Inventory Management System

CS157A Project Report

Presented to

Narayan Balasubramanian
Department of Data Science

San Jose State University

In partial fulfillment
Of the requirements for the class
CS157A

By

Bharath Kumar A (018221268), Sania Bandekar (015948036)
December2024

Table of Contents

Commands to create tables

1. Users Table

```
CREATE TABLE Users (
    UserID INT AUTO_INCREMENT PRIMARY KEY,
    Username VARCHAR(100) NOT NULL,
    Password VARCHAR(255) NOT NULL, -- Store hashed passwords
    Email VARCHAR(255) UNIQUE NOT NULL,
    Phone VARCHAR(15),
    CreatedAt TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

---

2. Inventory Table

```
CREATE TABLE Inventory (
    ItemID INT AUTO_INCREMENT PRIMARY KEY,
    ItemName VARCHAR(255) NOT NULL,
    Quantity INT DEFAULT 0,
    Description TEXT,
    Category VARCHAR(100),
    CostPrice DECIMAL(10, 2) NOT NULL,
    SellingPrice DECIMAL(10, 2) NOT NULL,
    CreatedAt TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

---

3. Vendors Table

```
CREATE TABLE Vendors (
    VendorID INT AUTO_INCREMENT PRIMARY KEY,
    VendorName VARCHAR(255) NOT NULL,
    ItemSupplied INT NOT NULL,
    Price DECIMAL(10, 2) NOT NULL,
    ContactInfo VARCHAR(255),
    FOREIGN KEY (ItemSupplied) REFERENCES Inventory(ItemID)
);
```

---

4. Orders Table

```
CREATE TABLE Orders (
    OrderID INT AUTO_INCREMENT PRIMARY KEY,
    UserID INT NOT NULL,
    OrderDate TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    TotalAmount DECIMAL(10, 2) NOT NULL,
    Status ENUM('Pending', 'Paid', 'Delivered') DEFAULT 'Pending',
    DeliveryDate TIMESTAMP NULL,
    FOREIGN KEY (UserID) REFERENCES Users(UserID)
);
```

---

5. Sales Table

```
CREATE TABLE Sales (
    SaleID INT AUTO_INCREMENT PRIMARY KEY,
    ItemID INT NOT NULL,
    Quantity INT NOT NULL,
    SellingPrice DECIMAL(10, 2) NOT NULL,
    TotalRevenue DECIMAL(10, 2) NOT NULL,
    SaleDate TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (ItemID) REFERENCES Inventory(ItemID));
```

6. BrokenItems Table

```
CREATE TABLE BrokenItems (
    BrokenItemID INT AUTO_INCREMENT PRIMARY KEY,
    ItemID INT NOT NULL,
    Quantity INT NOT NULL,
    ReportedDate TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    Remarks TEXT,
    FOREIGN KEY (ItemID) REFERENCES Inventory(ItemID)
);
```

Queries to populate data

1. Users Table

```
INSERT INTO Users (Username, Password, Email, Phone, CreatedAt)
VALUES
('john_doe', 'password123', 'john.doe@example.com', '1234567890', NOW()),
('jane_smith', 'securepwd', 'jane.smith@example.com', '1234567891', NOW()),
('mike_jones', 'mypassword', 'mike.jones@example.com', '1234567892', NOW()),
('alice_wong', 'alice123', 'alice.wong@example.com', '1234567893', NOW()),
('bob_brown', 'bobsecure', 'bob.brown@example.com', '1234567894', NOW()),
('susan_lee', 'lee2024', 'susan.lee@example.com', '1234567895', NOW()),
('kevin_clark', 'kevinpass', 'kevin.clark@example.com', '1234567896', NOW()),
('emma_white', 'emma123', 'emma.white@example.com', '1234567897', NOW()),
('oliver_green', 'olivepass', 'oliver.green@example.com', '1234567898', NOW()),
('sophia_adams', 'sophia2024', 'sophia.adams@example.com', '1234567899', NOW());
```

---

2. Inventory Table

```
INSERT INTO Inventory (ItemName, Quantity, Description, Category, CostPrice,
SellingPrice, CreatedAt)
VALUES
('Laptop', 50, '15-inch screen laptop', 'Electronics', 500.00, 750.00, NOW()),
('Smartphone', 100, '64GB smartphone', 'Electronics', 200.00, 300.00, NOW()),
('Headphones', 200, 'Noise-cancelling headphones', 'Accessories', 50.00, 80.00, NOW()),
('Keyboard', 150, 'Mechanical keyboard', 'Accessories', 25.00, 40.00, NOW()),
('Mouse', 150, 'Wireless mouse', 'Accessories', 15.00, 25.00, NOW()),
('Monitor', 75, '24-inch monitor', 'Electronics', 150.00, 200.00, NOW()),
('Desk', 50, 'Office desk', 'Furniture', 100.00, 150.00, NOW()),
('Chair', 75, 'Ergonomic chair', 'Furniture', 120.00, 180.00, NOW()),
('Printer', 30, 'All-in-one printer', 'Electronics', 80.00, 120.00, NOW()),
('Webcam', 90, 'HD webcam', 'Electronics', 30.00, 50.00, NOW());
```

---

3. Vendors Table

INSERT INTO Vendors (VendorName, ItemSupplied, Price, ContactInfo)

VALUES

('ElectroMart', 1, 500.00, 'contact@electromart.com'),

('MobileHub', 2, 200.00, 'contact@mobilehub.com'),

('SoundPlus', 3, 50.00, 'contact@soundplus.com'),

('KeyboardKing', 4, 25.00, 'contact@keyboardking.com'),

('MouseWorld', 5, 15.00, 'contact@mouseworld.com'),

('DisplayExperts', 6, 150.00, 'contact@displayexperts.com'),

('OfficeFurnishings', 7, 100.00, 'contact@officefurnishings.com'),

('ComfortSeating', 8, 120.00, 'contact@comfortseating.com'),

('PrinterPros', 9, 80.00, 'contact@printerpros.com'),

('CamVision', 10, 30.00, 'contact@camvision.com');

---

4. Orders Table

INSERT INTO Orders (UserID, OrderDate, TotalAmount, Status, DeliveryDate)

VALUES

(1, NOW(), 750.00, 'Delivered', NOW()),

(2, NOW(), 300.00, 'Delivered', NOW()),

(3, NOW(), 80.00, 'Delivered', NOW()),

(4, NOW(), 40.00, 'Pending', NULL),

(5, NOW(), 25.00, 'Pending', NULL),

(6, NOW(), 200.00, 'Delivered', NOW()),

(7, NOW(), 150.00, 'Delivered', NOW()),

(8, NOW(), 180.00, 'Paid', NULL),

(9, NOW(), 120.00, 'Delivered', NOW()),

(10, NOW(), 50.00, 'Delivered', NOW());

---

5. OrderDetails Table

INSERT INTO OrderDetails (OrderID, ItemID, Quantity, UnitPrice, TotalPrice)

VALUES

(1, 1, 1, 750.00, 750.00),

(2, 2, 1, 300.00, 300.00),

(3, 3, 1, 80.00, 80.00),

(4, 4, 1, 40.00, 40.00),

(5, 5, 1, 25.00, 25.00),

(6, 6, 1, 200.00, 200.00),

(7, 7, 1, 150.00, 150.00),

(8, 8, 1, 180.00, 180.00),

(9, 9, 1, 120.00, 120.00),

(10, 10, 1, 50.00, 50.00);

---

6. Sales Table

INSERT INTO Sales (ItemID, Quantity, SellingPrice, TotalRevenue, SaleDate)

VALUES

(1, 1, 750.00, 750.00, NOW()),

(2, 1, 300.00, 300.00, NOW()),

(3, 1, 80.00, 80.00, NOW()),

(4, 1, 40.00, 40.00, NOW()),

(5, 1, 25.00, 25.00, NOW()),

(6, 1, 200.00, 200.00, NOW()),

(7, 1, 150.00, 150.00, NOW()),

(8, 1, 180.00, 180.00, NOW()),

(9, 1, 120.00, 120.00, NOW()),

(10, 1, 50.00, 50.00, NOW());

---

7. BrokenItems Table

INSERT INTO BrokenItems (ItemID, Quantity, ReportedDate, Remarks)

VALUES

(1, 2, NOW(), 'Screen damaged'),

(2, 1, NOW(), 'Battery issue'),

(3, 3, NOW(), 'Wire damaged'),

(4, 1, NOW(), 'Keys broken'),

(5, 2, NOW(), 'Scroll wheel issue'),

(6, 1, NOW(), 'Screen flickering'),

(7, 1, NOW(), 'Scratched surface'),

(8, 2, NOW(), 'Wheel base broken'),

(9, 1, NOW(), 'Paper jammed'),

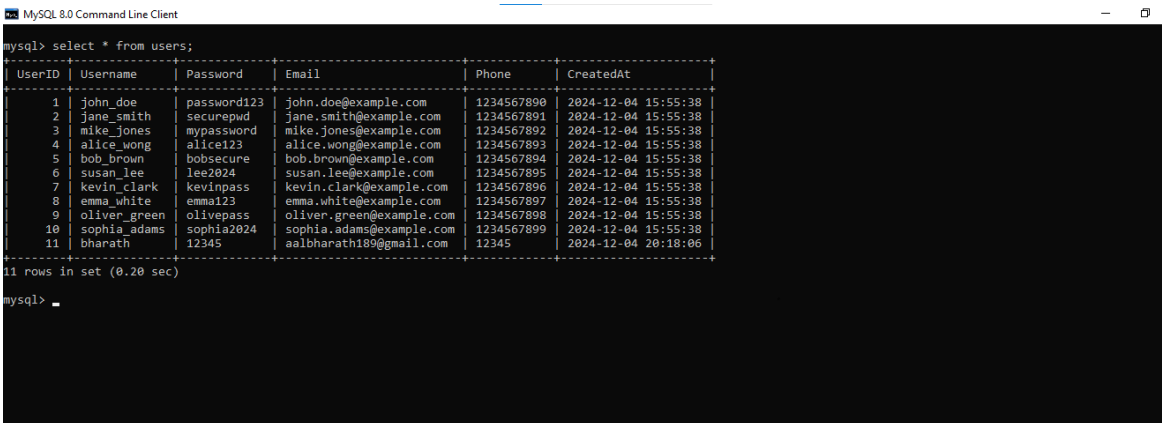
(10, 1, NOW(), 'Lens scratched');

Fetching the data

---

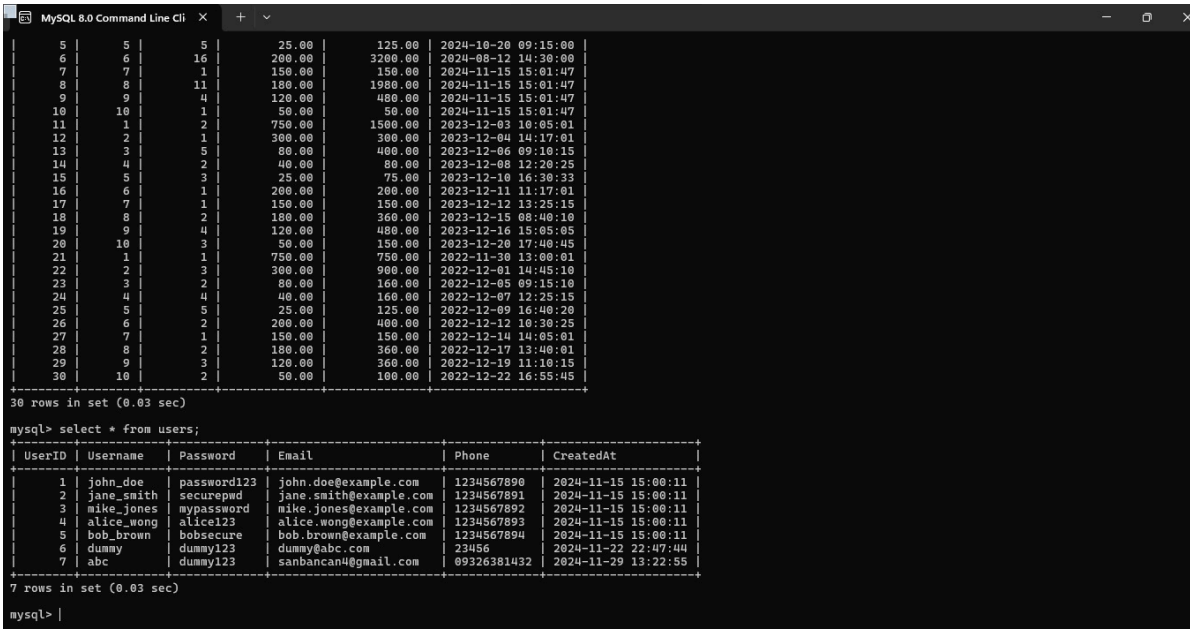
1. Users Table

SELECT * FROM Users;



```
MySQL 8.0 Command Line Client

mysql> select * from users;
+--------+-------------+-------------+-------------------------+------------+---------------------+
| UserID | Username    | Password    | Email                   | Phone      | CreatedAt           |
+--------+-------------+-------------+-------------------------+------------+---------------------+
|      1 | john_doe    | password123 | john.doe@example.com    | 1234567890 | 2024-12-04 15:55:38 |
|      2 | jane_smith  | securepwd   | jane.smith@example.com  | 1234567891 | 2024-12-04 15:55:38 |
|      3 | mike_jones  | mypassword  | mike.jones@example.com  | 1234567892 | 2024-12-04 15:55:38 |
|      4 | alice_wong  | alice123    | alice.wong@example.com  | 1234567893 | 2024-12-04 15:55:38 |
|      5 | bob_brown   | bobsecure   | bob.brown@example.com   | 1234567894 | 2024-12-04 15:55:38 |
|      6 | susan_lee   | lee2024     | susan.lee@example.com   | 1234567895 | 2024-12-04 15:55:38 |
|      7 | kevin_clark | kevinpass   | kevin.clark@example.com | 1234567896 | 2024-12-04 15:55:38 |
|      8 | emma_white  | emma123     | emma.white@example.com  | 1234567897 | 2024-12-04 15:55:38 |
|      9 | oliver_green| olivepass   | oliver.green@example.com| 1234567898 | 2024-12-04 15:55:38 |
|     10 | sophia_adams| sophia2024  | sophia.adams@example.com| 1234567899 | 2024-12-04 15:55:38 |
|     11 | bharath     | 12345       | aalbharath189@gmail.com | 12345      | 2024-12-04 20:18:06 |
+--------+-------------+-------------+-------------------------+------------+---------------------+
11 rows in set (0.20 sec)

mysql>
```

