



DAMG 6210

STUDENT

MARKETPLACE

Dhruv Sunil Parmar
Dilip Sharma
Robin Ignatious
Sai Kashyap Cheruku
Nishita Vikas Shewale

Submitted to: Prof. Manuel Montrond



Project Overview

The **Student Marketplace** is a digital platform designed to create a vibrant and sustainable ecosystem within the student community, offering a convenient space for students to buy and sell various items, including textbooks, electronics, furniture, and fashion. This marketplace aims to streamline the process of transactions, foster community engagement, and support the financial well-being of students by providing an accessible and user-friendly platform.

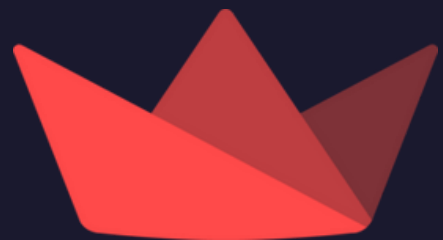


Tools Used



Tableau

Tableau is a data visualization tool used for converting raw data into an understandable format, primarily in the form of interactive and shareable dashboards.



Streamlit

Streamlit enables the creation of interactive web GUIs for database CRUD operations, allowing users to easily manage and interact with database data through a web interface built with Python code.



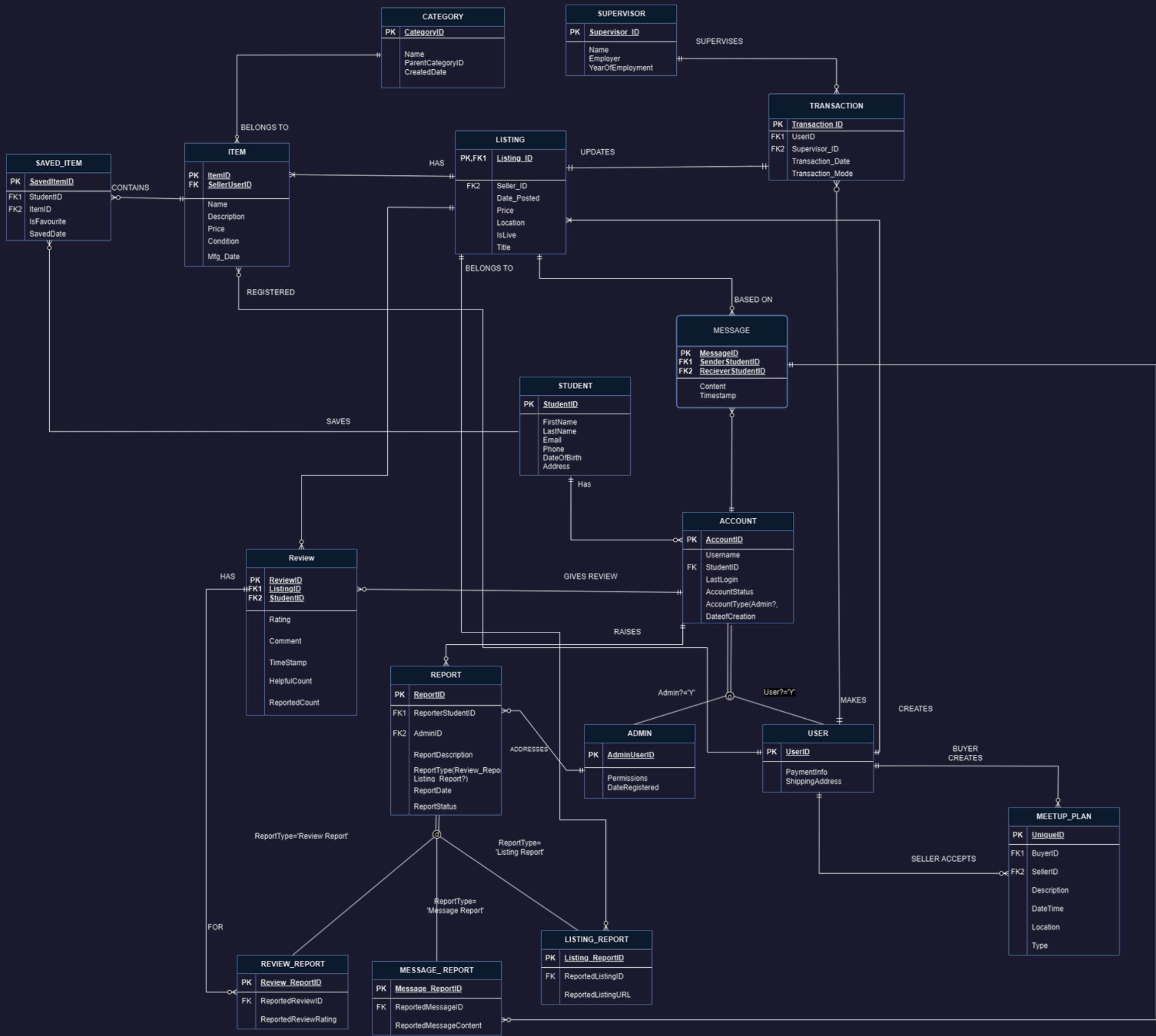
SQL

SQL (Structured Query Language) is a standard programming language used for managing and manipulating relational databases. It allows for storing, querying, updating, and deleting data, as well as managing database schemas.





