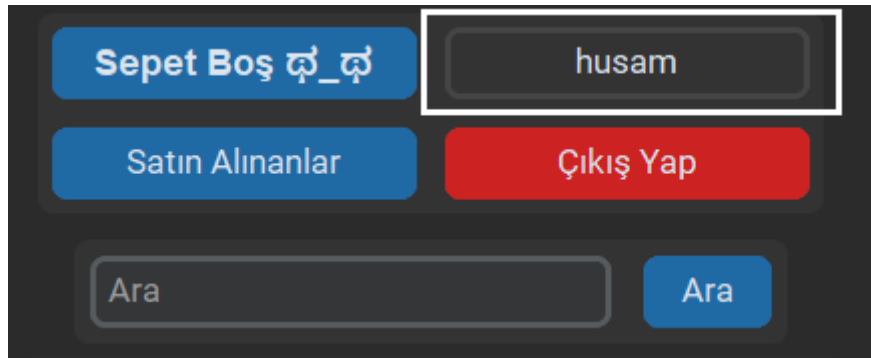
Sisteme henüz kaydolmamış kullanıcılar kendilerinden istenen bilgileri doldurarak üyelik oluşturabilmektedir. Burada her bir kullanıcı hem satıcı hem de alıcı olabildiğinden 18 yaş kısıtı getirilmiştir.

5.3. Ana Ekran



Kullanıcı sisteme giriş yaptıktan sonra doğrudan ana ekrana yönlendirilmektedir. Ana ekranda sepet, profil, satın alınan ürünler ve çıkış butonları ve satışa konulmuş ürünler ile ilgili arama ve filtreleme işlemlerinin yapılabileceği arama barı bulunmaktadır. Sistem ilk açıldığında veri tabanında bulunan bütün ürünler ilgili panel içerisinde sırasıyla görüntülenebilmektedir.

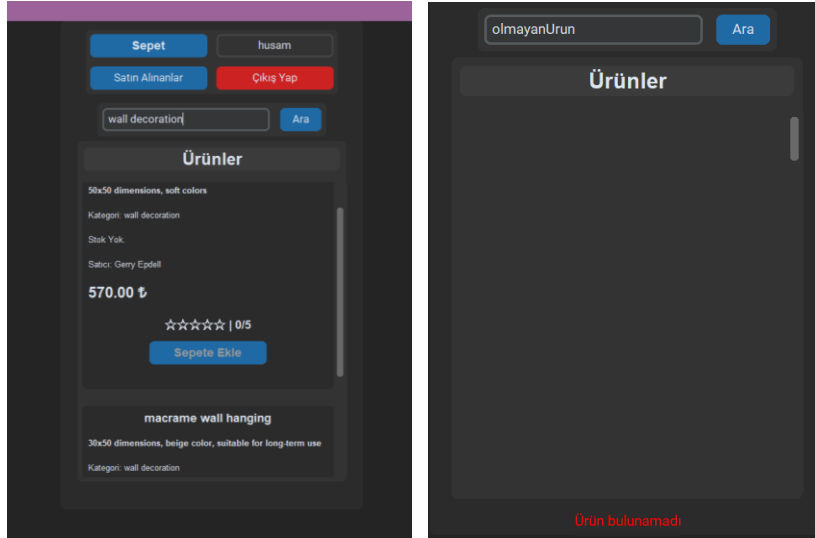
5.4. Profil Bilgileri Görüntüleme/Güncelleme



Kullanıcılar, ana ekranda kendilerine ait kullanıcı adının yazılı olduğu butona tıklayarak profilleri ile ilgili bilgileri görüntüleyebilecekleri ve düzenleyebilecekleri panele erişebilmektedir. Kullanıcı bilgilerinde değişiklik yaptıktan sonra ekranda yapılan değişiklik ile ilgili bir bilgilendirme mesajı görüntülenmektedir. Aşağıda bununla ilgili bir örnek verilmiştir.