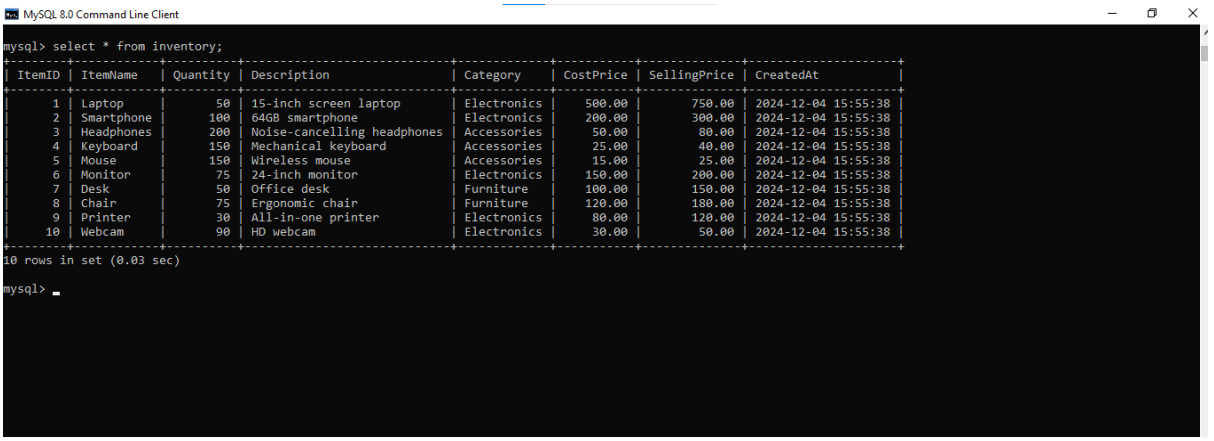
Current table:



```
MySQL 8.0 Command Line Cli

|      5 |      5 |        5 |        25.00 |       125.00 | 2024-10-20 09:15:00 |
|      6 |      6 |       16 |       200.00 |      3200.00 | 2024-08-12 14:30:00 |
|      7 |      7 |        1 |       150.00 |       150.00 | 2024-11-15 15:01:47 |
|      8 |      8 |       11 |       180.00 |      1980.00 | 2024-11-15 15:01:47 |
|      9 |      9 |        4 |       120.00 |       480.00 | 2024-11-15 15:01:47 |
|     10 |     10 |        1 |        50.00 |        50.00 | 2024-11-15 15:01:47 |
|     11 |      1 |        2 |       750.00 |      1500.00 | 2023-12-03 10:05:01 |
|     12 |      2 |        1 |       300.00 |       300.00 | 2023-12-04 14:17:01 |
|     13 |      3 |        5 |        80.00 |       400.00 | 2023-12-06 09:10:15 |
|     14 |      4 |        2 |        40.00 |        80.00 | 2023-12-08 12:20:25 |
|     15 |      5 |        3 |        25.00 |        75.00 | 2023-12-10 16:30:33 |
|     16 |      6 |        1 |       200.00 |       200.00 | 2023-12-11 11:17:01 |
|     17 |      7 |        1 |       150.00 |       150.00 | 2023-12-12 13:25:15 |
|     18 |      8 |        2 |       180.00 |       360.00 | 2023-12-15 08:40:10 |
|     19 |      9 |        4 |       120.00 |       480.00 | 2023-12-16 15:05:05 |
|     20 |     10 |        3 |        50.00 |       150.00 | 2023-12-20 17:40:45 |
|     21 |      1 |        1 |       750.00 |       750.00 | 2022-11-30 13:00:01 |
|     22 |      2 |        3 |       300.00 |       900.00 | 2022-12-01 14:45:10 |
|     23 |      3 |        2 |        80.00 |       160.00 | 2022-12-05 09:15:10 |
|     24 |      4 |        4 |        40.00 |       160.00 | 2022-12-07 12:25:15 |
|     25 |      5 |        5 |        25.00 |       125.00 | 2022-12-09 16:40:20 |
|     26 |      6 |        2 |       200.00 |       400.00 | 2022-12-12 10:30:25 |
|     27 |      7 |        1 |       150.00 |       150.00 | 2022-12-14 14:05:01 |
|     28 |      8 |        2 |       180.00 |       360.00 | 2022-12-17 13:40:01 |
|     29 |      9 |        3 |       120.00 |       360.00 | 2022-12-19 11:10:15 |
|     30 |     10 |        2 |        50.00 |       100.00 | 2022-12-22 16:55:45 |
+--------+--------+----------+--------------+--------------+---------------------+
30 rows in set (0.03 sec)

mysql> select * from users;
+--------+------------+-------------+-------------------------+-------------+---------------------+
| UserID | Username   | Password    | Email                   | Phone       | CreatedAt           |
+--------+------------+-------------+-------------------------+-------------+---------------------+
|      1 | john_doe   | password123 | john.doe@example.com    | 1234567890  | 2024-11-15 15:00:11 |
|      2 | jane_smith | securepwd   | jane.smith@example.com  | 1234567891  | 2024-11-15 15:00:11 |
|      3 | mike_jones | mypassword  | mike.jones@example.com  | 1234567892  | 2024-11-15 15:00:11 |
|      4 | alice_wong | alice123    | alice.wong@example.com  | 1234567893  | 2024-11-15 15:00:11 |
|      5 | bob_brown  | bobsecure   | bob.brown@example.com   | 1234567894  | 2024-11-15 15:00:11 |
|      6 | dummy      | dummy123    | dummy@abc.com           | 23456       | 2024-11-22 22:47:44 |
|      7 | abc        | dummy123    | sanbancan4@gmail.com    | 09326381432 | 2024-11-29 13:22:55 |
+--------+------------+-------------+-------------------------+-------------+---------------------+
7 rows in set (0.03 sec)

mysql>
```

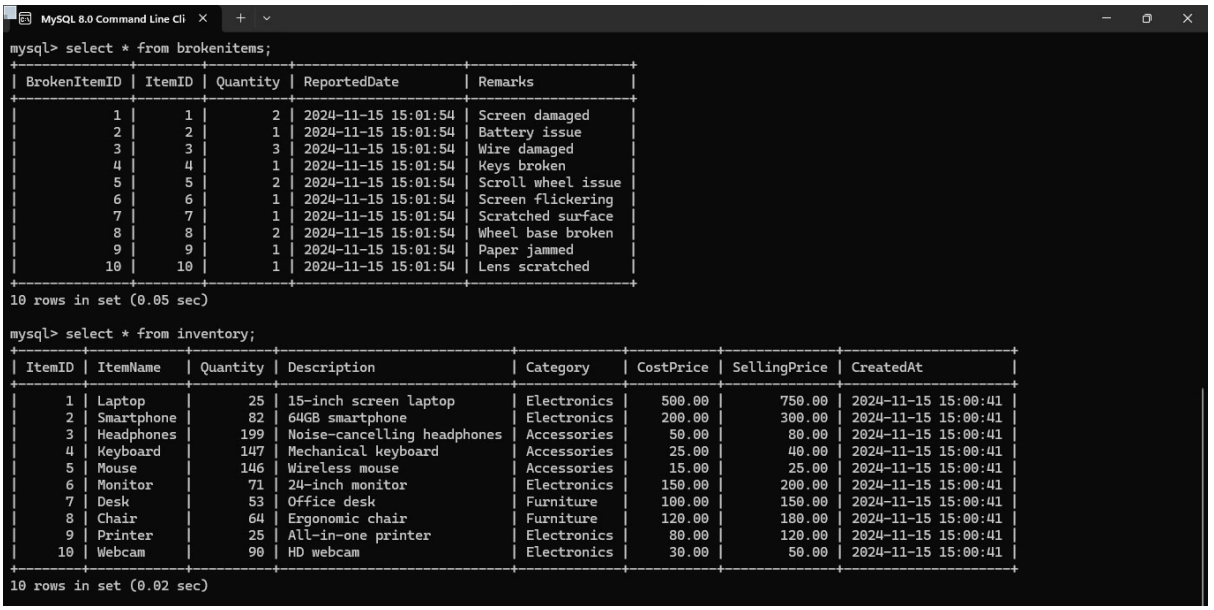
2. Inventory Table

SELECT * FROM Inventory;



```
MySQL 8.0 Command Line Client

mysql> select * from inventory;
+--------+------------+----------+-----------------------------+-------------+-----------+--------------+---------------------+
| ItemID | ItemName   | Quantity | Description                 | Category    | CostPrice | SellingPrice | CreatedAt           |
+--------+------------+----------+-----------------------------+-------------+-----------+--------------+---------------------+
|      1 | Laptop     |       50 | 15-inch screen laptop       | Electronics |    500.00 |       750.00 | 2024-12-04 15:55:38 |
|      2 | Smartphone |      100 | 64GB smartphone             | Electronics |    200.00 |       300.00 | 2024-12-04 15:55:38 |
|      3 | Headphones |      200 | Noise-cancelling headphones | Accessories |     50.00 |        80.00 | 2024-12-04 15:55:38 |
|      4 | Keyboard   |      150 | Mechanical keyboard         | Accessories |     25.00 |        40.00 | 2024-12-04 15:55:38 |
|      5 | Mouse      |      150 | Wireless mouse              | Accessories |     15.00 |        25.00 | 2024-12-04 15:55:38 |
|      6 | Monitor    |       75 | 24-inch monitor             | Electronics |    150.00 |       200.00 | 2024-12-04 15:55:38 |
|      7 | Desk       |       50 | Office desk                 | Furniture   |    100.00 |       150.00 | 2024-12-04 15:55:38 |
|      8 | Chair      |       75 | Ergonomic chair             | Furniture   |    120.00 |       180.00 | 2024-12-04 15:55:38 |
|      9 | Printer    |       30 | All-in-one printer          | Electronics |     80.00 |       120.00 | 2024-12-04 15:55:38 |
|     10 | Webcam     |       90 | HD webcam                   | Electronics |     30.00 |        50.00 | 2024-12-04 15:55:38 |
+--------+------------+----------+-----------------------------+-------------+-----------+--------------+---------------------+
10 rows in set (0.03 sec)

mysql>
```

current table:



```
MySQL 8.0 Command Line Cli

mysql> select * from brokenitems;
+--------------+--------+----------+---------------------+--------------------+
| BrokenItemID | ItemID | Quantity | ReportedDate        | Remarks            |
+--------------+--------+----------+---------------------+--------------------+
|            1 |      1 |        2 | 2024-11-15 15:01:54 | Screen damaged     |
|            2 |      2 |        1 | 2024-11-15 15:01:54 | Battery issue      |
|            3 |      3 |        3 | 2024-11-15 15:01:54 | Wire damaged       |
|            4 |      4 |        1 | 2024-11-15 15:01:54 | Keys broken        |
|            5 |      5 |        2 | 2024-11-15 15:01:54 | Scroll wheel issue |
|            6 |      6 |        1 | 2024-11-15 15:01:54 | Screen flickering  |
|            7 |      7 |        1 | 2024-11-15 15:01:54 | Scratched surface  |
|            8 |      8 |        2 | 2024-11-15 15:01:54 | Wheel base broken  |
|            9 |      9 |        1 | 2024-11-15 15:01:54 | Paper jammed       |
|           10 |     10 |        1 | 2024-11-15 15:01:54 | Lens scratched     |
+--------------+--------+----------+---------------------+--------------------+
10 rows in set (0.05 sec)

mysql> select * from inventory;
+--------+------------+----------+-----------------------------+-------------+-----------+--------------+---------------------+
| ItemID | ItemName   | Quantity | Description                 | Category    | CostPrice | SellingPrice | CreatedAt           |
+--------+------------+----------+-----------------------------+-------------+-----------+--------------+---------------------+
|      1 | Laptop     |       25 | 15-inch screen laptop       | Electronics |    500.00 |       750.00 | 2024-11-15 15:00:41 |
|      2 | Smartphone |       82 | 64GB smartphone             | Electronics |    200.00 |       300.00 | 2024-11-15 15:00:41 |
|      3 | Headphones |      199 | Noise-cancelling headphones | Accessories |     50.00 |        80.00 | 2024-11-15 15:00:41 |
|      4 | Keyboard   |      147 | Mechanical keyboard         | Accessories |     25.00 |        40.00 | 2024-11-15 15:00:41 |
|      5 | Mouse      |      146 | Wireless mouse              | Accessories |     15.00 |        25.00 | 2024-11-15 15:00:41 |
|      6 | Monitor    |       71 | 24-inch monitor             | Electronics |    150.00 |       200.00 | 2024-11-15 15:00:41 |
|      7 | Desk       |       53 | Office desk                 | Furniture   |    100.00 |       150.00 | 2024-11-15 15:00:41 |
|      8 | Chair      |       64 | Ergonomic chair             | Furniture   |    120.00 |       180.00 | 2024-11-15 15:00:41 |
|      9 | Printer    |       25 | All-in-one printer          | Electronics |     80.00 |       120.00 | 2024-11-15 15:00:41 |
|     10 | Webcam     |       90 | HD webcam                   | Electronics |     30.00 |        50.00 | 2024-11-15 15:00:41 |
+--------+------------+----------+-----------------------------+-------------+-----------+--------------+---------------------+
10 rows in set (0.02 sec)
```