ERD



STORED PROCEDURES

```
/*Stored Procedure for getting reviews on a particular LISTING*/
CREATE PROCEDURE GetReviewsOnItem
    @ListingID VARCHAR(100)
AS
BEGIN
    SELECT
        I.Item_Name,
        L.Listing_ID,
        L.Date_Posted AS Listing_Date,
        R.Rating,
        R.Comment
    FROM
        LISTING L
    JOIN
        ITEM I ON L.Listing_ID = I.Item_ID
    LEFT JOIN
        REVIEW R ON I.Item_ID = R.Listing_ID
    WHERE
        L.Listing_ID = 'A01';
END;

EXEC GetReviewsOnItem @ListingID = 'A01';
```

	Item_Name	Listing_ID	Listing_Date	Rating	Comment
1	Smartphone X	A01	2022-01-20	8	Excellent smartphone! Working perfectly.

```
/*Stored Procedure for Add to Cart / Save an Item from the Live Listings */
CREATE PROCEDURE AddtoCart @student_id INTEGER, @listing_id VARCHAR(100)
AS
BEGIN
    IF(EXISTS(SELECT * FROM LISTING WHERE LISTING.Listing_ID = @listing_id AND LISTING.IsLive = 'TRUE'))
    BEGIN
        INSERT INTO SAVED_ITEM VALUES(@student_id,@listing_id,'Y',GETDATE())
    END;
    ELSE
    BEGIN
        SELECT 'Cannot serve request' as Failure
    END;
END;

EXEC AddtoCart @student_id = '2', @listing_id = 'F23';
```

```
/*Stored Procedure for getting avg, +ve review count, -ve review count,
neutral review count and total review count for a user with listings*/
CREATE PROCEDURE GetSellerStats @user_id INTEGER,
    @avg_rating DECIMAL(10,2) OUTPUT,
    @pos_review INTEGER OUTPUT,
    @neg_review INTEGER OUTPUT,
    @neu_review INTEGER OUTPUT,
    @total_review INTEGER OUTPUT AS
BEGIN
    CREATE TABLE t(
        rating DECIMAL NOT NULL,
        range_review VARCHAR(20) NOT NULL
    )
    INSERT INTO t(rating,range_review)
    (SELECT r.Rating as rating, CASE
        WHEN r.Rating >= 8 THEN 'GOOD'
        WHEN r.Rating >= 5 THEN 'AVERAGE'
        ELSE 'BAD' END as range_review FROM REVIEW as r JOIN LISTING as l on r.Listing_ID=l.Listing_ID WHERE l.Listing_Seller_ID=@user_id);
    SELECT @avg_rating=AVG(rating) FROM t;
    SELECT @pos_review=COUNT(*) FROM t WHERE range_review LIKE 'GOOD';
    SELECT @neu_review=COUNT(*) FROM t WHERE range_review LIKE 'AVERAGE';
    SELECT @neg_review=COUNT(*) FROM t WHERE range_review LIKE 'BAD';
    SELECT @total_review=COUNT(*) FROM t;
    DROP TABLE t;
END;
GO
DECLARE
    @user_id INTEGER,
    @avg_rating DECIMAL(10,2),
    @pos_review INTEGER,
    @neg_review INTEGER,
    @neu_review INTEGER,
    @total_review INTEGER;

SET @user_id = 117;
EXEC GetSellerStats
    @user_id = @user_id,
    @avg_rating = @avg_rating OUTPUT,
    @pos_review = @pos_review OUTPUT,
    @neg_review = @neg_review OUTPUT,
    @neu_review = @neu_review OUTPUT,
    @total_review = @total_review OUTPUT;

PRINT 'Average Rating: ' + CONVERT(VARCHAR(10), @avg_rating);
PRINT 'Positive Reviews: ' + CONVERT(VARCHAR(10), @pos_review);
PRINT 'Negative Reviews: ' + CONVERT(VARCHAR(10), @neg_review);
PRINT 'Neutral Reviews: ' + CONVERT(VARCHAR(10), @neu_review);
PRINT 'Total Reviews: ' + CONVERT(VARCHAR(10), @total_review);
```

```
Average Rating: 8.00
Positive Reviews: 1
Negative Reviews: 0
Neutral Reviews: 0
Total Reviews: 1
```