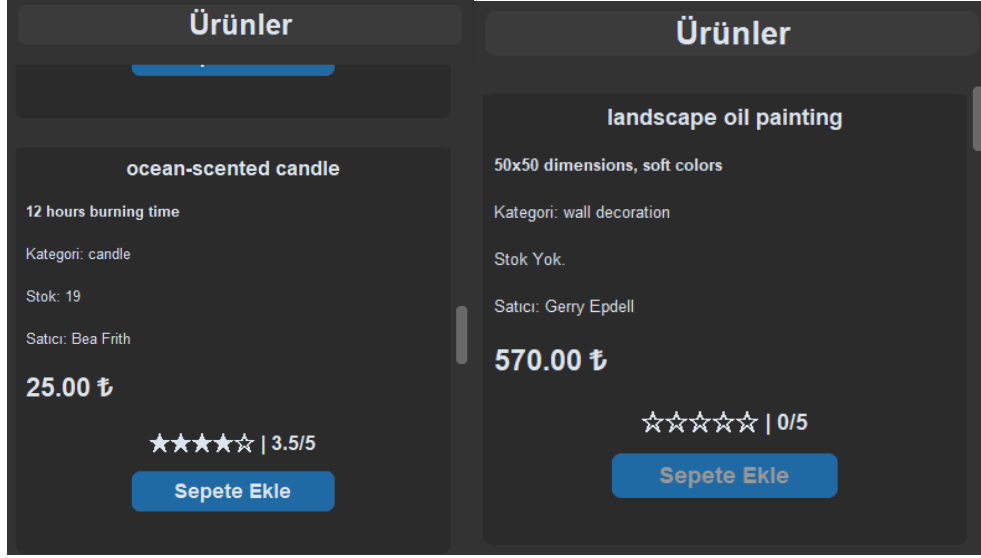


5.5. Ürün Arama/Filtreleme



Kullanıcılar arama barı aracılığıyla hobi ürünleri arasında, kategorilere göre filtreleme yapabilmekte ya da doğrudan ismini bildiği ürünü aratabilmektedir. Satışı bulunmayan bir ürün aratıldığında arayüzde hata mesajı gösterilmektedir.

5.6. Sepete Ürün Ekleme

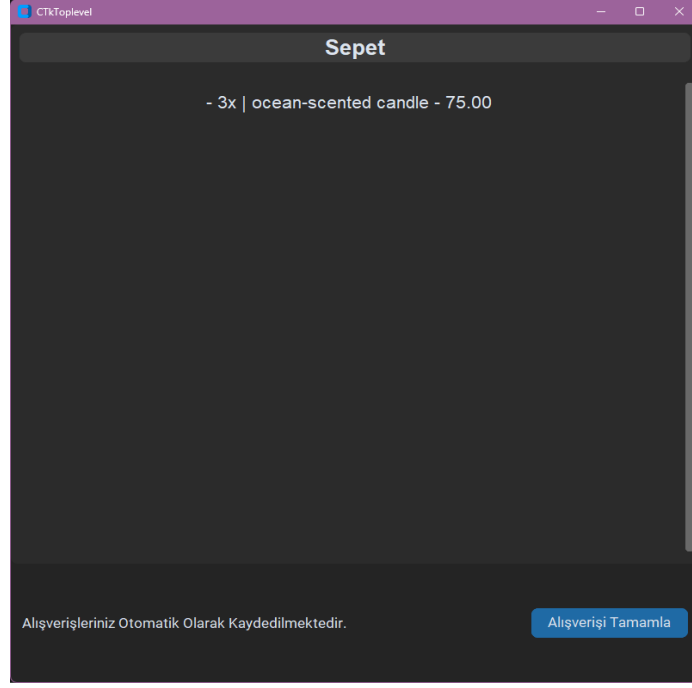


Kullanıcılar görüntüledikleri ürünleri “Sepete Ekle” butonu aracılığıyla sepetlerine ekleyebilmektedir. Görüntülenen ürünlere ilişkin stok bulunmaması halinde, hatalı eklemenin önüne geçilmesi için, buton aktif durumdan pasif duruma geçmektedir. Projenin gelecekteki çok kullanıcıli versiyonları için arayüz, bir ürün sepete eklendiğinde ve ürünü satın alma işlemi henüz gerçekleştirilmediğinde geçici olarak stok durumunu güncellenmektedir.

5.7. Sepet Görüntüsü

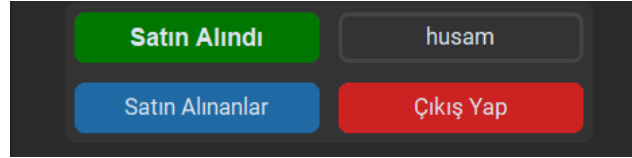
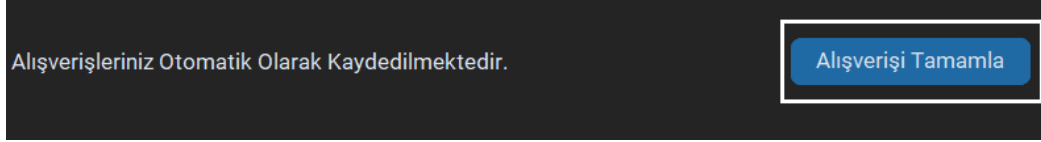


Kullanıcı sepete henüz hiçbir ürün eklediğinde sepetin boş olduğu bilgisi ana ekranda sepet ile ilgili butonda görüntülenebilmektedir. Sepet boş olduğu sürece ilgili buton herhangi bir pencereyi tetiklememektedir. Kullanıcı sepetine ürün ekledikten sonra buton, sepetteki ürün sayısını gösterecek şekilde güncellenmektedir.