3. Vendors Table

SELECT * FROM Vendors;

```
mysql> select * from vendors;
+----------+--------------------+--------------+--------+--------------------------------+
| VendorID | VendorName         | ItemSupplied | Price  | ContactInfo                    |
+----------+--------------------+--------------+--------+--------------------------------+
|        1 | ElectroMart        |            1 | 500.00 | contact@electromart.com        |
|        2 | MobileHub          |            2 | 200.00 | contact@mobilehub.com          |
|        3 | SoundPlus          |            3 |  50.00 | contact@soundplus.com          |
|        4 | KeyboardKing       |            4 |  25.00 | contact@keyboardking.com       |
|        5 | MouseWorld         |            5 |  15.00 | contact@mouseworld.com         |
|        6 | DisplayExperts     |            6 | 150.00 | contact@displayexperts.com     |
|        7 | OfficeFurnishings  |            7 | 100.00 | contact@officefurnishings.com  |
|        8 | ComfortSeating     |            8 | 120.00 | contact@comfortseating.com     |
|        9 | PrinterPros        |            9 |  80.00 | contact@printerpros.com        |
|       10 | CamVision          |           10 |  30.00 | contact@camvision.com          |
+----------+--------------------+--------------+--------+--------------------------------+
10 rows in set (0.02 sec)

mysql>
```

current table:

```
MySQL 8.0 Command Line Cli
|     22 |      2 |        3 |      300.00 |       900.00 | 2022-12-01 14:45:10 |
|     23 |      3 |        2 |       80.00 |       160.00 | 2022-12-05 09:15:10 |
|     24 |      4 |        4 |       40.00 |       160.00 | 2022-12-07 12:25:15 |
|     25 |      5 |        5 |       25.00 |       125.00 | 2022-12-09 16:40:20 |
|     26 |      6 |        2 |      200.00 |       400.00 | 2022-12-12 10:30:25 |
|     27 |      7 |        1 |      150.00 |       150.00 | 2022-12-14 14:05:01 |
|     28 |      8 |        2 |      180.00 |       360.00 | 2022-12-17 13:40:01 |
|     29 |      9 |        3 |      120.00 |       360.00 | 2022-12-19 11:10:15 |
|     30 |     10 |        2 |       50.00 |       100.00 | 2022-12-22 16:55:45 |
+--------+--------+----------+-------------+--------------+---------------------+
30 rows in set (0.03 sec)

mysql> select * from users;
+--------+-----------+-------------+-----------------------+-------------+---------------------+
| UserID | Username  | Password    | Email                 | Phone       | CreatedAt           |
+--------+-----------+-------------+-----------------------+-------------+---------------------+
|      1 | john_doe  | password123 | john.doe@example.com  | 1234567890  | 2024-11-15 15:00:11 |
|      2 | jane_smith| securepwd   | jane.smith@example.com| 1234567891  | 2024-11-15 15:00:11 |
|      3 | mike_jones| mypassword  | mike.jones@example.com| 1234567892  | 2024-11-15 15:00:11 |
|      4 | alice_wong| alice123    | alice.wong@example.com| 1234567893  | 2024-11-15 15:00:11 |
|      5 | bob_brown | bobsecure   | bob.brown@example.com | 1234567894  | 2024-11-15 15:00:11 |
|      6 | dummy     | dummy123    | dummy@abc.com         | 23456       | 2024-11-22 22:47:44 |
|      7 | abc       | dummy123    | sanbancan4@gmail.com  | 09326381432 | 2024-11-29 13:22:55 |
+--------+-----------+-------------+-----------------------+-------------+---------------------+
7 rows in set (0.03 sec)

mysql> select * from vendors;
+----------+--------------------+--------+--------+--------------------------------+------------------+
| VendorID | VendorName         | ItemID | Price  | ContactInfo                    | QuantitySupplied |
+----------+--------------------+--------+--------+--------------------------------+------------------+
|        1 | ElectroMart        |      1 | 500.00 | contact@electromart.com        |               45 |
|        2 | MobileHub          |      2 | 200.00 | contact@mobilehub.com          |               10 |
|        3 | SoundPlus          |      3 |  50.00 | contact@soundplus.com          |               10 |
|        4 | KeyboardKing       |      4 |  25.00 | contact@keyboardking.com       |               10 |
|        5 | MouseWorld         |      5 |  15.00 | contact@mouseworld.com         |               10 |
|        6 | DisplayExperts     |      6 | 150.00 | contact@displayexperts.com     |               10 |
|        7 | OfficeFurnishings  |      7 | 100.00 | contact@officefurnishings.com  |               10 |
|        8 | ComfortSeating     |      8 | 120.00 | contact@comfortseating.com     |               10 |
|        9 | PrinterPros        |      9 |  80.00 | contact@printerpros.com        |               10 |
|       10 | CamVision          |     10 |  30.00 | contact@camvision.com          |               10 |
+----------+--------------------+--------+--------+--------------------------------+------------------+
10 rows in set (0.00 sec)
```

4. Orders Table

SELECT * FROM Orders;

```
mysql> select * from orders;
+---------+--------+---------------------+-------------+-----------+---------------------+
| OrderID | UserID | OrderDate           | TotalAmount | Status    | DeliveryDate        |
+---------+--------+---------------------+-------------+-----------+---------------------+
|       1 |      1 | 2024-12-04 15:55:38 |      750.00 | Delivered | 2024-12-04 15:55:38 |
|       2 |      2 | 2024-12-04 15:55:38 |      300.00 | Delivered | 2024-12-04 15:55:38 |
|       3 |      3 | 2024-12-04 15:55:38 |       80.00 | Delivered | 2024-12-04 15:55:38 |
|       4 |      4 | 2024-12-04 15:55:38 |       40.00 | Pending   | NULL                |
|       5 |      5 | 2024-12-04 15:55:38 |       25.00 | Pending   | NULL                |
|       6 |      6 | 2024-12-04 15:55:38 |      200.00 | Delivered | 2024-12-04 15:55:38 |
|       7 |      7 | 2024-12-04 15:55:38 |      150.00 | Delivered | 2024-12-04 15:55:38 |
|       8 |      8 | 2024-12-04 15:55:38 |      180.00 | Paid      | NULL                |
|       9 |      9 | 2024-12-04 15:55:38 |      120.00 | Delivered | 2024-12-04 15:55:38 |
|      10 |     10 | 2024-12-04 15:55:38 |       50.00 | Delivered | 2024-12-04 15:55:38 |
+---------+--------+---------------------+-------------+-----------+---------------------+
10 rows in set (0.02 sec)

mysql>
```

current table:

```
MySQL 8.0 Command Line Cli
|     10 | Webcam     |       90 | HD webcam                 | Electronics |     30.00 |       50.00 | 2024-11-15 15:00:41 |
+--------+------------+----------+---------------------------+-------------+-----------+--------------+---------------------+
10 rows in set (0.02 sec)

mysql> select * from orders;
+---------+--------+---------------------+-------------+--------+----------+-----------+
| OrderID | UserID | OrderDate           | TotalAmount | ItemID | Quantity | status    |
+---------+--------+---------------------+-------------+--------+----------+-----------+
|       1 |      1 | 2024-11-15 15:01:22 |      750.00 |      1 |        2 | purchased |
|       2 |      2 | 2024-11-15 15:01:22 |      300.00 |      2 |        1 | purchased |
|       3 |      3 | 2024-11-15 15:01:22 |       80.00 |      3 |        3 | purchased |
|       4 |      4 | 2024-11-15 15:01:22 |       40.00 |      4 |        2 | purchased |
|       5 |      5 | 2024-11-15 15:01:22 |       25.00 |      5 |        1 | purchased |
|       6 |      6 | 2024-11-15 15:01:22 |      200.00 |      6 |        2 | purchased |
|       7 |      7 | 2024-11-15 15:01:22 |      150.00 |      7 |        1 | purchased |
|       8 |      8 | 2024-11-15 15:01:22 |      180.00 |      8 |        3 | purchased |
|       9 |      9 | 2024-11-15 15:01:22 |      120.00 |      9 |        1 | purchased |
|      10 |      6 | 2024-11-22 22:51:49 |      400.00 |      6 |        2 | purchased |
|      11 |      6 | 2024-11-22 22:53:19 |      400.00 |      6 |        2 | sold      |
|      12 |      7 | 2024-11-22 23:08:10 |      900.00 |      2 |        3 | purchased |
|      13 |      7 | 2024-11-25 14:59:33 |      900.00 |      2 |        3 | purchased |
|      14 |      7 | 2024-11-28 14:16:14 |      900.00 |      2 |        3 | sold      |
|      15 |      7 | 2024-11-28 14:16:29 |      900.00 |      2 |        3 | purchased |
|      16 |      7 | 2024-11-28 14:20:34 |      900.00 |      2 |        3 | sold      |
|      17 |      7 | 2024-11-29 13:23:46 |      900.00 |      2 |        3 | purchased |
|      18 |      7 | 2024-11-29 13:24:14 |      360.00 |      9 |        3 | purchased |
|      19 |      7 | 2024-11-29 13:27:08 |     1800.00 |      6 |        9 | sold      |
|      20 |      3 | 2024-12-02 16:33:54 |      120.00 |      4 |        3 | purchased |
|      21 |      3 | 2024-12-02 16:35:16 |      240.00 |      3 |        3 | sold      |
|      22 |      7 | 2024-12-02 16:52:44 |     4500.00 |      1 |        6 | purchased |
|      23 |      7 | 2024-12-02 16:53:29 |     4500.00 |      1 |        6 | sold      |
+---------+--------+---------------------+-------------+--------+----------+-----------+
23 rows in set (0.04 sec)

mysql>
```

5. Sales Table

SELECT * FROM Sales;

```
mysql> select * from sales;
+--------+--------+----------+--------------+--------------+---------------------+
| SaleID | ItemID | Quantity | SellingPrice | TotalRevenue | SaleDate            |
+--------+--------+----------+--------------+--------------+---------------------+
|      1 |      1 |        1 |       750.00 |       750.00 | 2024-12-04 15:55:38 |
|      2 |      2 |        1 |       300.00 |       300.00 | 2024-12-04 15:55:38 |
|      3 |      3 |        1 |        80.00 |        80.00 | 2024-12-04 15:55:38 |
|      4 |      4 |        1 |        40.00 |        40.00 | 2024-12-04 15:55:38 |
|      5 |      5 |        1 |        25.00 |        25.00 | 2024-12-04 15:55:38 |
|      6 |      6 |        1 |       200.00 |       200.00 | 2024-12-04 15:55:38 |
|      7 |      7 |        1 |       150.00 |       150.00 | 2024-12-04 15:55:38 |
|      8 |      8 |        1 |       180.00 |       180.00 | 2024-12-04 15:55:38 |
|      9 |      9 |        1 |       120.00 |       120.00 | 2024-12-04 15:55:38 |
|     10 |     10 |        1 |        50.00 |        50.00 | 2024-12-04 15:55:38 |
+--------+--------+----------+--------------+--------------+---------------------+
10 rows in set (0.02 sec)

mysql>
```

current table:

```
MySQL 8.0 Command Line Cli
+---------+--------+----------------------+-------------+--------+----------+-----------+
23 rows in set (0.04 sec)

mysql> select * from sales;
+--------+--------+----------+--------------+--------------+---------------------+
| SaleID | ItemID | Quantity | SellingPrice | TotalRevenue | SaleDate            |
+--------+--------+----------+--------------+--------------+---------------------+
|      1 |      1 |       10 |       750.00 |      7800.00 | 2024-11-15 15:01:47 |
|      2 |      2 |       22 |       300.00 |      6600.00 | 2024-08-12 14:30:00 |
|      3 |      3 |        2 |        80.00 |       160.00 | 2024-10-20 09:15:00 |
|      4 |      4 |        4 |        40.00 |       160.00 | 2024-10-20 09:15:00 |
|      5 |      5 |        5 |        25.00 |       125.00 | 2024-10-20 09:15:00 |
|      6 |      6 |       16 |       200.00 |      3200.00 | 2024-08-12 14:30:00 |
|      7 |      7 |        1 |       150.00 |       150.00 | 2024-11-15 15:01:47 |
|      8 |      8 |       11 |       180.00 |      1980.00 | 2024-11-15 15:01:47 |
|      9 |      9 |        4 |       120.00 |       480.00 | 2024-11-15 15:01:47 |
|     10 |     10 |        1 |        50.00 |        50.00 | 2024-11-15 15:01:47 |
|     11 |      1 |        2 |       750.00 |      1500.00 | 2023-12-03 10:05:01 |
|     12 |      2 |        1 |       300.00 |       300.00 | 2023-12-04 14:17:01 |
|     13 |      3 |        5 |        80.00 |       400.00 | 2023-12-06 09:10:15 |
|     14 |      4 |        2 |        40.00 |        80.00 | 2023-12-08 12:20:25 |
|     15 |      5 |        3 |        25.00 |        75.00 | 2023-12-10 16:30:33 |
|     16 |      6 |        1 |       200.00 |       200.00 | 2023-12-11 11:17:01 |
|     17 |      7 |        1 |       150.00 |       150.00 | 2023-12-12 13:25:15 |
|     18 |      8 |        2 |       180.00 |       360.00 | 2023-12-15 08:40:10 |
|     19 |      9 |        4 |       120.00 |       480.00 | 2023-12-16 15:05:05 |
|     20 |     10 |        3 |        50.00 |       150.00 | 2023-12-20 17:40:45 |
|     21 |      1 |        1 |       750.00 |       750.00 | 2022-11-30 13:00:01 |
|     22 |      2 |        3 |       300.00 |       900.00 | 2022-12-01 14:45:10 |
|     23 |      3 |        2 |        80.00 |       160.00 | 2022-12-05 09:15:10 |
|     24 |      4 |        4 |        40.00 |       160.00 | 2022-12-07 12:25:15 |
|     25 |      5 |        5 |        25.00 |       125.00 | 2022-12-09 16:40:20 |
|     26 |      6 |        2 |       200.00 |       400.00 | 2022-12-12 10:30:25 |
|     27 |      7 |        1 |       150.00 |       150.00 | 2022-12-14 14:05:01 |
|     28 |      8 |        2 |       180.00 |       360.00 | 2022-12-17 13:40:01 |
|     29 |      9 |        3 |       120.00 |       360.00 | 2022-12-19 11:10:15 |
|     30 |     10 |        2 |        50.00 |       100.00 | 2022-12-22 16:55:45 |
+--------+--------+----------+--------------+--------------+---------------------+
30 rows in set (0.03 sec)

mysql>
```

6. BrokenItems Table

SELECT * FROM BrokenItems;