VIEWS

```
/* View reviews on items added to cart / saved item */
CREATE VIEW saved_item_reviews_view AS
SELECT
t.item_id as item_id,
t.item_name as item_name,
r.Comment as comment,
r.Rating as rating
FROM (SELECT i.Item_ID as item_id,
i.Item_Name as item_name
FROM SAVED_ITEM as saved,
ITEM as i WHERE saved.Item_ID=i.Item_ID) as t
JOIN REVIEW as r on t.item_id=r.Listing_ID;
```

item_id	item_name	comment	rating
A01	Smartphone X	Excellent smartphone! Working perfectly.	8
B02	Designer Jeans	Nice pair of jeans. Good quality.	7
C03	Gaming Laptop Pro	Gaming laptop is excellent. No issues.	9
D04	Wireless Headpho...	Decent headphones. Working well.	6
E05	Men Running Shoes	Love these running shoes! Excellent cho...	10

```
/*View reported messages made by users across listings*/
CREATE VIEW reported_message_view AS
SELECT
i.Item_ID as item_id, i.Item_Name as item_name, m.content, s1.First_Name as Sender_Name, s2.First_Name as Receiver_Name
FROM MESSAGE_REPORT as mr JOIN REPORT as r on mr.Report_ID = r.Report_ID JOIN
MESSAGE as m ON mr.Message_ID = m.Message_ID JOIN ITEM as i on m.Listing_ID = i.Item_ID JOIN STUDENT as s1 on s1.Student_id = m.Sender_Student_ID
JOIN STUDENT as s2 on s2.Student_id = m.Receiver_Student_ID;
```

	item_id	item_name	content	Sender_Name	Receiver_Name
1	B02	Designer Jeans	Hello John, can I have your Number & House Addre...	Jane	John
2	C03	Gaming Laptop Pro	Hi Emily, I saw your Gaming Laptop Pro listing. Can ...	Bob	Emily
3	E05	Men Running Shoes	Hi Grace, I love the Men Running Shoes you listed. ...	Alex	Grace
4	G07	Smartwatch Elite	Hi Olivia, I'm interested in the Smartwatch Elite. Doe...	Michael	Olivia
5	I09	Vintage Camera	Hi Sophia, the Vintage Camera looks amazing! Can ...	Henry	Sophia
6	K11	Desktop Workstation	Hi Ava, the Desktop Workstation would be perfect for...	Liam	Ava

```
/* Active Sellers */
CREATE VIEW active_sellers_view AS
SELECT
I.Listing_Seller_ID as Seller_ID,
COUNT(DISTINCT I.Listing_ID) as Total_Active_Listings
FROM
LISTING I
WHERE
I.IsLive = 'TRUE'
GROUP BY
I.Listing_Seller_ID;
```

```
SELECT * FROM active_sellers_view;
```

	Seller_ID	Total_Active_Listings
1	101	2
2	102	3
3	103	1
4	104	1
5	105	1
6	106	3
7	107	1
8	108	1
9	109	2

```
/* Recently Posted Items */
CREATE VIEW recently_posted_items_view AS
SELECT TOP 25 PERCENT
I.Listing_ID,
I.Date_Posted,
i.Item_Name,
c.Category_Name,
I.Location
FROM
LISTING I
JOIN
ITEM i ON I.Listing_ID = i.Item_ID
JOIN
CATEGORY c ON i.Item_Category = c.Category_ID
ORDER BY
I.Date_Posted DESC;
```

	Listing_ID	Date_Posted	Item_Name	Category_Name	Location
1	H25	2024-01-25	Smartphone Accessories Kit	Mobile Accessories	Lowell
2	G24	2023-12-15	Formal Wallet Collection	Wallets	Lowell
3	F23	2023-11-30	Wireless Gaming Consoles	Gaming Consoles	Lowell
4	E22	2023-10-28	Portable Chargers Set	Portable Chargers	Boston
5	D21	2023-09-20	Fashionable Glasses	Smart Glasses	Lowell
6	C20	2023-08-07	Power Bank Essentials	Power Banks	Boston
7	B19	2023-07-30	Travel Backpack	Travel Backpacks	Lowell