Kullanıcının sepetine eklediği ürünler sırayla sepet panelinde görüntülenmektedir.

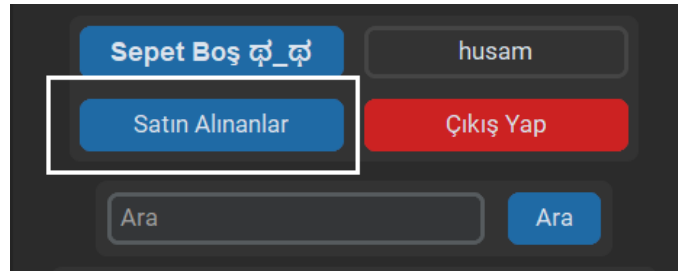
5.8. Ürün Satın Alma

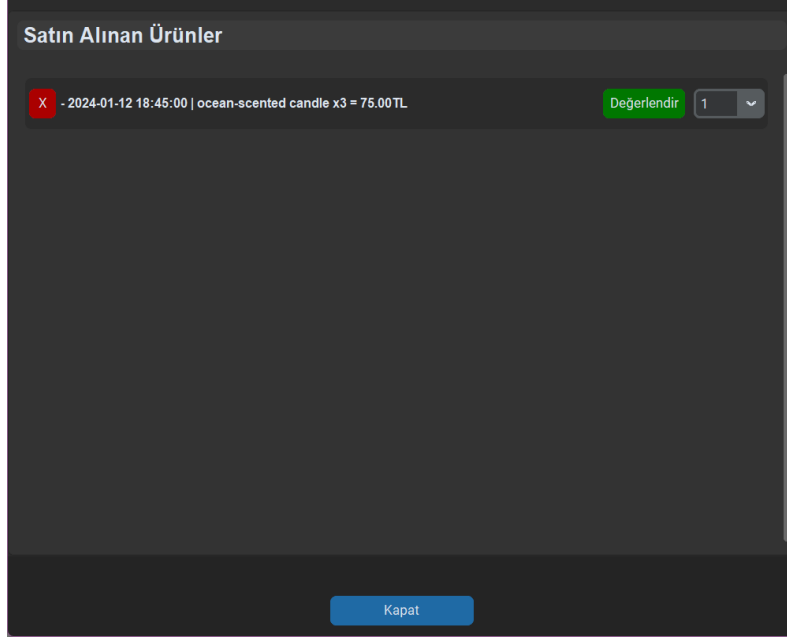


Kullanıcı “Alışverişi Tamamla” butonuna bastığında satın alma işlemi gerçekleşmiş olur. Eş zamanlı olarak veri tabanı ve arayüz içerisinde gerekli güncellemeler de gerçekleştirilir ve kullanıcıya satın alma işleminin gerçekleştirildiği bilgisi verilir.