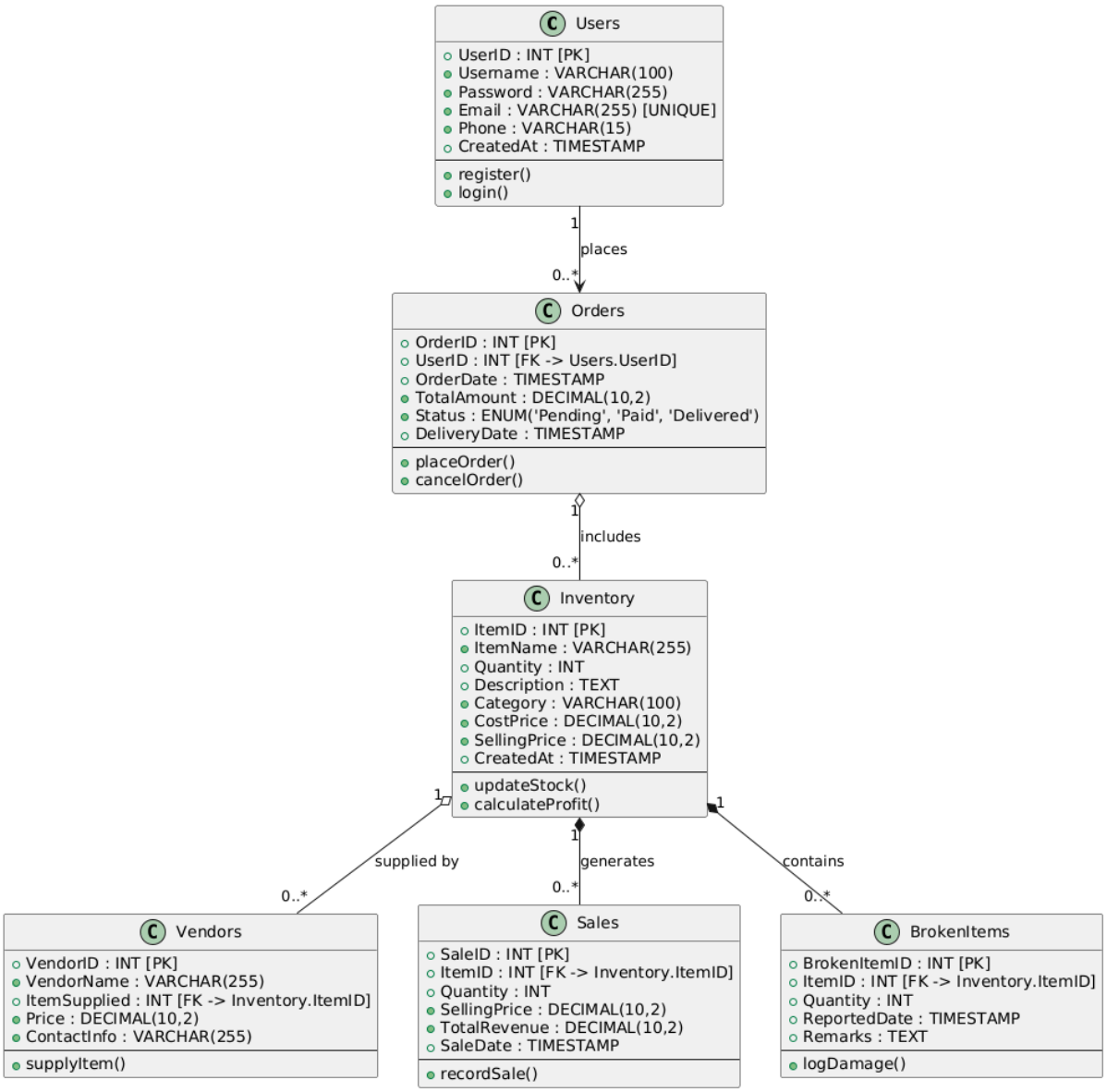
```
mysql> SELECT * FROM BrokenItems;
+--------------+--------+----------+---------------------+--------------------+
| BrokenItemID | ItemID | Quantity | ReportedDate        | Remarks            |
+--------------+--------+----------+---------------------+--------------------+
|            1 |      1 |        2 | 2024-12-04 15:55:38 | Screen damaged     |
|            2 |      2 |        1 | 2024-12-04 15:55:38 | Battery issue      |
|            3 |      3 |        3 | 2024-12-04 15:55:38 | Wire damaged       |
|            4 |      4 |        1 | 2024-12-04 15:55:38 | Keys broken        |
|            5 |      5 |        2 | 2024-12-04 15:55:38 | Scroll wheel issue |
|            6 |      6 |        1 | 2024-12-04 15:55:38 | Screen flickering  |
|            7 |      7 |        1 | 2024-12-04 15:55:38 | Scratched surface  |
|            8 |      8 |        2 | 2024-12-04 15:55:38 | Wheel base broken  |
|            9 |      9 |        1 | 2024-12-04 15:55:38 | Paper jammed       |
|           10 |     10 |        1 | 2024-12-04 15:55:38 | Lens scratched     |
+--------------+--------+----------+---------------------+--------------------+
10 rows in set (0.01 sec)

mysql>
```

currenttable:

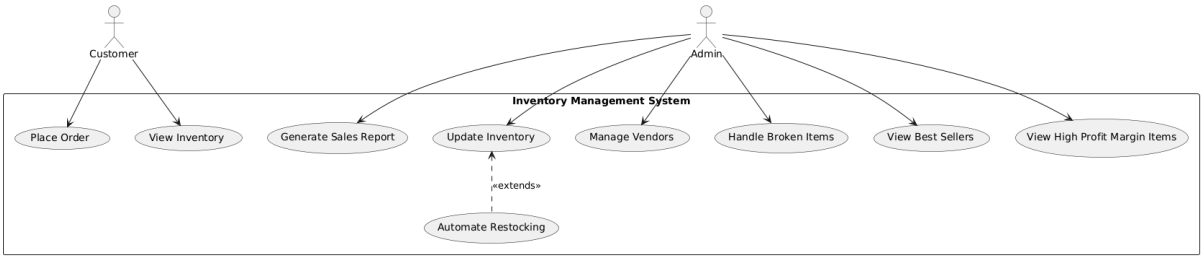
```
MySQL 8.0 Command Line Cli
10 rows in set (0.05 sec)

mysql> use inventory_management;
Database changed
mysql> show tables;
+--------------------------------+
| Tables_in_inventory_management |
+--------------------------------+
| brokenitems                    |
| inventory                      |
| orders                         |
| sales                          |
| users                          |
| vendors                        |
+--------------------------------+
6 rows in set (0.00 sec)

mysql> select * from brokenitems;
+--------------+--------+----------+---------------------+--------------------+
| BrokenItemID | ItemID | Quantity | ReportedDate        | Remarks            |
+--------------+--------+----------+---------------------+--------------------+
|            1 |      1 |        2 | 2024-11-15 15:01:54 | Screen damaged     |
|            2 |      2 |        1 | 2024-11-15 15:01:54 | Battery issue      |
|            3 |      3 |        3 | 2024-11-15 15:01:54 | Wire damaged       |
|            4 |      4 |        1 | 2024-11-15 15:01:54 | Keys broken        |
|            5 |      5 |        2 | 2024-11-15 15:01:54 | Scroll wheel issue |
|            6 |      6 |        1 | 2024-11-15 15:01:54 | Screen flickering  |
|            7 |      7 |        1 | 2024-11-15 15:01:54 | Scratched surface  |
|            8 |      8 |        2 | 2024-11-15 15:01:54 | Wheel base broken  |
|            9 |      9 |        1 | 2024-11-15 15:01:54 | Paper jammed       |
|           10 |     10 |        1 | 2024-11-15 15:01:54 | Lens scratched     |
+--------------+--------+----------+---------------------+--------------------+
10 rows in set (0.05 sec)

mysql>
```

UML DIAGRAMS

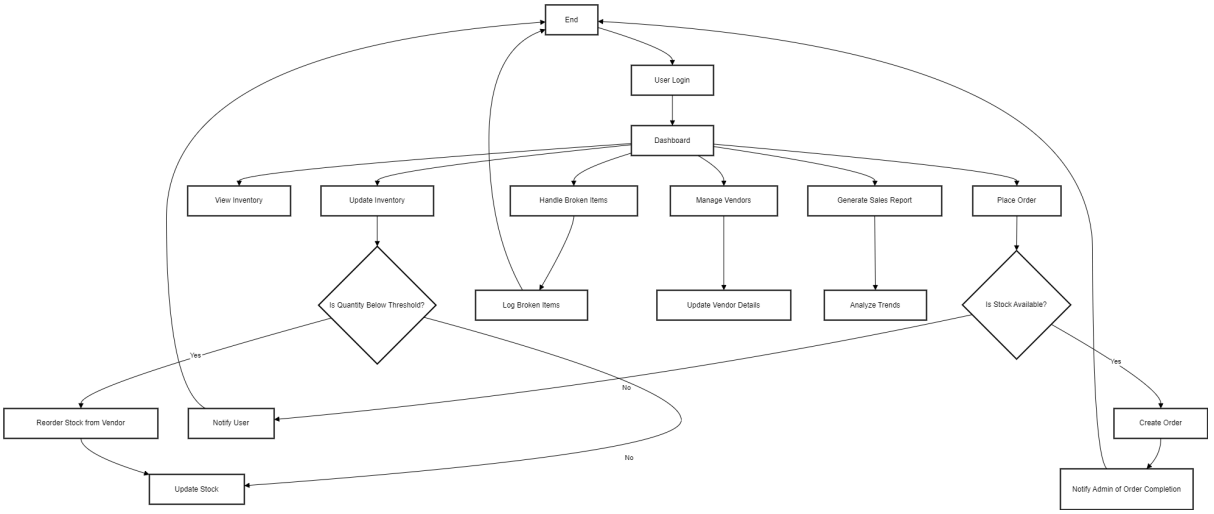
1. Class Diagram



2.Use Case



3.Flowchart

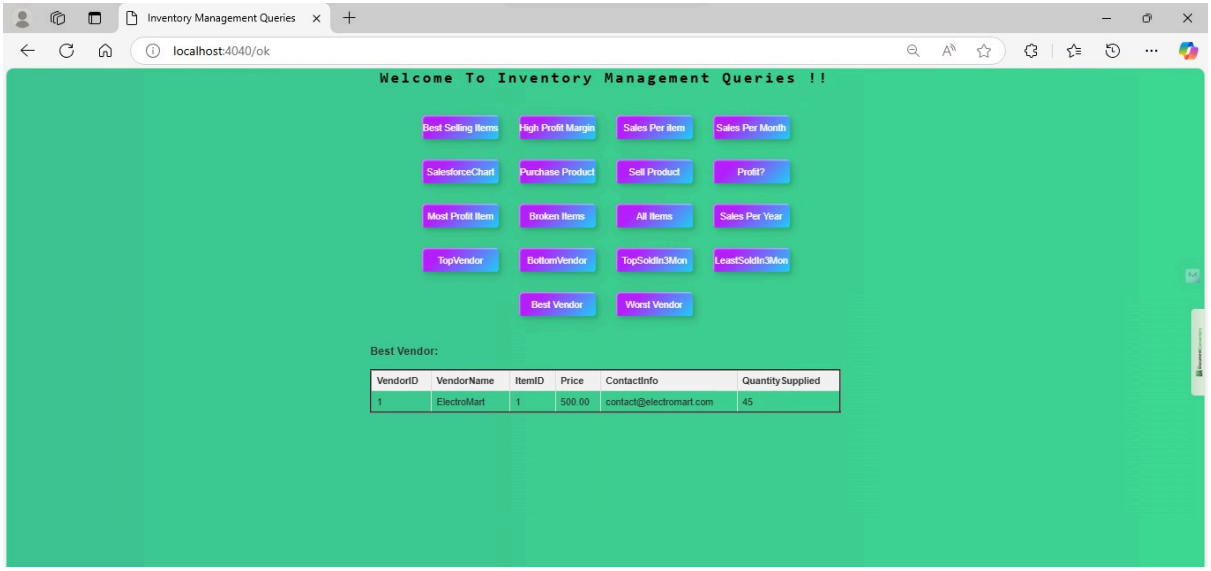


Executing Queries

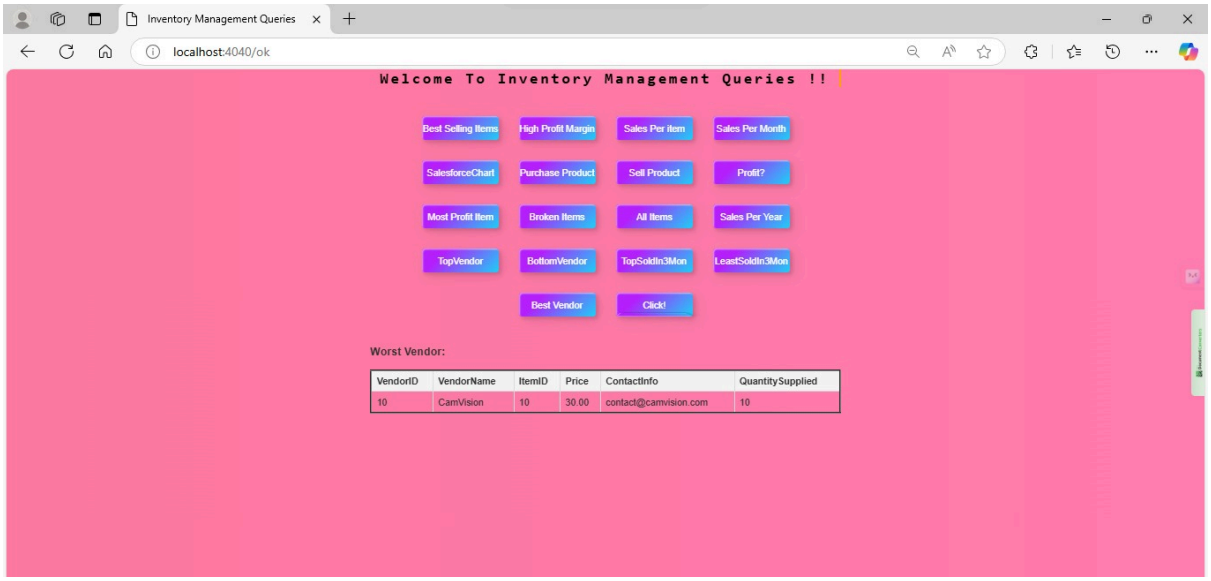
1)Sales for the month of November 2024. Is this an increase over the previous year - November 2023. Is it an increase over the previous month - October 2024

2)Identify the best seller. Similarly, identify the worst seller

Best Vendor:

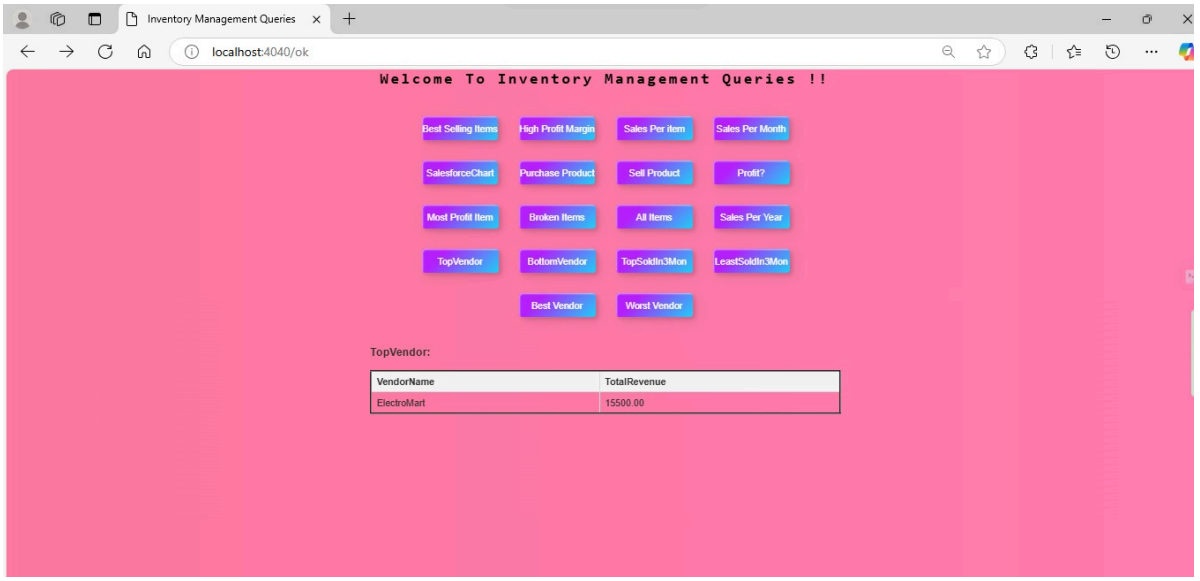


Worst Vendor:

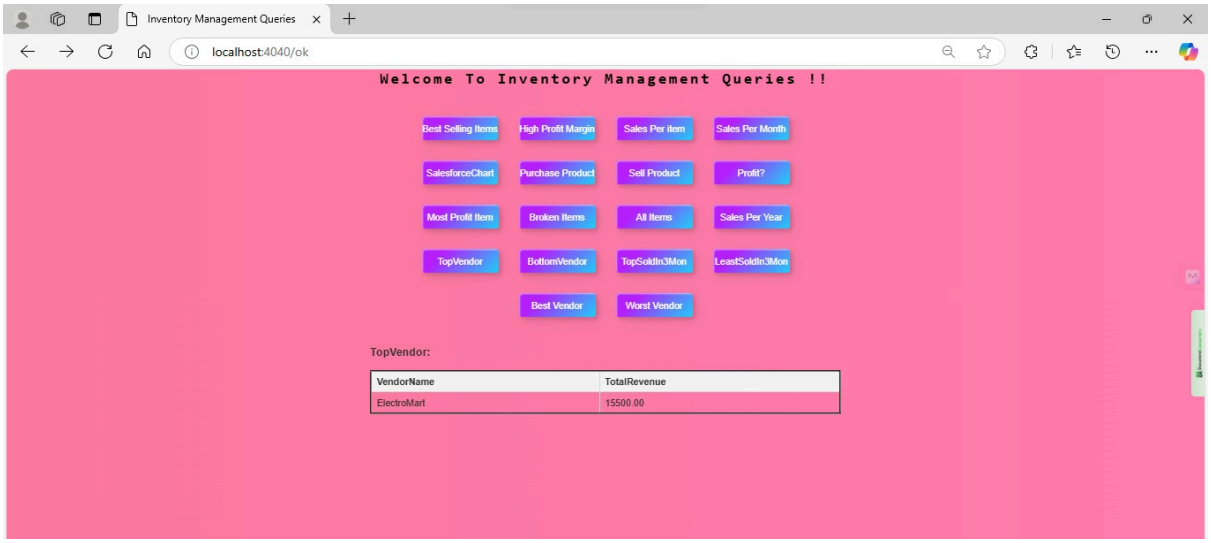


3)Identify the most profitable vendor and the least profitable vendor

Most Profitable Vendor:



Least Profitable Vendor:



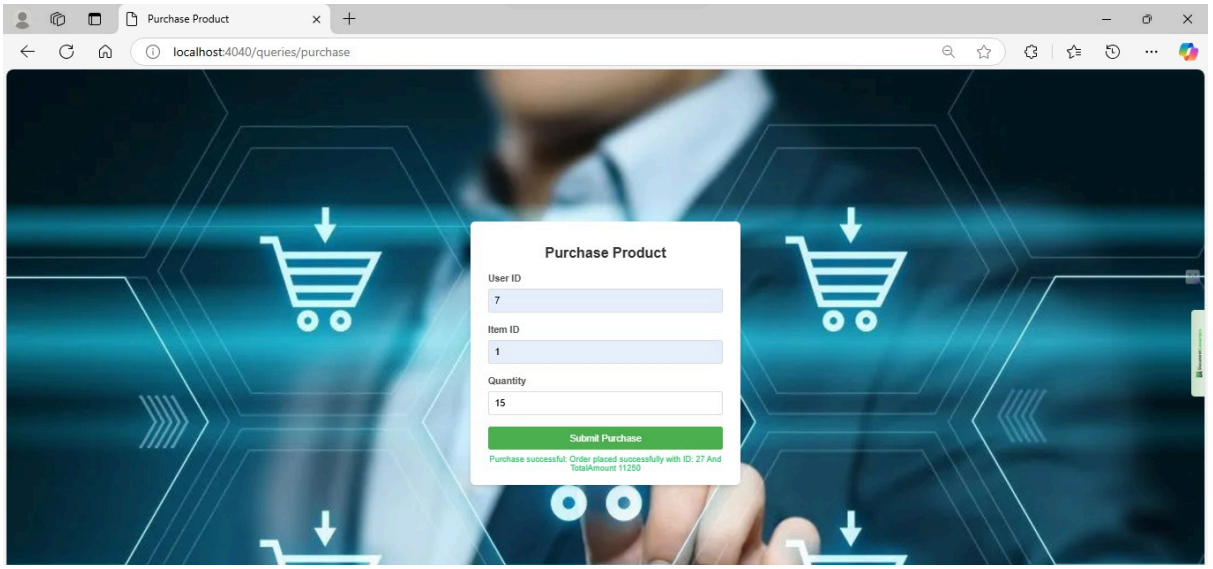
4)As soon as the inventory hits a threshold (say 5 pieces - actually this could be different for different items) an order for this item should be placed with the respective vendor. This is so that you never run out of items.

Inventory Before purchasing:

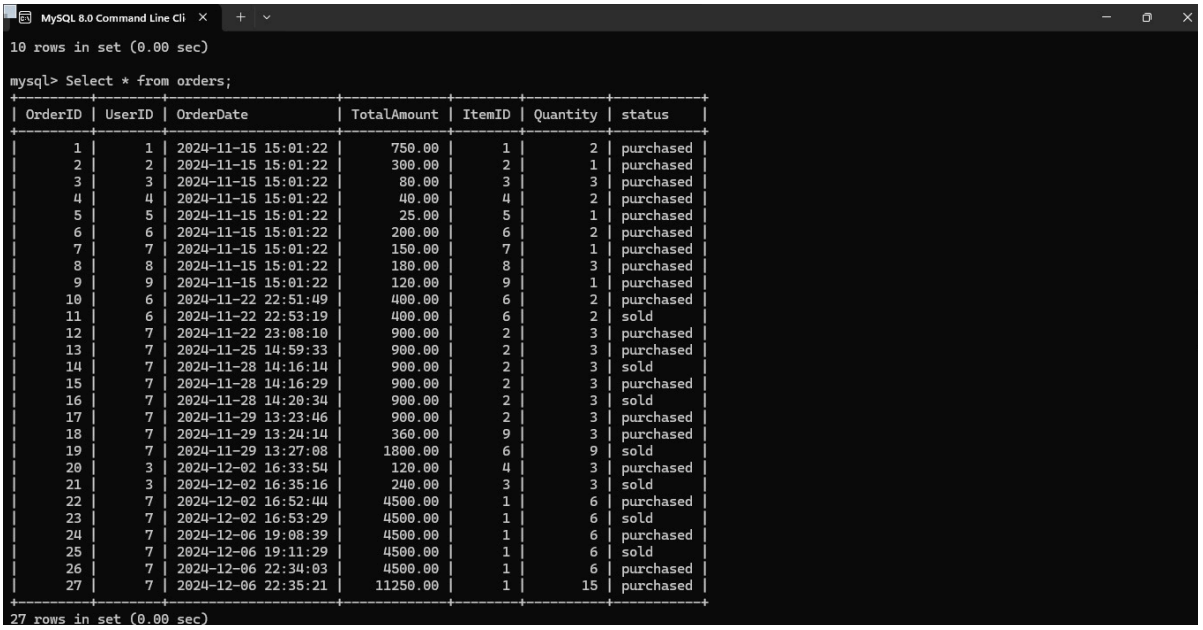
```
mysql> Select * from inventory;
+--------+------------+----------+-----------------------------+-------------+-----------+--------------+---------------------+
| ItemID | ItemName   | Quantity | Description                 | Category    | CostPrice | SellingPrice | CreatedAt           |
+--------+------------+----------+-----------------------------+-------------+-----------+--------------+---------------------+
|      1 | Laptop     |       19 | 15-inch screen laptop       | Electronics |    500.00 |       750.00 | 2024-11-15 15:00:41 |
|      2 | Smartphone |       82 | 64GB smartphone             | Electronics |    200.00 |       300.00 | 2024-11-15 15:00:41 |
|      3 | Headphones |      199 | Noise-cancelling headphones | Accessories |     50.00 |        80.00 | 2024-11-15 15:00:41 |
|      4 | Keyboard   |      147 | Mechanical keyboard         | Accessories |     25.00 |        40.00 | 2024-11-15 15:00:41 |
|      5 | Mouse      |      146 | Wireless mouse              | Accessories |     15.00 |        25.00 | 2024-11-15 15:00:41 |
|      6 | Monitor    |       71 | 24-inch monitor             | Electronics |    150.00 |       200.00 | 2024-11-15 15:00:41 |
|      7 | Desk       |       53 | Office desk                 | Furniture   |    100.00 |       150.00 | 2024-11-15 15:00:41 |
|      8 | Chair      |       64 | Ergonomic chair             | Furniture   |    120.00 |       180.00 | 2024-11-15 15:00:41 |
|      9 | Printer    |       25 | All-in-one printer          | Electronics |     80.00 |       120.00 | 2024-11-15 15:00:41 |
|     10 | Webcam     |       90 | HD webcam                   | Electronics |     30.00 |        50.00 | 2024-11-15 15:00:41 |
+--------+------------+----------+-----------------------------+-------------+-----------+--------------+---------------------+
10 rows in set (0.00 sec)
```

We had 19 items of Laptop, we placed an order for 15 items, the result is 4 items which is below the threshold 5. So we automatically place and order for 30items as it is below threshold.

Purchasing 15 items:



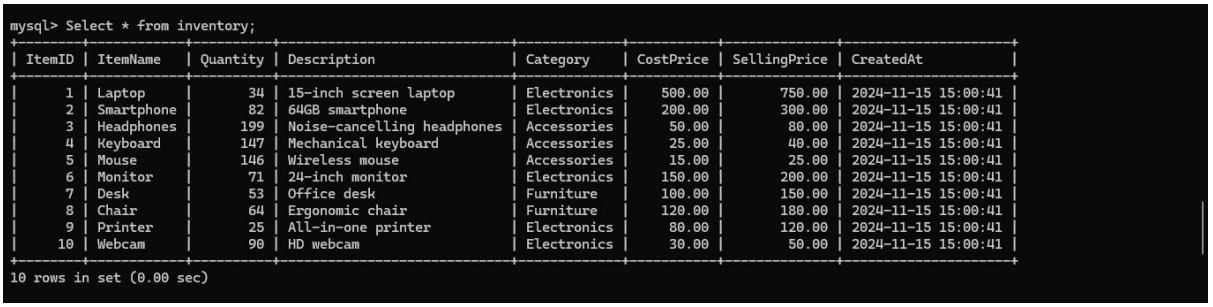
You can see the order for 15 items:



MySQL 8.0 Command Line Cli

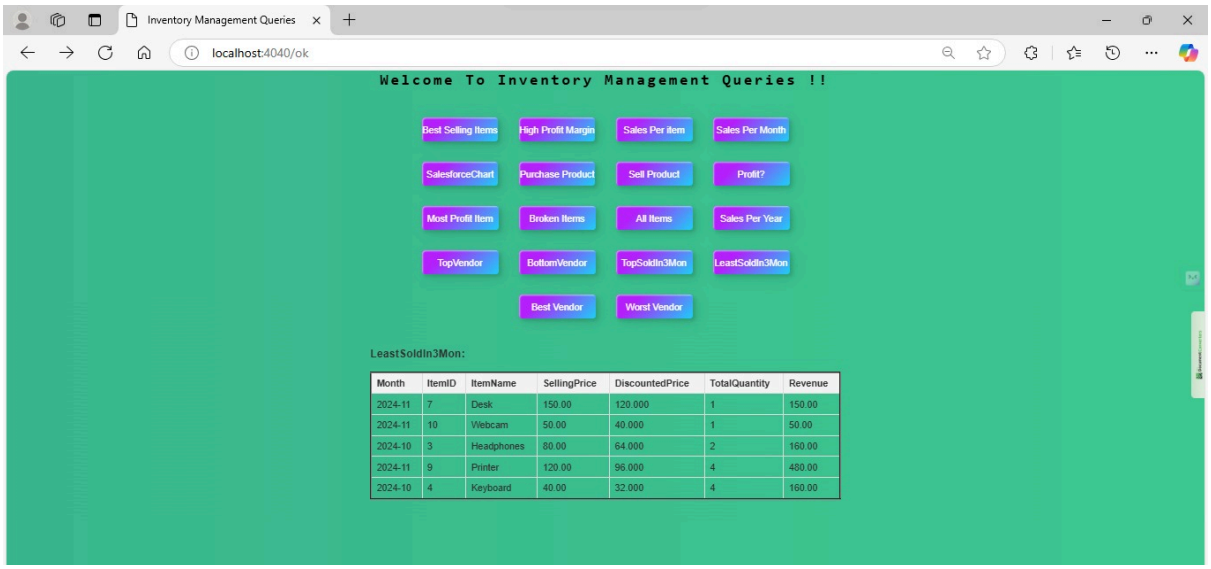
```
10 rows in set (0.00 sec)

mysql> Select * from orders;
+---------+--------+---------------------+-------------+--------+----------+-----------+
| OrderID | UserID | OrderDate           | TotalAmount | ItemID | Quantity | status    |
+---------+--------+---------------------+-------------+--------+----------+-----------+
|       1 |      1 | 2024-11-15 15:01:22 |      750.00 |      1 |        2 | purchased |
|       2 |      2 | 2024-11-15 15:01:22 |      300.00 |      2 |        1 | purchased |
|       3 |      3 | 2024-11-15 15:01:22 |       80.00 |      3 |        3 | purchased |
|       4 |      4 | 2024-11-15 15:01:22 |       40.00 |      4 |        2 | purchased |
|       5 |      5 | 2024-11-15 15:01:22 |       25.00 |      5 |        1 | purchased |
|       6 |      6 | 2024-11-15 15:01:22 |      200.00 |      6 |        2 | purchased |
|       7 |      7 | 2024-11-15 15:01:22 |      150.00 |      7 |        1 | purchased |
|       8 |      8 | 2024-11-15 15:01:22 |      180.00 |      8 |        3 | purchased |
|       9 |      9 | 2024-11-15 15:01:22 |      120.00 |      9 |        1 | purchased |
|      10 |      6 | 2024-11-22 22:51:49 |      400.00 |      6 |        2 | purchased |
|      11 |      6 | 2024-11-22 22:53:19 |      400.00 |      6 |        2 | sold      |
|      12 |      7 | 2024-11-22 23:08:10 |      900.00 |      2 |        3 | purchased |
|      13 |      7 | 2024-11-25 14:59:33 |      900.00 |      2 |        3 | purchased |
|      14 |      7 | 2024-11-28 14:16:14 |      900.00 |      2 |        3 | sold      |
|      15 |      7 | 2024-11-28 14:16:29 |      900.00 |      2 |        3 | purchased |
|      16 |      7 | 2024-11-28 14:20:34 |      900.00 |      2 |        3 | sold      |
|      17 |      7 | 2024-11-29 13:23:46 |      900.00 |      2 |        3 | purchased |
|      18 |      7 | 2024-11-29 13:24:14 |      360.00 |      9 |        3 | purchased |
|      19 |      7 | 2024-11-29 13:27:08 |     1800.00 |      6 |        9 | sold      |
|      20 |      3 | 2024-12-02 16:33:54 |      120.00 |      4 |        3 | purchased |
|      21 |      3 | 2024-12-02 16:35:16 |      240.00 |      3 |        3 | sold      |
|      22 |      7 | 2024-12-02 16:52:44 |     4500.00 |      1 |        6 | purchased |
|      23 |      7 | 2024-12-02 16:53:29 |     4500.00 |      1 |        6 | sold      |
|      24 |      7 | 2024-12-06 19:08:39 |     4500.00 |      1 |        6 | purchased |
|      25 |      7 | 2024-12-06 19:11:29 |     4500.00 |      1 |        6 | sold      |
|      26 |      7 | 2024-12-06 22:34:03 |     4500.00 |      1 |        6 | purchased |
|      27 |      7 | 2024-12-06 22:35:21 |    11250.00 |      1 |       15 | purchased |
+---------+--------+---------------------+-------------+--------+----------+-----------+
27 rows in set (0.00 sec)
```

Now you can see in the inventory it is automatically 34.