DATA ENCRYPTION ON PERSONALLY IDENTIFIABLE INFORMATION

Student Table (Phone,Email)-

Student_ID	First_Name	Last_Name	Phone	Email
1	John	Doe	0x0007995CF4C88844AF86FE44E764B9E202000000F2B11ECC55...	0x0096889410EF364EAAB689F71F86119B02000000C6EDD77163...
2	Jane	Smith	0x0007995CF4C88844AF86FE44E764B9E2020000008C6B131D3E...	0x0096889410EF364EAAB689F71F86119B02000000A8637FF4BA...
3	Bob	Johnson	0x0007995CF4C88844AF86FE44E764B9E2020000002852A6D38A...	0x0096889410EF364EAAB689F71F86119B020000001E5F29C043...
4	Emily	Davis	0x0007995CF4C88844AF86FE44E764B9E202000000C73CB574E0...	0x0096889410EF364EAAB689F71F86119B02000000143C176D2A...
5	Alex	Martin	0x0007995CF4C88844AF86FE44E764B9E202000000C15F781FB7...	0x0096889410EF364EAAB689F71F86119B02000000E8C2A40E28...
6	Grace	Wilson	0x0007995CF4C88844AF86FE44E764B9E202000000E153741F8A...	0x0096889410EF364EAAB689F71F86119B020000008EF62F0A37...
7	Michael	Clark	0x0007995CF4C88844AF86FE44E764B9E20200000020BC02FE52...	0x0096889410EF364EAAB689F71F86119B02000000E6FB9595C5...
8	Olivia	Jones	0x0007995CF4C88844AF86FE44E764B9E2020000003272EB21D8...	0x0096889410EF364EAAB689F71F86119B02000000AC7EF43E91...
9	Henry	Anderson	0x0007995CF4C88844AF86FE44E764B9E202000000FE5B3DFE8F...	0x0096889410EF364EAAB689F71F86119B020000008F7168E950...
10	Sophia	Moore	0x0007995CF4C88844AF86FE44E764B9E202000000B641D24BB6...	0x0096889410EF364EAAB689F71F86119B020000003937F32D57...
11	Liam	Taylor	0x0007995CF4C88844AF86FE44E764B9E2020000002BFC1AB32C...	0x0096889410EF364EAAB689F71F86119B0200000089B98F5EFD...

User Table (Payment Info)-

Account_ID	User_ID	Shipping_Address	Payment_Info
1001	101	123 Main St	0x0096889410EF364EAAB689F71F86119B02000000E602364...
1002	102	456 Oak St	0x0096889410EF364EAAB689F71F86119B0200000054A199E...
1003	103	789 Maple Ave	0x0096889410EF364EAAB689F71F86119B02000000A23D475...
1004	104	101 Pine St	0x0096889410EF364EAAB689F71F86119B0200000064745E2...
1005	105	202 Maple Ave	0x0096889410EF364EAAB689F71F86119B02000000FFFE121...
1006	106	303 Elm St	0x0096889410EF364EAAB689F71F86119B02000000B5D1D02...
1007	107	404 Birch Ave	0x0096889410EF364EAAB689F71F86119B0200000071B0D6E...
1008	108	505 Cedar St	0x0096889410EF364EAAB689F71F86119B0200000058AC4D7...
1010	109	606 Maple Ave	0x0096889410EF364EAAB689F71F86119B020000000543978...
1011	110	707 Pine St	0x0096889410EF364EAAB689F71F86119B020000005C9A2DD...
1013	111	808 Elm St	0x0096889410EF364EAAB689F71F86119B020000002345BF3...

USER DEFINED FUNCTIONS

```
/*Age */

SELECT * FROM marketplace.dbo.STUDENT;

CREATE FUNCTION CalculateAge (@DOB DATE)
RETURNS INT
AS
BEGIN
    DECLARE @Age INT
    SELECT @Age = DATEDIFF(YEAR, @DOB, GETDATE())
    RETURN @Age
END;
```

Student_ID	First_Name	Last_Name	Date_of_Birth	Address	Zip_Code	Phone	Age	Email
1	John	Doe	1995-05-15	123 Main St	12345	0x0007995...	28	0x009
2	Jane	Smith	1998-08-22	456 Oak St	54321	0x0007995...	25	0x009
3	Bob	Johnson	1990-03-10	789 Maple Ave	67890	0x0007995...	33	0x009
4	Emily	Davis	1997-12-05	101 Pine St	45678	0x0007995...	26	0x009
5	Alex	Martin	1992-06-18	202 Oak Ave	23456	0x0007995...	31	0x009
6	Grace	Wilson	1994-09-30	303 Elm St	78901	0x0007995...	29	0x009
7	Michael	Clark	1988-11-20	404 Birch Ave	12389	0x0007995...	35	0x009
8	Olivia	Jones	1996-02-14	505 Cedar St	56743	0x0007995...	27	0x009
9	Henry	Anderson	1993-07-25	606 Maple Ave	90876	0x0007995...	30	0x009
10	Sophia	Moore	1991-04-08	707 Pine St	23457	0x0007995...	32	0x009
11	Liam	Taylor	1995-09-17	808 Elm St	67890	0x0007995...	28	0x009

```
/*Discount Percentage on each ITEM */

CREATE FUNCTION dbo.CalculateDiscount (@ItemPrice DECIMAL(18, 2), @SellingPrice DECIMAL(18, 2))
RETURNS DECIMAL(18)
AS
BEGIN
    DECLARE @Discount DECIMAL(18);

    -- Calculate discount percentage
    IF @ItemPrice > 0
        SET @Discount = ((@ItemPrice - @SellingPrice) / @ItemPrice) * 100;
    ELSE
        SET @Discount = 0;

    RETURN @Discount;
END;
```