5.9. Daha Önce Satın Alınan Ürünler İle İlgili İşlemler

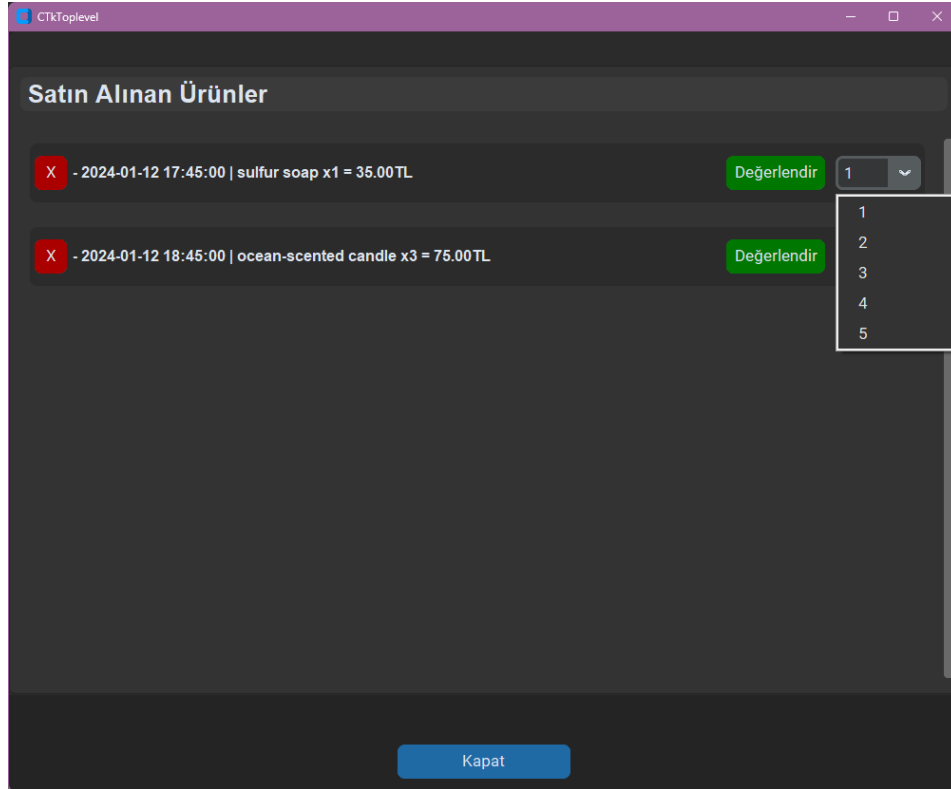
5.9.1. Görüntüleme

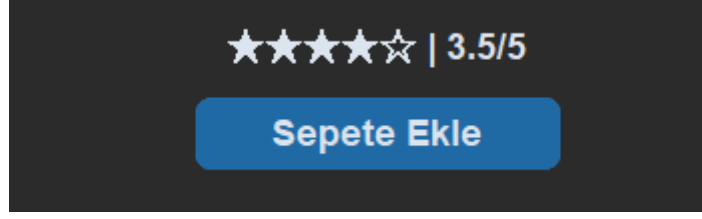




Kullanıcılar satın aldıkları ürünler ile ilgili değerlendirmede bulunabilmektedir. Ana sayfada bulunan “Satın Alınanlar” butonu ile birlikte daha önce satın alınmış olan ürünlerin listelendiği “Satın Alınan Ürünler” sayfası açılmaktadır. Burada satın alınan ürünün ismi, adet ve fiyat bilgisi, işlem tarihi ve değerlendirme bölümü bulunmaktadır.

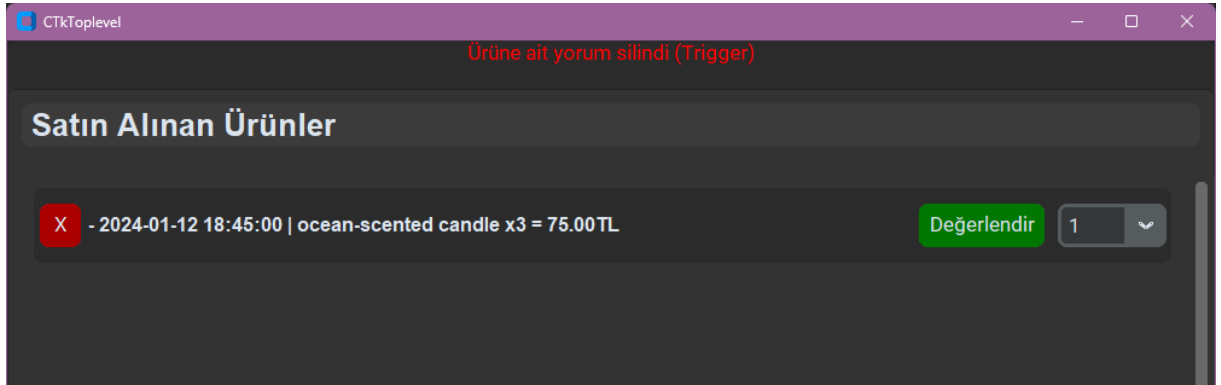
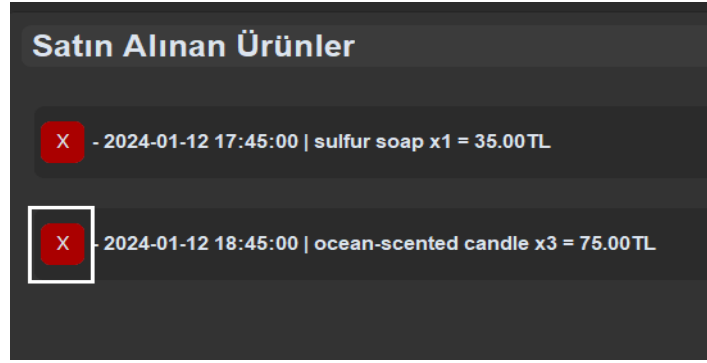
5.9.2. Değerlendirme





Kullanıcılar, satın aldıkları ürünlerin görüntülenebildiği panel üzerinden bu ürünler ile ilgili değerlendirme yapabilmektedir. Değerlendirmeler 1-5 arası puanlama şeklinde yapılabilmektedir. Ürünü satın alan kullanıcıların yaptıkları değerlendirmelerin ortalaması ürün ile ilgili bilgilerin verildiği ana sayfadaki panelde görüntülenmektedir.

5.9.3. Değerlendirmeyi Silme



Kullanıcılar satın alma kayıtlarını (X) butonu aracılığıyla silebilmektedir. Silme işleminin ardından o satışa ait değerlendirme varsa trigger aracılığıyla silinir ve bilgi mesajı verilir.