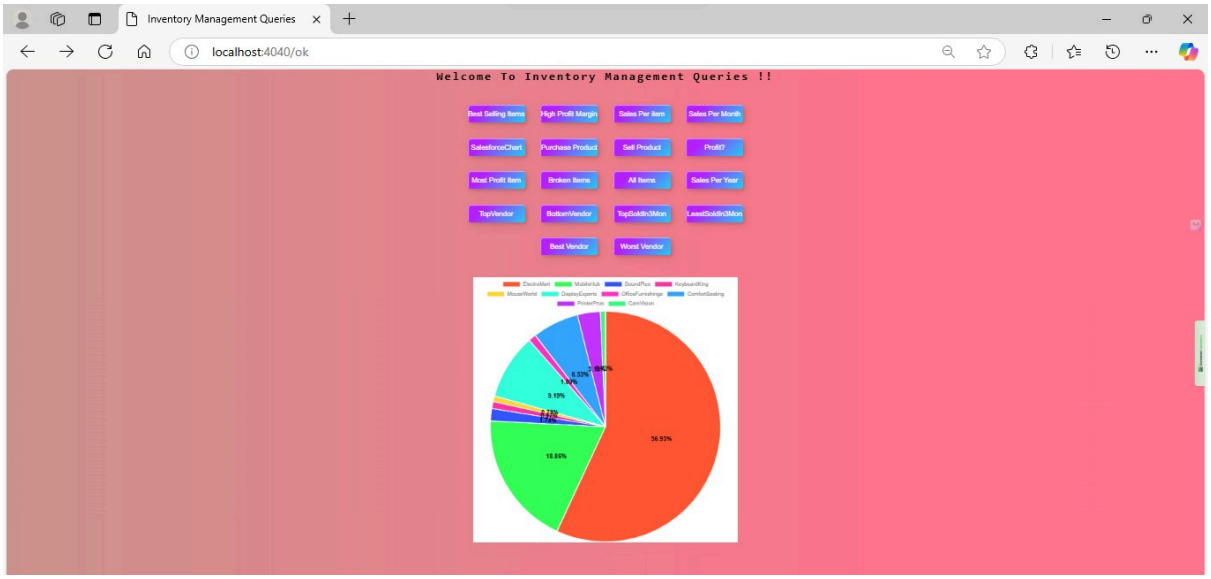
```
mysql> Select * from inventory;
+--------+------------+----------+-----------------------------+-------------+-----------+--------------+---------------------+
| ItemID | ItemName   | Quantity | Description                 | Category    | CostPrice | SellingPrice | CreatedAt           |
+--------+------------+----------+-----------------------------+-------------+-----------+--------------+---------------------+
|      1 | Laptop     |       34 | 15-inch screen laptop       | Electronics |    500.00 |       750.00 | 2024-11-15 15:00:41 |
|      2 | Smartphone |       82 | 64GB smartphone             | Electronics |    200.00 |       300.00 | 2024-11-15 15:00:41 |
|      3 | Headphones |      199 | Noise-cancelling headphones | Accessories |     50.00 |        80.00 | 2024-11-15 15:00:41 |
|      4 | Keyboard   |      147 | Mechanical keyboard         | Accessories |     25.00 |        40.00 | 2024-11-15 15:00:41 |
|      5 | Mouse      |      146 | Wireless mouse              | Accessories |     15.00 |        25.00 | 2024-11-15 15:00:41 |
|      6 | Monitor    |       71 | 24-inch monitor             | Electronics |    150.00 |       200.00 | 2024-11-15 15:00:41 |
|      7 | Desk       |       53 | Office desk                 | Furniture   |    100.00 |       150.00 | 2024-11-15 15:00:41 |
|      8 | Chair      |       64 | Ergonomic chair             | Furniture   |    120.00 |       180.00 | 2024-11-15 15:00:41 |
|      9 | Printer    |       25 | All-in-one printer          | Electronics |     80.00 |       120.00 | 2024-11-15 15:00:41 |
|     10 | Webcam     |       90 | HD webcam                   | Electronics |     30.00 |        50.00 | 2024-11-15 15:00:41 |
+--------+------------+----------+-----------------------------+-------------+-----------+--------------+---------------------+
10 rows in set (0.00 sec)
```

5)Identify items that haven't sold in the past 3 months and come up with a sale. This sale would list such items and sell them at a discount.

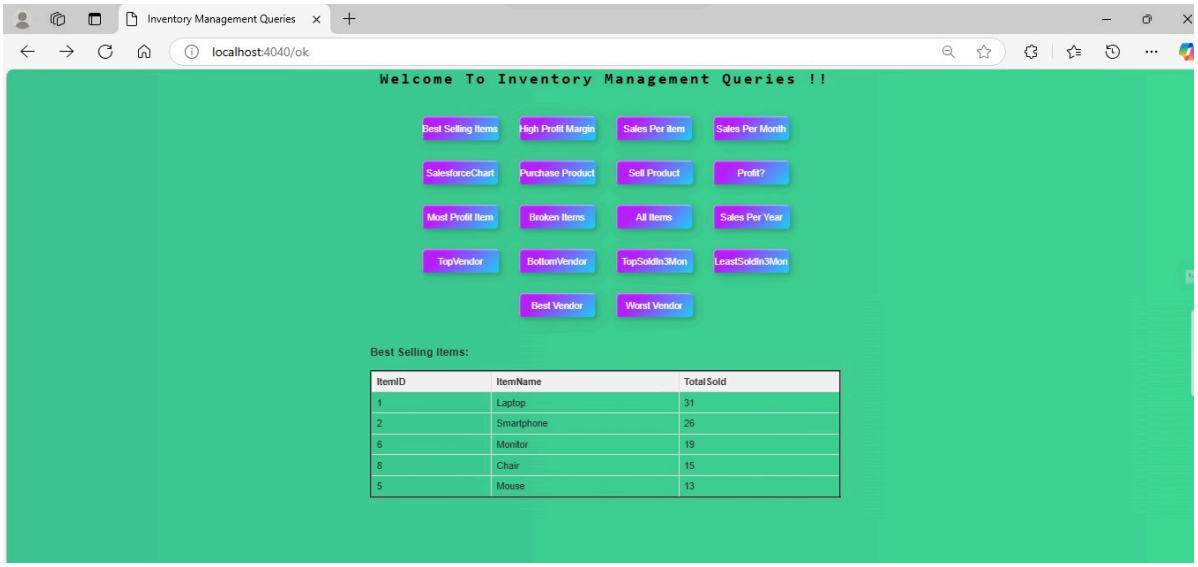


Month	ItemID	ItemName	SellingPrice	DiscountedPrice	TotalQuantity	Revenue
2024-11	7	Desk	150.00	120.000	1	150.00
2024-11	10	Webcam	50.00	40.000	1	50.00
2024-10	3	Headphones	80.00	64.000	2	160.00
2024-11	9	Printer	120.00	96.000	4	480.00
2024-10	4	Keyboard	40.00	32.000	4	160.00