Seller...	Item_Name	Item_Desc	Item_Cate...	Item_Price	Item_Cond	Mfg_Date	Selling_Price	DiscountPercentage
101	Smart Home Starter Kit	Intelligent device...	46	300	New	2023-06-20	240	20
102	Designer Jeans	Premium denim jean...	19	45	Like-New	2022-02-20	31	31
102	Travel Backpack	Spacious and durab...	47	50	Like-New	2023-07-28	35	30
103	Gaming Laptop Pro	Powerful gaming la...	4	1500	Refurbished	2022-03-10	300	80
102	Power Bank Essentials	Portable chargers ...	48	25	Refurbished	2023-08-05	12	52
104	Wireless Headphones	High-quality wirel...	8	80	Good	2022-04-05	80	0
106	Fashionable Glasses	Stylish smart glas...	49	150	Good	2023-09-15	150	0
105	Men Running Shoes	Comfortable and st...	20	55	Acceptable	2022-05-12	55	0
106	Portable Chargers Set	Convenient set of ...	50	40	Acceptable	2023-10-23	40	0
106	Designer Dress	Elegant dress desi...	14	70	New	2022-06-18	56	20
109	Wireless Gaming Consoles	Gaming consoles wi...	37	350	New	2023-11-30	280	20

TRIGGERS

```
CREATE TRIGGER update_supervisor_trigger
ON [TRANSACTION]
AFTER INSERT
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @v_supervisor_id INT;

    -- Find the supervisor employed at the time of the transaction
    SELECT @v_supervisor_id = Supervisor_ID
    FROM SUPERVISOR
    WHERE EXISTS (
        SELECT 1
        FROM INSERTED i
        WHERE i.Transaction_Date >= SUPERVISOR.Date_of_Employment