6. PROJE İSTERLERİNİN GERÇEKLENMESİ

6.1. 4 Adet Tablo ve En Az 10 Adet Kayıt

```
CREATE TABLE IF NOT EXISTS users (
```

```
CREATE TABLE IF NOT EXISTS products (
```

```
CREATE TABLE IF NOT EXISTS sales (
```

```
CREATE TABLE IF NOT EXISTS comments(
```

```
-- userID, username,password,name,surname,birthDate,email,phoneNumber,address
INSERT INTO users VALUES (nextval('users_seq'), 'bhains0', 'wJ4/Fuj#M', 'Bentley', 'Hains', '1990-04-14', 'bhains0@gmail.com', '+905052196
INSERT INTO users VALUES (nextval('users_seq'), 'gepdell1', 'gU0a*abc', 'Gerry', 'Epdell', '1982-06-05', 'gepdell1@hotmail.com', '+9054795
INSERT INTO users VALUES (nextval('users_seq'), 'jprobate2', 'wI9&2.Oe3', 'Jenn', 'Probate', '1998-05-14', 'jprobate2@gmail.com', '+905949
INSERT INTO users VALUES (nextval('users_seq'), 'gvuitte3', 'sD99AfctkPV', 'Gale', 'Vuitte', '1997-09-24', 'gvuitte3@hotmail.com', '+90503
INSERT INTO users VALUES (nextval('users_seq'), 'aluberti4', 'yG6h4n9J', 'Anson', 'Luberti', '1979-04-13', 'aluberti4@gmail.com', '+905128
INSERT INTO users VALUES (nextval('users_seq'), 'jcrudgington5', 'sX4Y2dLj', 'Johnnie', 'Crudgington', '1979-08-11', 'jcrudgington5@gmail.
INSERT INTO users VALUES (nextval('users_seq'), 'bfrith6', 'wX3T,V7Q8yF0', 'Bea', 'Frith', '1991-01-13', 'bfrith6@hotmail.com', '+90537317
INSERT INTO users VALUES (nextval('users_seq'), 'sflode7', 'gB7DrIxiw', 'Sloan', 'Flode', '2003-12-17', 'sflode7@hotmail.com', '+9052384038
INSERT INTO users VALUES (nextval('users_seq'), 'thayller8', 'gA5XsX4K', 'Tirrell', 'Hayller', '1981-02-21', 'thayller8@hotmail.com', '+90
INSERT INTO users VALUES (nextval('users_seq'), 'lcheney9', 'hu4suf6Su', 'Luther', 'Cheney', '2000-10-03', 'lcheney9@gmail.com', '+9051173
INSERT INTO users VALUES (nextval('users_seq'), 'rbellaya', 'v55U&0WVVunn8', 'Rossie', 'Bellay', '1980-05-27', 'rbellaya@gmail.com', '+905
INSERT INTO users VALUES (nextval('users_seq'), 'alivockb', 'dV4C_4r_cy83', 'Alfonse', 'Livock', '1976-08-19', 'alivockb@gmail.com', '+905
INSERT INTO users VALUES (nextval('users_seq'), 'lwyantc', 'pY9cKYICl', 'Loutitia', 'Wyant', '1989-06-10', 'lwyantc@gmail.com', '+90538886
INSERT INTO users VALUES (nextval('users_seq'), 'aclohissyd', 'sD0HXGH#N', 'Alane', 'Clohissy', '2001-08-25', 'aclohissyd@gmail.com', '+9
INSERT INTO users VALUES (nextval('users_seq'), 'sspareye', 'uM9T1234wd', 'Stormy', 'Sparey', '1998-03-11', 'sspareye@gmail.com', '+905301
```

```
-- productID, sellerID , productName, description, productCategory, price, stockQuantity, soldQuantity
INSERT INTO products VALUES (nextval('products_seq'), 1, 'macrame wall hanging', '30x50 dimensions, beige color, suitable for long-ter
INSERT INTO products VALUES (nextval('products_seq'), 2, 'landscape oil painting', '50x50 dimensions, soft colors', 'wall decoration',
INSERT INTO products VALUES (nextval('products_seq'), 3, 'friendship bracelet', 'adjustable size, colors may vary depending on stock a
INSERT INTO products VALUES (nextval('products_seq'), 4, 'quartz beaded necklace', 'adjustable size, symbolizes power', 'jewelry', 160
INSERT INTO products VALUES (nextval('products_seq'), 5, 'brass ring', 'fully handmade, tarnish free', 'jewelry', 240.00, 20, 2);
INSERT INTO products VALUES (nextval('products_seq'), 5, 'koi fish brass brooch', 'tarnish free, koi fish symbolize perseverance', 'je
INSERT INTO products VALUES (nextval('products_seq'), 6, 'denim backpack', 'suitable for a 14" laptop', 'bag', 400.00, 5, 2);
INSERT INTO products VALUES (nextval('products_seq'), 1, 'macrame phone pouch', '20x14 dimensions, available in black and blue colors'
INSERT INTO products VALUES (nextval('products_seq'), 6, 'crossbody leather bag', '25x15 dimensions, adjustable size, unisex', 'bag',
INSERT INTO products VALUES (nextval('products_seq'), 7, 'ocean-scented candle', '12 hours burning time', 'candle', 25.00, 27, 4);
INSERT INTO products VALUES (nextval('products_seq'), 2, 'natural soap', 'perfume-free', 'soap', 30.00, 25, 3);
INSERT INTO products VALUES (nextval('products_seq'), 7, 'sulfur soap', 'good for eczema, not suitable for the face', 'soap', 35.00, 4
```

```
-- saleID, userID, productID, purchaseDate, quantity, totalAmount
INSERT INTO sales VALUES (nextval('sales_seq'), 11, 3, '2023-12-01 12:55:55', 2, calculateTotalAmount(3, 2));
INSERT INTO sales VALUES (nextval('sales_seq'), 13, 3, '2023-11-25 14:30:12', 3, calculateTotalAmount(3, 3));
INSERT INTO sales VALUES (nextval('sales_seq'), 15, 4, '2023-11-26 12:55:55', 1, calculateTotalAmount(4, 1));
INSERT INTO sales VALUES (nextval('sales_seq'), 15, 6, '2023-12-16 18:23:20', 1, calculateTotalAmount(6, 1));
INSERT INTO sales VALUES (nextval('sales_seq'), 10, 4, '2023-12-17 16:27:12', 1, calculateTotalAmount(4, 1));
INSERT INTO sales VALUES (nextval('sales_seq'), 7, 4, '2023-11-12 22:50:55', 1, calculateTotalAmount(4, 1));
INSERT INTO sales VALUES (nextval('sales_seq'), 6, 5, '2023-10-01 23:13:12', 2, calculateTotalAmount(5, 2));
INSERT INTO sales VALUES (nextval('sales_seq'), 8, 6, '2023-10-25 21:45:15', 1, calculateTotalAmount(6, 1));
INSERT INTO sales VALUES (nextval('sales_seq'), 4, 7, '2023-10-14 16:24:34', 1, calculateTotalAmount(7, 1));
INSERT INTO sales VALUES (nextval('sales_seq'), 5, 7, '2023-10-18 19:25:34', 1, calculateTotalAmount(7, 1));
INSERT INTO sales VALUES (nextval('sales_seq'), 2, 8, '2023-10-18 17:45:34', 1, calculateTotalAmount(8, 1));
INSERT INTO sales VALUES (nextval('sales_seq'), 5, 9, '2023-10-18 15:25:34', 2, calculateTotalAmount(9, 2));
INSERT INTO sales VALUES (nextval('sales_seq'), 1, 9, '2023-10-15 19:25:34', 1, calculateTotalAmount(9, 1));
INSERT INTO sales VALUES (nextval('sales_seq'), 3, 10, '2023-10-19 19:25:34', 2, calculateTotalAmount(10, 2));
INSERT INTO sales VALUES (nextval('sales_seq'), 5, 10, '2023-11-18 22:25:34', 2, calculateTotalAmount(10, 2));
INSERT INTO sales VALUES (nextval('sales_seq'), 8, 11, '2023-12-18 19:25:34', 3, calculateTotalAmount(11, 3));
INSERT INTO sales VALUES (nextval('sales_seq'), 7, 12, '2023-11-14 20:25:34', 1, calculateTotalAmount(12, 1));
INSERT INTO sales VALUES (nextval('sales_seq'), 8, 12, '2023-10-20 12:25:34', 5, calculateTotalAmount(12, 5));
INSERT INTO sales VALUES (nextval('sales_seq'), 5, 12, '2023-10-05 14:25:34', 1, calculateTotalAmount(12, 1));
```

```
-- saleID, rate
INSERT INTO comments VALUES (1, 4);
INSERT INTO comments VALUES (2, 5);
INSERT INTO comments VALUES (3, 3);
INSERT INTO comments VALUES (4, 5);
INSERT INTO comments VALUES (5, 5);
INSERT INTO comments VALUES (6, 4);
INSERT INTO comments VALUES (7, 4);
INSERT INTO comments VALUES (8, 5);
INSERT INTO comments VALUES (9, 3);
INSERT INTO comments VALUES (10, 4);
INSERT INTO comments VALUES (11, 3);
INSERT INTO comments VALUES (12, 5);
INSERT INTO comments VALUES (13, 5);
INSERT INTO comments VALUES (14, 3);
INSERT INTO comments VALUES (15, 4);
INSERT INTO comments VALUES (16, 3);
INSERT INTO comments VALUES (17, 2);
INSERT INTO comments VALUES (18, 3);
INSERT INTO comments VALUES (19, 2);
```