    );

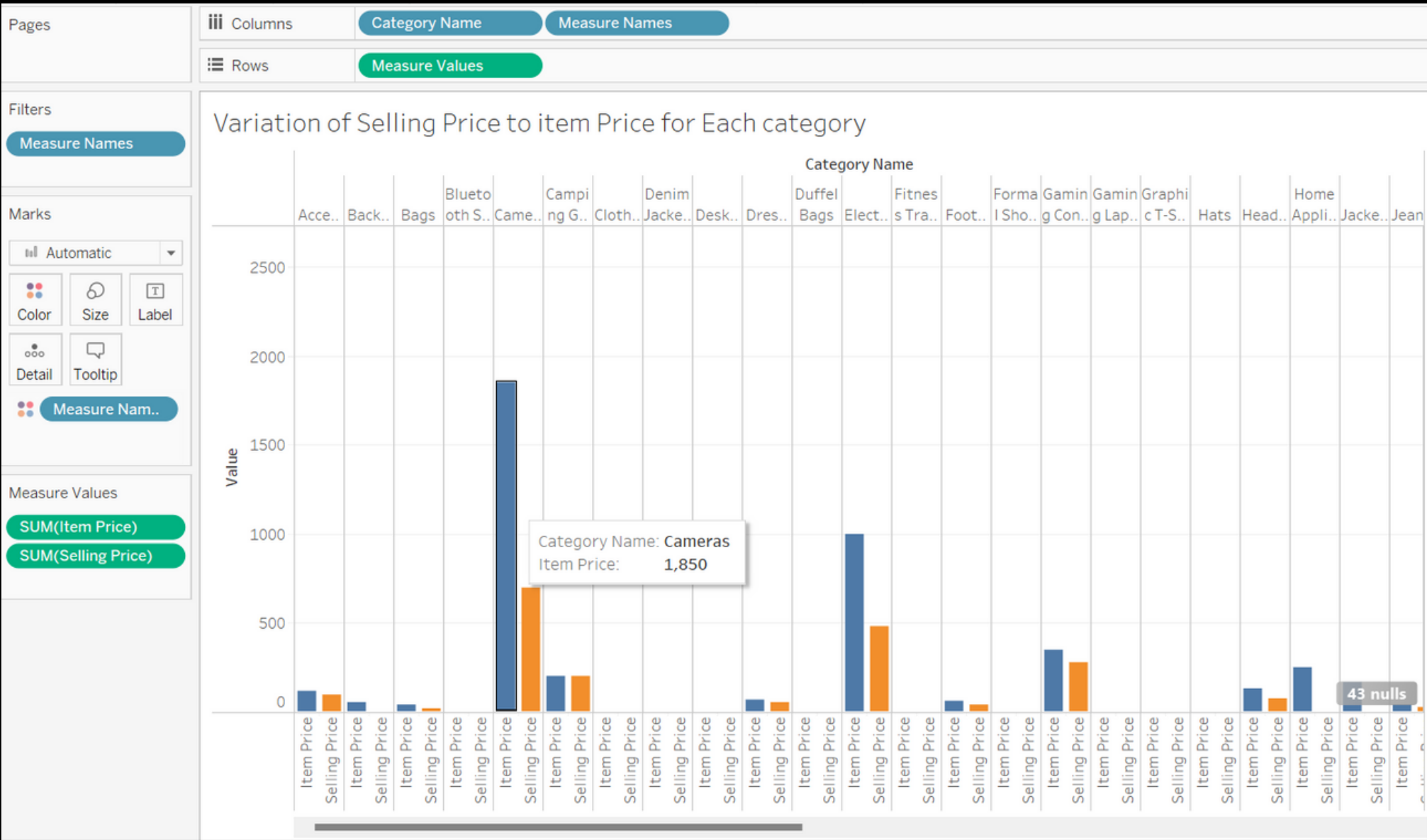
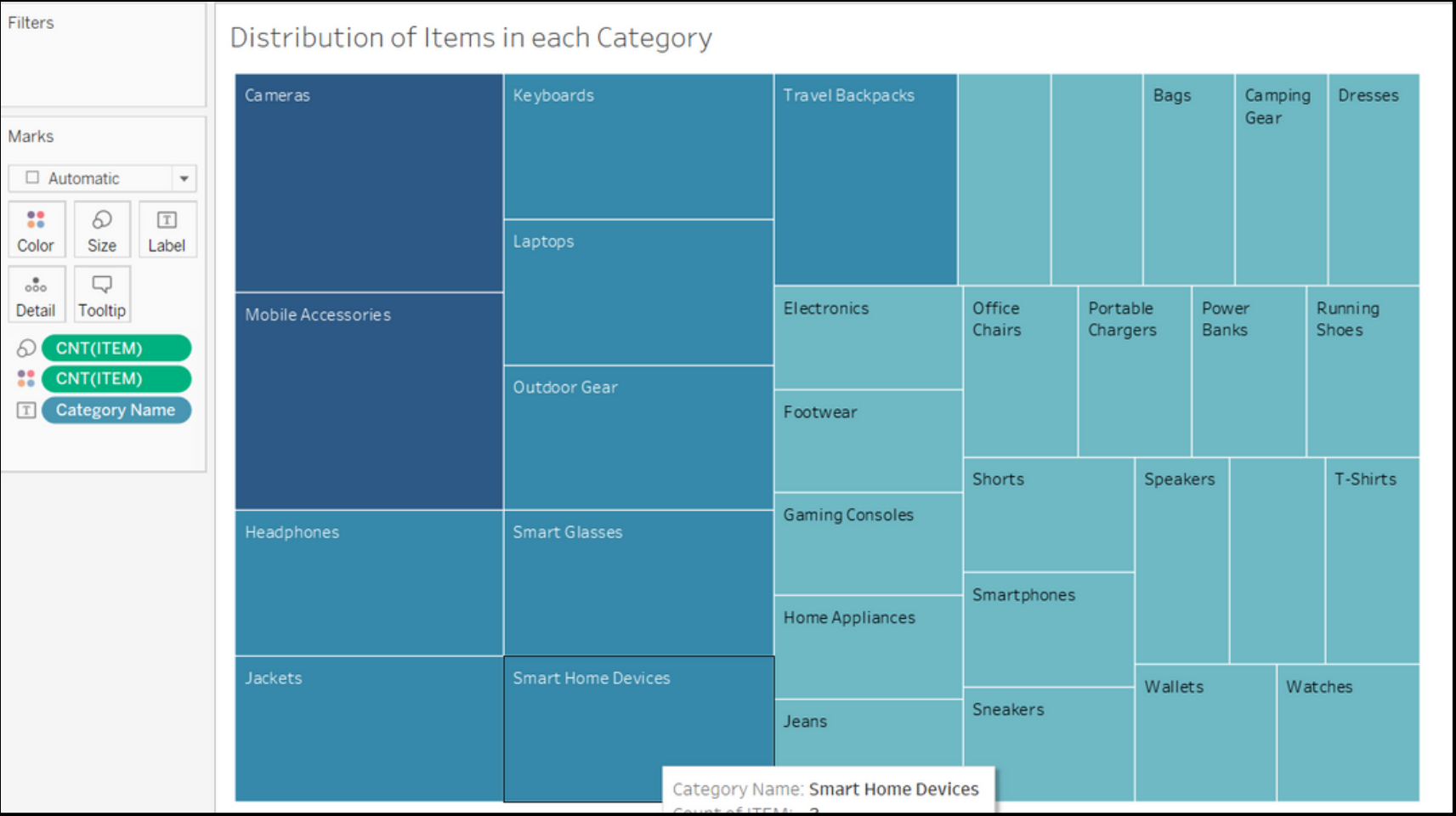
    -- Update the Supervisor_ID in the inserted rows
    UPDATE t
    SET t.Supervisor_ID = @v_supervisor_id
    FROM [TRANSACTION] t
    JOIN INSERTED i ON t.Transaction_ID = i.Transaction_ID;
END;

-- Trigger should automatically assign Supervisor_ID to a New Transaction
INSERT INTO [TRANSACTION] (Transaction_ID, Transaction_Date, Listing_ID, Transaction_Mode, User_ID, Supervisor_ID)
VALUES
    (28, '2024-03-01', 'Q17', 'DEBIT', 112, NULL);

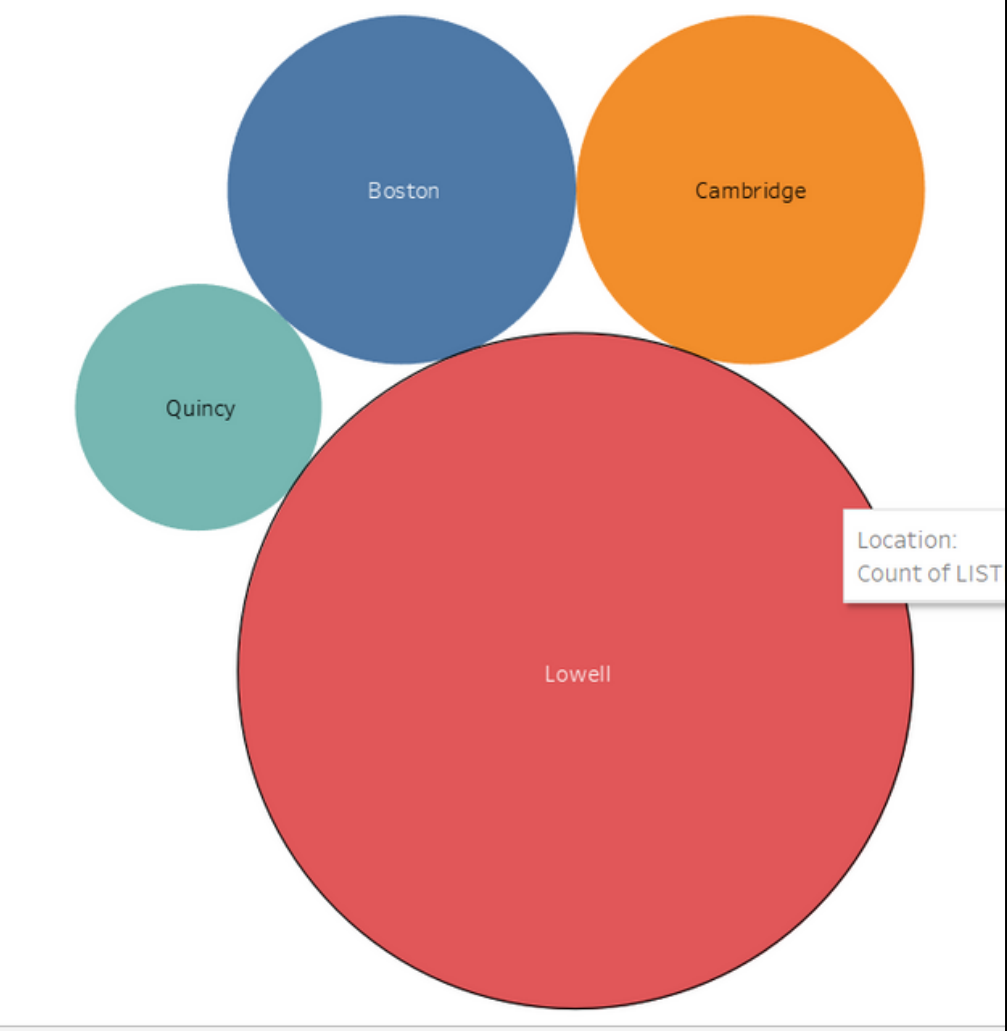
select * from [TRANSACTION] where Transaction_ID = 28;
```