6.2. Primary Key ve Foreign Key

```
CREATE TABLE IF NOT EXISTS products (
    productID INT NOT NULL PRIMARY KEY,
    sellerID INT NOT NULL,
    productName VARCHAR(50) UNIQUE NOT NULL,
    description text,
    productCategory VARCHAR(30) NOT NULL,
    price NUMERIC NOT NULL,
    stockQuantity INT NOT NULL,
    soldQuantity INT,
    CHECK (stockQuantity >= 0),
    FOREIGN KEY (sellerID)
        REFERENCES users (userID) ON DELETE CASCADE
);
```


6.3. Silme ve Sayı Kısıtları

```
CREATE OR REPLACE FUNCTION checkAge(birthDate TIMESTAMP)
RETURNS BOOLEAN AS $$
BEGIN
    RETURN birthDate <= NOW() - INTERVAL '18 years';
END;
$$ LANGUAGE plpgsql;

CREATE TABLE IF NOT EXISTS users (
    userID INT NOT NULL PRIMARY KEY,
    username VARCHAR(20) UNIQUE NOT NULL,
    password VARCHAR(20) NOT NULL,
    name VARCHAR(20) NOT NULL,
    surname VARCHAR(20) NOT NULL,
    birthDate DATE NOT NULL,
    email VARCHAR(50) UNIQUE NOT NULL,
    phoneNumber VARCHAR(13),
    address VARCHAR(255),
    CHECK (checkAge(birthDate) = TRUE),
    CONSTRAINT check_phoneNumber CHECK (phoneNumber LIKE '+905%')
);
```

```
CREATE TABLE IF NOT EXISTS products (
    productID INT NOT NULL PRIMARY KEY,
    sellerID INT NOT NULL,
    productName VARCHAR(50) UNIQUE NOT NULL,
    description text,
    productCategory VARCHAR(30) NOT NULL,
    price NUMERIC NOT NULL,
    stockQuantity INT NOT NULL,
    soldQuantity INT,
    CHECK (stockQuantity >= 0),
    FOREIGN KEY (sellerID)
        REFERENCES users (userID) ON DELETE CASCADE
);
```

6.4. Arayüzden Insert, Update ve Delete İşlemleri

```
def buy_products(self, product_list_all):
    global global_userid
    # Satın alma işlemini veritabanında kaydetme

    for product in product_list_all:
        # Örnek bir SQL sorgusu, burada 'purchases' tablosu ve sütunlar varsayılan olarak belirlenmiştir.
        # """INSERT INTO users VALUES (nextval('users_seq'), %s, %s, %s, %s, %s, %s, %s, %s)"""
        cur.execute("""INSERT INTO sales (saleid, userid, productid, quantity, totalamount)
                        VALUES (nextval('sales_seq'), %s, %s, %s, %s)""",
                        (global_userid, product[1], product[3], product[4]))

        cur.execute("""UPDATE products
                        SET stockquantity = stockquantity - %s
                        WHERE productid = %s""", (product[3], product[1],))

    if self.info_pop is None or not self.info_pop.wininfo_exists():
        self.info_pop = info_popup(self)
        self.info_pop.focus_set()
        self.info_pop.grab_set()
        self.wait_window(self.info_pop)
    else:
        self.info_pop.focus()

def delete_kayit(self, sale_id):

    cur.execute("""DELETE FROM sales
                    WHERE saleid = %s""", (sale_id,))

    conn.commit()
```

6.5. Arayüzden girilecek değere göre ekrana sonuçların listelenmesi

```
--Lists products in the desired category or name
CREATE OR REPLACE FUNCTION get_products_by_name(p_productName TEXT)
RETURNS SETOF product_type AS $$
BEGIN
    IF p_productName = ' ' THEN
        RETURN QUERY
        SELECT productID, sellerID, productName, description,
        productCategory, price, stockQuantity, soldQuantity
        FROM products;
    ELSE
        RETURN QUERY
        SELECT productID, sellerID, productName, description,
        productCategory, price, stockQuantity, soldQuantity
        FROM products
        WHERE productName ILIKE '%' || p_productName || '%' or
        productCategory ILIKE '%' || p_productName || '%';
    END IF;
END;
$$ LANGUAGE plpgsql;
```

```
def search(self, product_name):
    if product_name == "":
        cur.execute("""SELECT * FROM get_products_by_name(' ');""")
    else:
        cur.execute("""SELECT * FROM get_products_by_name(%s);""", (product_name,))
    return cur.fetchall()
```