	Transaction_ID ▾	Transaction_Date ▾	Listing_ID ▾	Transaction_Mode ▾	User_ID ▾	Supervisor_ID ▾
1	28	2024-03-01	Q17	DEBIT	112	10

DATA VISUALIZATION USING TABLEAU

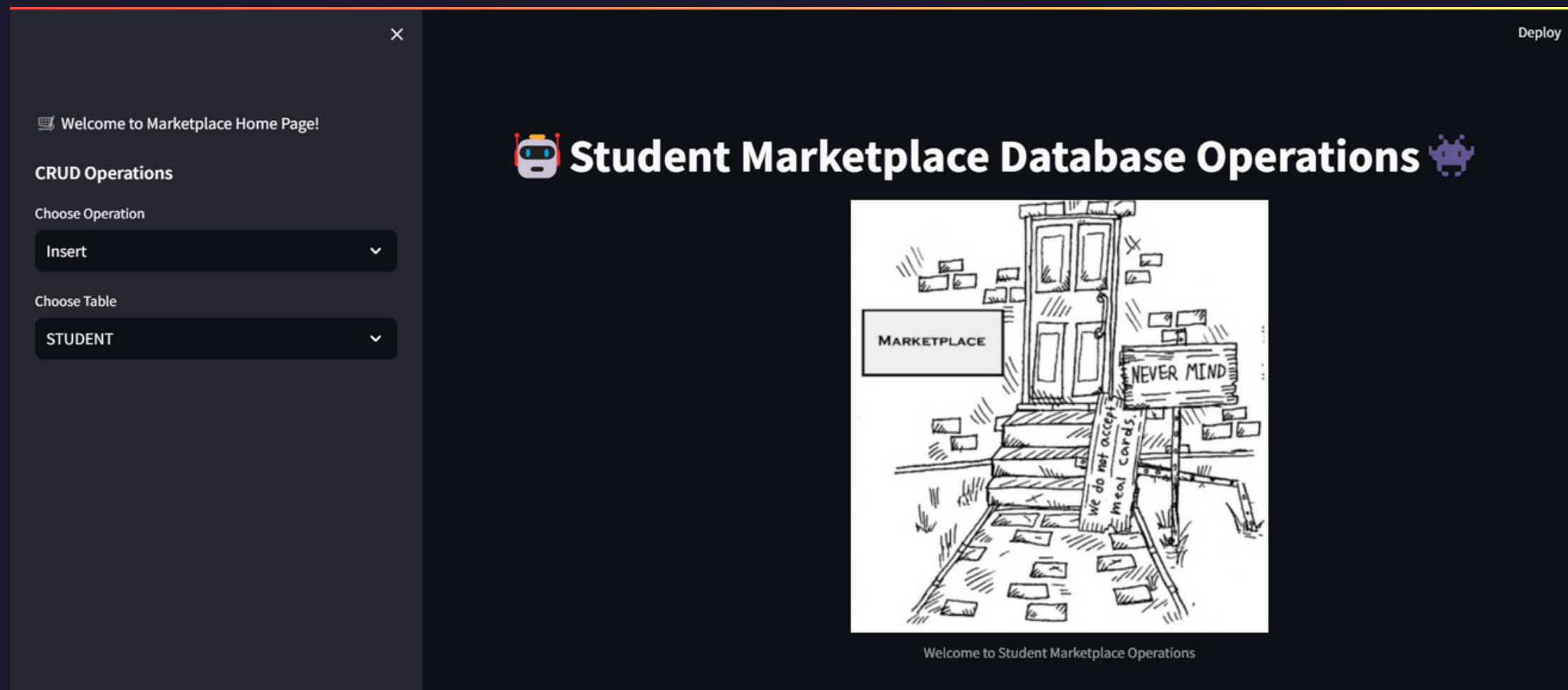


Listings per Location



GUI- STREAMLIT

HOME PAGE



GUI- STREAMLIT

CRUD PAGE

×

🛒 Welcome to Marketplace Home Page!

CRUD Operations

Choose Operation

Read

Choose Table

STUDENT

Deploy

⋮

Welcome to Student Marketplace Operations

	Student_ID	First_Name	Last_Name	Email
8	9	Henry	Anderson	henry.anderson@example.com
9	10	Sophia	Moore	sophia.moore@example.com
10	11	Liam	Taylor	liam.taylor@example.com
11	12	Ava	Miller	ava.miller@example.com
12	13	Noah	Brown	noah.brown@example.com
13	14	Emma	Wilson	emma.wilson@example.com
14	15	Ethan	Thomas	ethan.thomas@example.com
15	16	Isabella	Jones	isabella.jones@example.com
16	17	Mason	Clark	mason.clark@example.com
17	18	Sophia	Evans	sophia.evans@example.com
18	19	Jackson	Smith	jackson.smith@example.com

3x4

Made with Streamlit



**THANK
YOU**