6.6. Arayüzden “View” Sorgusu Çağırılması

```
--VIEW

CREATE VIEW product_view AS
SELECT * FROM products;
```

6.7. Sequence

```
CREATE SEQUENCE sales_seq
START 1
INCREMENT 1
OWNED BY sales.saleID;

INSERT INTO sales VALUES (nextval('sales_seq'), 11, 3, '2023-12-01 12:55:55', 2, calculateTotalAmount(3, 2));
```

6.8. Union

```
SELECT username, userid, password
FROM users
WHERE username = 'admin'
UNION
SELECT NULL, NULL, password
FROM users
WHERE username = 'admin'
LIMIT 1;
```

6.9. Aggregate ve Having

```
--Lists the average rate given to the selected product
SELECT s.productID, AVG(rate)
FROM comments c, sales s
WHERE s.saleID = c.saleID AND c.saleID = 5
GROUP BY s.productID ;
```



```
--Lists products with an average rate above 3
SELECT p.productName, AVG(c.rate)
FROM comments c, sales s, products p
WHERE s.saleID = c.saleID AND s.productID = p.productID --??
GROUP BY p.productID
HAVING AVG(c.rate) > 3
ORDER BY AVG(c.rate) desc
```

6.10. Record ve Cursor İçeren 3 Farklı Fonksiyon

```
--CURSOR, FUNCTION
--Updates if there is a comment, if not inserts
CREATE OR REPLACE FUNCTION update_or_insert_comment(p_saleID integer, p_rate numeric)
RETURNS BOOLEAN AS $$
DECLARE
    comment_cursor CURSOR FOR SELECT * FROM comments WHERE saleID = p_saleID;
    comment_row comments%ROWTYPE;
    is_success BOOLEAN := false;
BEGIN
    OPEN comment_cursor;
    FETCH comment_cursor INTO comment_row;

    IF FOUND THEN
        -- Update if row exists
        UPDATE comments SET rate = p_rate WHERE saleID = p_saleID;
        is_success := true;
    ELSE
        -- Insert if not exist
        INSERT INTO comments(saleID, rate) VALUES (p_saleID, p_rate);
        is_success := true;
    END IF;

    CLOSE comment_cursor;
    RETURN is_success;
END;
$$ LANGUAGE plpgsql;
```

```

--Returns the total price of the product
CREATE OR REPLACE FUNCTION calculateTotalAmount(p_productID INT,p_quantity INT)
RETURNS NUMERIC AS $$
DECLARE
    v_price NUMERIC;
BEGIN
    SELECT price INTO v_price
    FROM products
    WHERE productID = p_productID;
    RETURN v_price * p_quantity;
END;
$$ LANGUAGE plpgsql;

```

```

CREATE TYPE product_type AS (
    productid INT,
    sellerid INT,
    productname VARCHAR(50),
    description TEXT,
    productcategory VARCHAR(30),
    price NUMERIC,
    stockquantity INT,
    soldquantity INT
);

```

```

--FUNCTION
--Lists products in the desired category or name
CREATE OR REPLACE FUNCTION get_products_by_name(p_productName TEXT)
RETURNS SETOF product_type AS $$
BEGIN
    IF p_productName = ' ' THEN
        RETURN QUERY
        SELECT productID, sellerID, productName, description,
        productCategory, price, stockQuantity, soldQuantity
        FROM products;
    ELSE
        RETURN QUERY
        SELECT productID, sellerID, productName, description,
        productCategory, price, stockQuantity, soldQuantity
        FROM products
        WHERE productName ILIKE '%' || p_productName || '%' or
        productCategory ILIKE '%' || p_productName || '%' ;
    END IF;
END;
$$ LANGUAGE plpgsql;

```

6.11. 2 Adet Trigger

```
--TRIGGER
--Set purchase date of new sale with current date and time
CREATE OR REPLACE FUNCTION update_purchase_date()
RETURNS TRIGGER AS $$
BEGIN
    NEW.purchasedate := TO_CHAR(CURRENT_TIMESTAMP, 'YYYY-MM-DD HH24:MI');
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER trigger_update_purchase_date
BEFORE INSERT ON sales
FOR EACH ROW
EXECUTE FUNCTION update_purchase_date();
```

```
--TRIGGER
--Deletes comments related to the deleted sale
CREATE OR REPLACE FUNCTION deleteRelatedComments()
RETURNS TRIGGER AS $$
BEGIN
    DELETE FROM comments
    WHERE saleID = OLD.saleID;
    RAISE NOTICE 'Comment for sale ID % has been deleted', OLD.saleID;
    RETURN OLD;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER deleteRelatedCommentsTrigger
AFTER DELETE
ON sales
FOR EACH ROW
EXECUTE FUNCTION deleteRelatedComments();
```