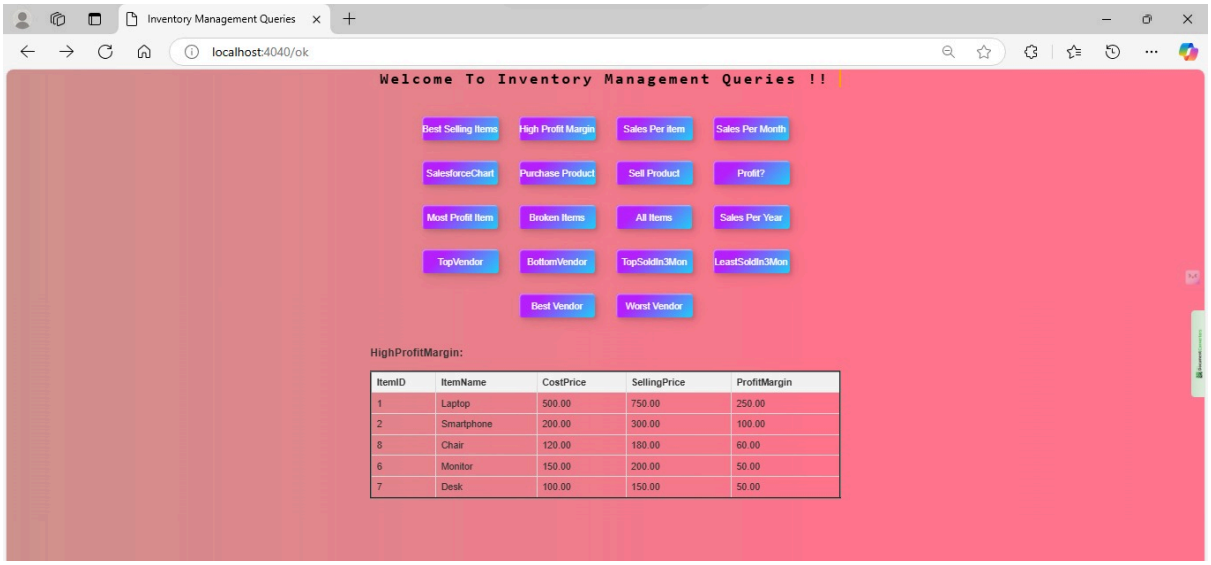
6) A pie chart List Vendor Revenue Data



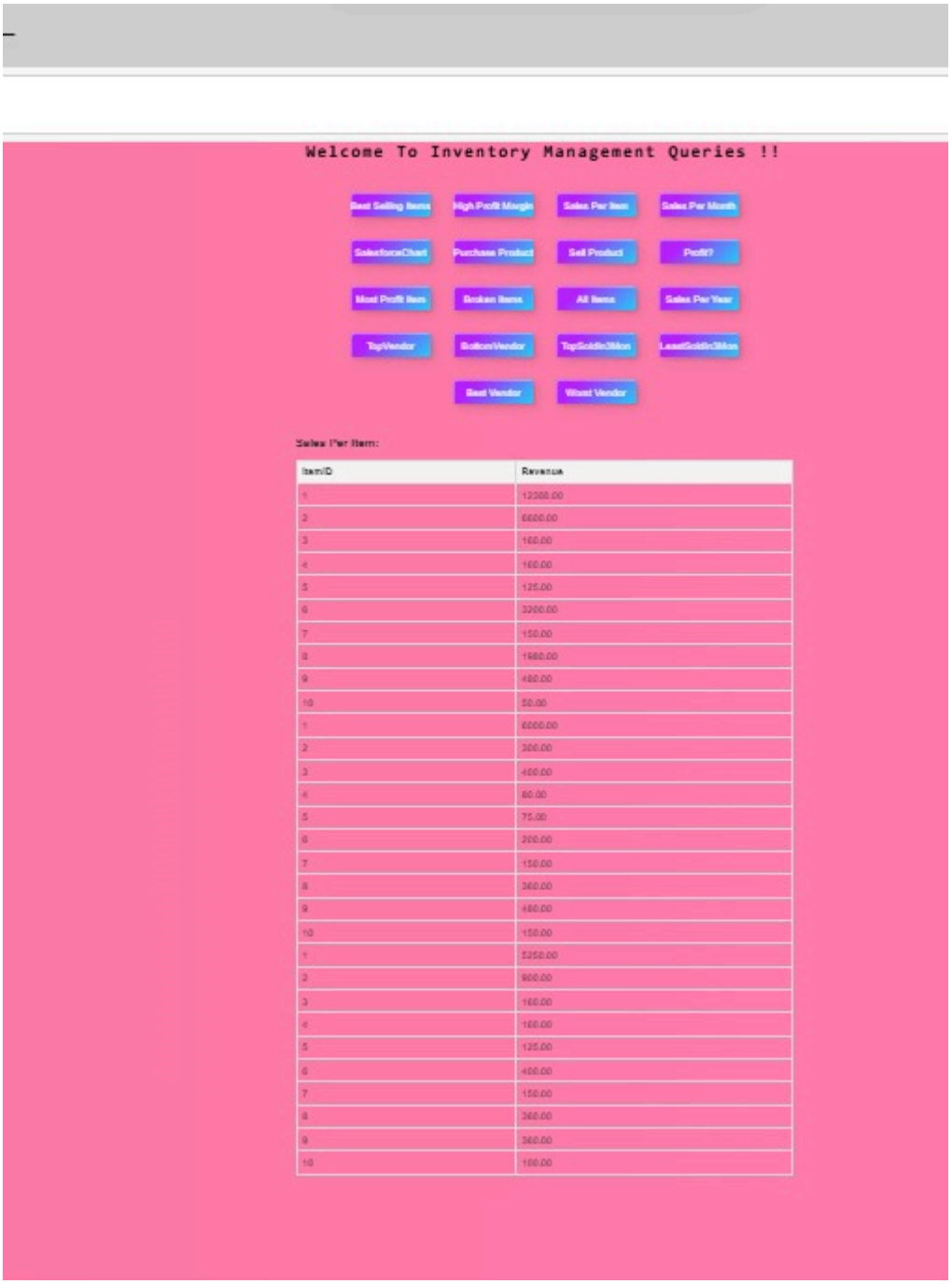
7)List the Top 5 Best-Selling Items



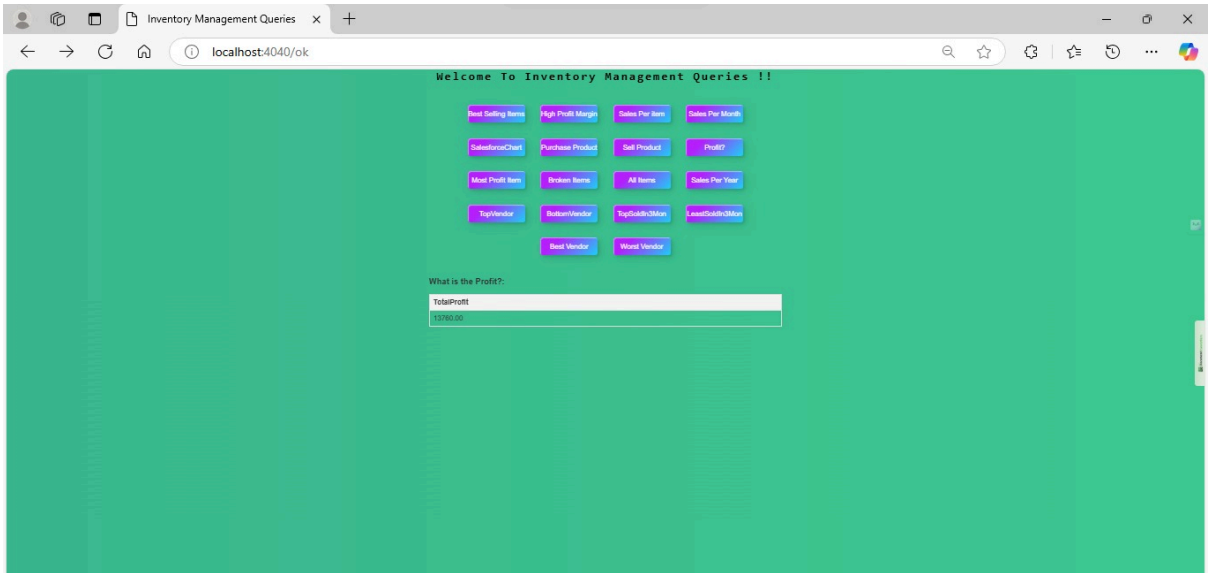
8)List the Top 5 Items with the Highest Profit Margin



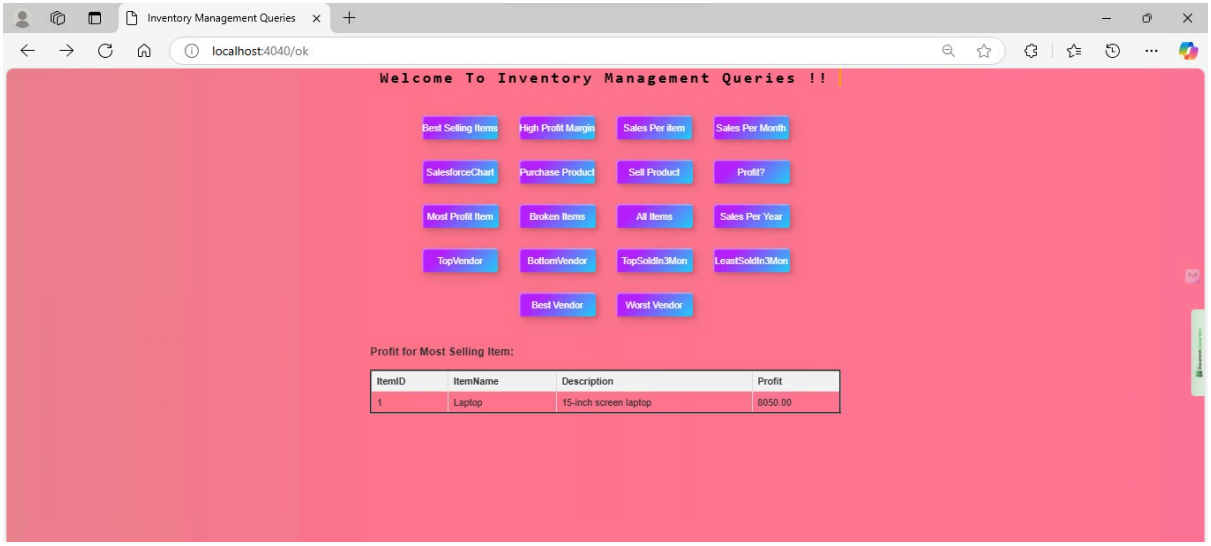
9)List the Total Revenue Generated by Each Item



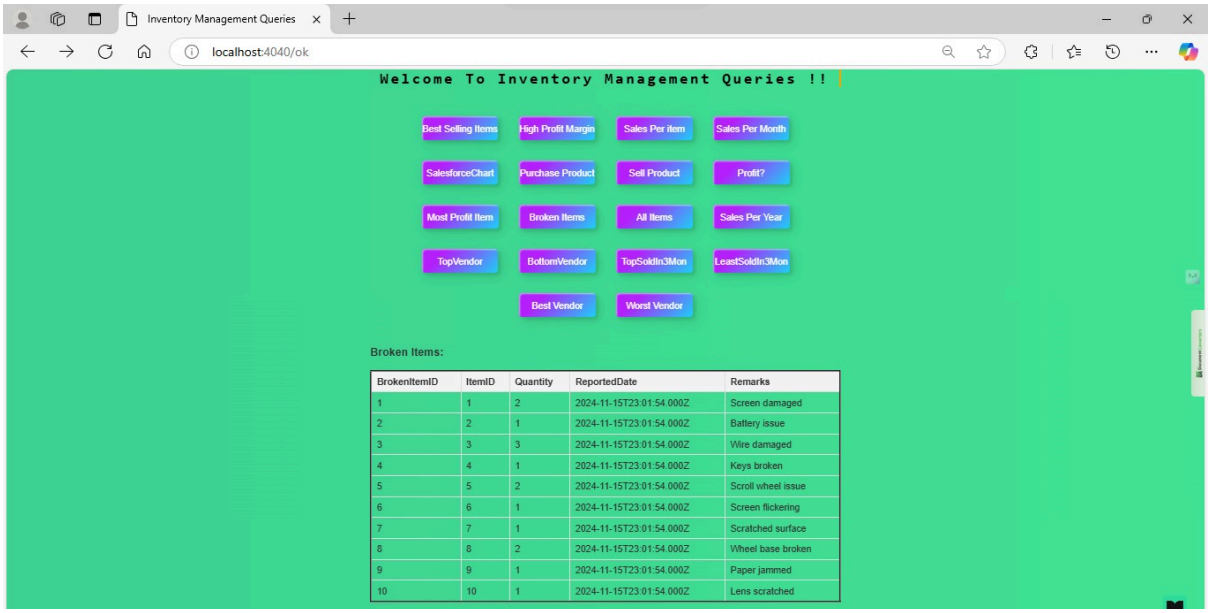
10)Calculate the Total Profit Across All Sales



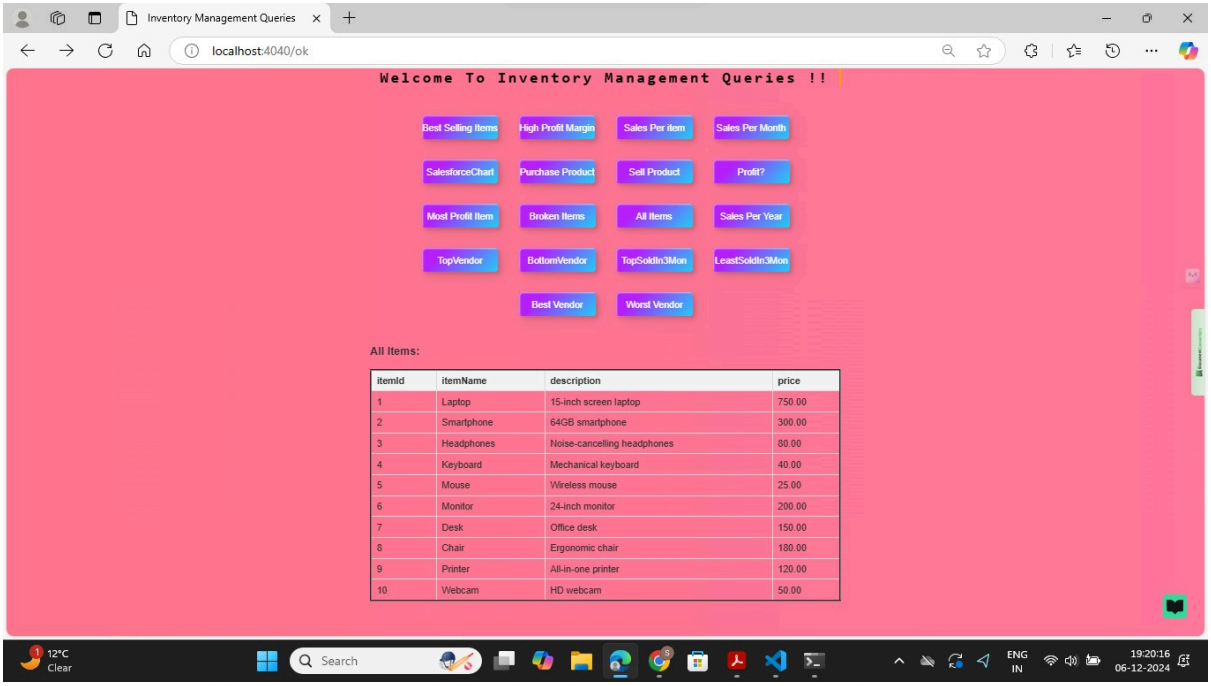
11)Retrieve the Most Profitable Item



12)List All Broken Items and Their Details



13)List All Items from Inventory with Their Details and Prices

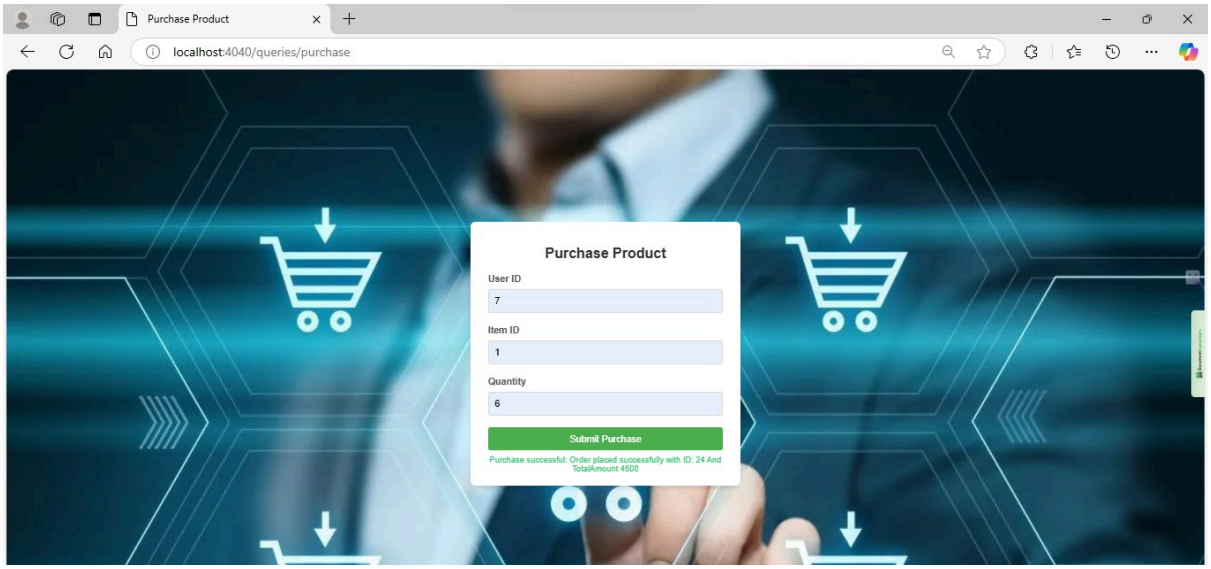


14)Process a Product Purchase by Deducting Stock, Updating Sales and Orders, and Reordering if Stock Falls Below Threshold

Inventory details: Before purchasing

```
mysql> select * from inventory;
+--------+------------+----------+-----------------------------+-------------+-----------+--------------+---------------------+
| ItemID | ItemName   | Quantity | Description                 | Category    | CostPrice | SellingPrice | CreatedAt           |
+--------+------------+----------+-----------------------------+-------------+-----------+--------------+---------------------+
|      1 | Laptop     |       25 | 15-inch screen laptop       | Electronics |    500.00 |       750.00 | 2024-11-15 15:00:41 |
|      2 | Smartphone |       82 | 64GB smartphone             | Electronics |    200.00 |       300.00 | 2024-11-15 15:00:41 |
|      3 | Headphones |      199 | Noise-cancelling headphones | Accessories |     50.00 |        80.00 | 2024-11-15 15:00:41 |
|      4 | Keyboard   |      147 | Mechanical keyboard         | Accessories |     25.00 |        40.00 | 2024-11-15 15:00:41 |
|      5 | Mouse      |      146 | Wireless mouse              | Accessories |     15.00 |        25.00 | 2024-11-15 15:00:41 |
|      6 | Monitor    |       71 | 24-inch monitor             | Electronics |    150.00 |       200.00 | 2024-11-15 15:00:41 |
|      7 | Desk       |       53 | Office desk                 | Furniture   |    100.00 |       150.00 | 2024-11-15 15:00:41 |
|      8 | Chair      |       64 | Ergonomic chair             | Furniture   |    120.00 |       180.00 | 2024-11-15 15:00:41 |
|      9 | Printer    |       25 | All-in-one printer          | Electronics |     80.00 |       120.00 | 2024-11-15 15:00:41 |
|     10 | Webcam     |       90 | HD webcam                   | Electronics |     30.00 |        50.00 | 2024-11-15 15:00:41 |
+--------+------------+----------+-----------------------------+-------------+-----------+--------------+---------------------+
10 rows in set (0.02 sec)
```

Purchasing a product:



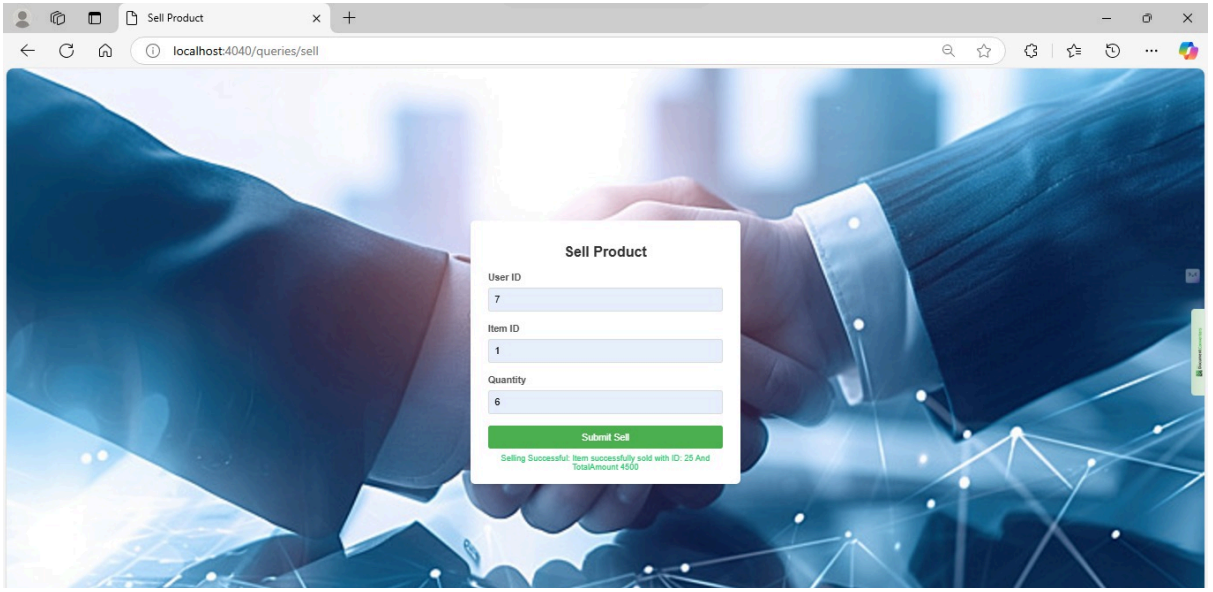
The purchased product details in the orders table:

```
mysql> select * from orders;
+---------+--------+---------------------+-------------+--------+----------+-----------+
| OrderID | UserID | OrderDate           | TotalAmount | ItemID | Quantity | status    |
+---------+--------+---------------------+-------------+--------+----------+-----------+
|       1 |      1 | 2024-11-15 15:01:22 |      750.00 |      1 |        2 | purchased |
|       2 |      2 | 2024-11-15 15:01:22 |      300.00 |      2 |        1 | purchased |
|       3 |      3 | 2024-11-15 15:01:22 |       80.00 |      3 |        3 | purchased |
|       4 |      4 | 2024-11-15 15:01:22 |       40.00 |      4 |        2 | purchased |
|       5 |      5 | 2024-11-15 15:01:22 |       25.00 |      5 |        1 | purchased |
|       6 |      6 | 2024-11-15 15:01:22 |      200.00 |      6 |        2 | purchased |
|       7 |      7 | 2024-11-15 15:01:22 |      150.00 |      7 |        1 | purchased |
|       8 |      8 | 2024-11-15 15:01:22 |      180.00 |      8 |        3 | purchased |
|       9 |      9 | 2024-11-15 15:01:22 |      120.00 |      9 |        1 | purchased |
|      10 |      6 | 2024-11-22 22:51:49 |      400.00 |      6 |        2 | purchased |
|      11 |      6 | 2024-11-22 22:53:19 |      400.00 |      6 |        2 | sold      |
|      12 |      7 | 2024-11-22 23:08:10 |      900.00 |      2 |        3 | purchased |
|      13 |      7 | 2024-11-25 14:59:33 |      900.00 |      2 |        3 | purchased |
|      14 |      7 | 2024-11-28 14:16:14 |      900.00 |      2 |        3 | sold      |
|      15 |      7 | 2024-11-28 14:16:29 |      900.00 |      2 |        3 | purchased |
|      16 |      7 | 2024-11-28 14:20:34 |      900.00 |      2 |        3 | sold      |
|      17 |      7 | 2024-11-29 13:23:46 |      900.00 |      2 |        3 | purchased |
|      18 |      7 | 2024-11-29 13:24:14 |      360.00 |      9 |        3 | purchased |
|      19 |      7 | 2024-11-29 13:27:08 |     1800.00 |      6 |        9 | sold      |
|      20 |      3 | 2024-12-02 16:33:54 |      120.00 |      4 |        3 | purchased |
|      21 |      3 | 2024-12-02 16:35:16 |      240.00 |      3 |        3 | sold      |
|      22 |      7 | 2024-12-02 16:52:44 |     4500.00 |      1 |        6 | purchased |
```

Inventory details:After purchasing

```
mysql> select * from inventory;
+--------+------------+----------+-----------------------------+-------------+-----------+--------------+---------------------+
| ItemID | ItemName   | Quantity | Description                 | Category    | CostPrice | SellingPrice | CreatedAt           |
+--------+------------+----------+-----------------------------+-------------+-----------+--------------+---------------------+
|      1 | Laptop     |       19 | 15-inch screen laptop       | Electronics |    500.00 |       750.00 | 2024-11-15 15:00:41 |
|      2 | Smartphone |       82 | 64GB smartphone             | Electronics |    200.00 |       300.00 | 2024-11-15 15:00:41 |
|      3 | Headphones |      199 | Noise-cancelling headphones | Accessories |     50.00 |        80.00 | 2024-11-15 15:00:41 |
|      4 | Keyboard   |      147 | Mechanical keyboard         | Accessories |     25.00 |        40.00 | 2024-11-15 15:00:41 |
|      5 | Mouse      |      146 | Wireless mouse              | Accessories |     15.00 |        25.00 | 2024-11-15 15:00:41 |
|      6 | Monitor    |       71 | 24-inch monitor             | Electronics |    150.00 |       200.00 | 2024-11-15 15:00:41 |
|      7 | Desk       |       53 | Office desk                 | Furniture   |    100.00 |       150.00 | 2024-11-15 15:00:41 |
|      8 | Chair      |       64 | Ergonomic chair             | Furniture   |    120.00 |       180.00 | 2024-11-15 15:00:41 |
|      9 | Printer    |       25 | All-in-one printer          | Electronics |     80.00 |       120.00 | 2024-11-15 15:00:41 |
|     10 | Webcam     |       90 | HD webcam                   | Electronics |     30.00 |        50.00 | 2024-11-15 15:00:41 |
+--------+------------+----------+-----------------------------+-------------+-----------+--------------+---------------------+
10 rows in set (0.00 sec)
```

15)Processes a Sale by Adding the Sold Quantity Back to Inventory and Recording the Transaction in the Orders Table



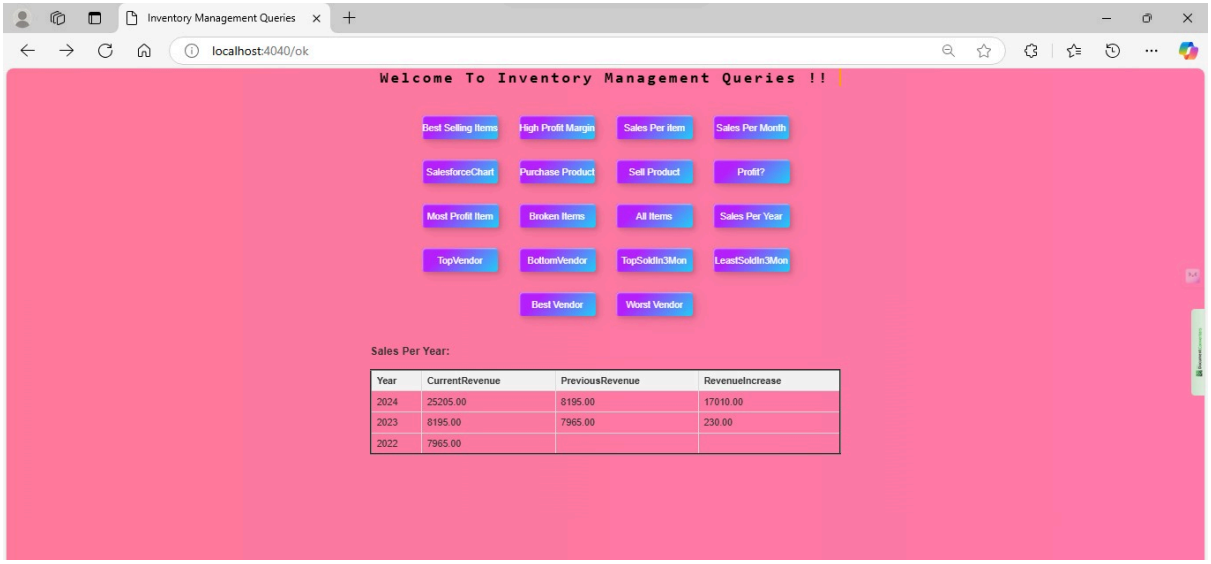
Orders table;

```
mysql> select * from orders;
+---------+--------+---------------------+-------------+--------+----------+-----------+
| OrderID | UserID | OrderDate           | TotalAmount | ItemID | Quantity | status    |
+---------+--------+---------------------+-------------+--------+----------+-----------+
|       1 |      1 | 2024-11-15 15:01:22 |      750.00 |      1 |        2 | purchased |
|       2 |      2 | 2024-11-15 15:01:22 |      300.00 |      2 |        1 | purchased |
|       3 |      3 | 2024-11-15 15:01:22 |       80.00 |      3 |        3 | purchased |
|       4 |      4 | 2024-11-15 15:01:22 |       40.00 |      4 |        2 | purchased |
|       5 |      5 | 2024-11-15 15:01:22 |       25.00 |      5 |        1 | purchased |
|       6 |      6 | 2024-11-15 15:01:22 |      200.00 |      6 |        2 | purchased |
|       7 |      7 | 2024-11-15 15:01:22 |      150.00 |      7 |        1 | purchased |
|       8 |      8 | 2024-11-15 15:01:22 |      180.00 |      8 |        3 | purchased |
|       9 |      9 | 2024-11-15 15:01:22 |      120.00 |      9 |        1 | purchased |
|      10 |      6 | 2024-11-22 22:51:49 |      400.00 |      6 |        2 | purchased |
|      11 |      6 | 2024-11-22 22:53:19 |      400.00 |      6 |        2 | sold      |
|      12 |      7 | 2024-11-22 23:08:10 |      900.00 |      2 |        3 | purchased |
|      13 |      7 | 2024-11-25 14:59:33 |      900.00 |      2 |        3 | purchased |
|      14 |      7 | 2024-11-28 14:16:14 |      900.00 |      2 |        3 | sold      |
|      15 |      7 | 2024-11-28 14:16:29 |      900.00 |      2 |        3 | purchased |
|      16 |      7 | 2024-11-28 14:20:34 |      900.00 |      2 |        3 | sold      |
|      17 |      7 | 2024-11-29 13:23:46 |      900.00 |      2 |        3 | purchased |
|      18 |      7 | 2024-11-29 13:24:14 |      360.00 |      9 |        3 | purchased |
|      19 |      7 | 2024-11-29 13:27:08 |     1800.00 |      6 |        9 | sold      |
|      20 |      3 | 2024-12-02 16:33:54 |      120.00 |      4 |        3 | purchased |
|      21 |      3 | 2024-12-02 16:35:16 |      240.00 |      3 |        3 | sold      |
|      22 |      7 | 2024-12-02 16:52:44 |     4500.00 |      1 |        6 | purchased |
|      23 |      7 | 2024-12-02 16:53:29 |     4500.00 |      1 |        6 | sold      |
```

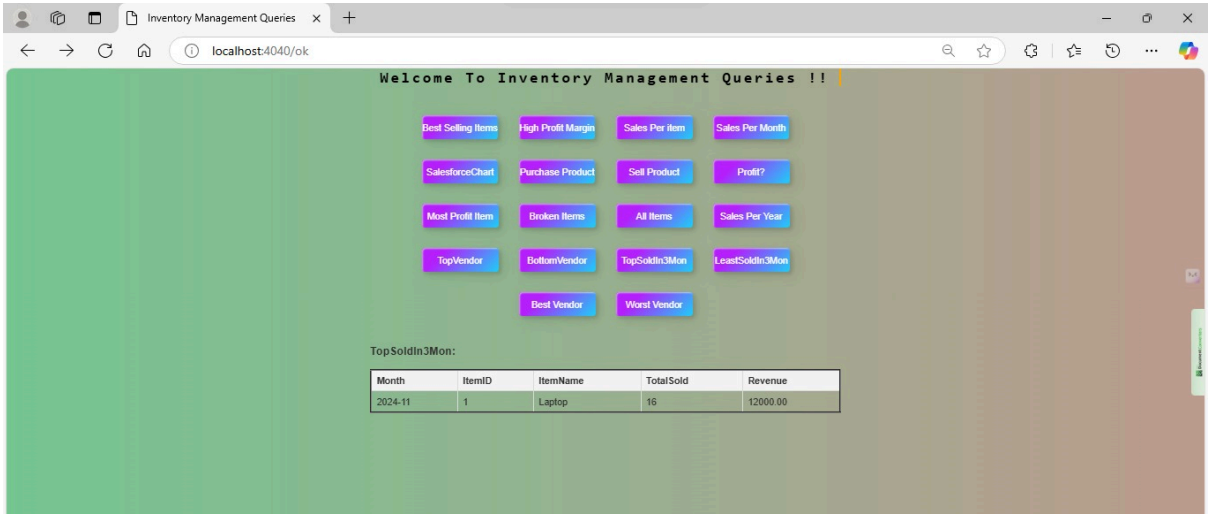
Inventory:

```
mysql> select * from inventory;
+--------+------------+----------+-----------------------------+-------------+-----------+--------------+---------------------+
| ItemID | ItemName   | Quantity | Description                 | Category    | CostPrice | SellingPrice | CreatedAt           |
+--------+------------+----------+-----------------------------+-------------+-----------+--------------+---------------------+
|      1 | Laptop     |       25 | 15-inch screen laptop       | Electronics |    500.00 |       750.00 | 2024-11-15 15:00:41 |
|      2 | Smartphone |       82 | 64GB smartphone             | Electronics |    200.00 |       300.00 | 2024-11-15 15:00:41 |
|      3 | Headphones |      199 | Noise-cancelling headphones | Accessories |     50.00 |        80.00 | 2024-11-15 15:00:41 |
|      4 | Keyboard   |      147 | Mechanical keyboard         | Accessories |     25.00 |        40.00 | 2024-11-15 15:00:41 |
|      5 | Mouse      |      146 | Wireless mouse              | Accessories |     15.00 |        25.00 | 2024-11-15 15:00:41 |
|      6 | Monitor    |       71 | 24-inch monitor             | Electronics |    150.00 |       200.00 | 2024-11-15 15:00:41 |
|      7 | Desk       |       53 | Office desk                 | Furniture   |    100.00 |       150.00 | 2024-11-15 15:00:41 |
|      8 | Chair      |       64 | Ergonomic chair             | Furniture   |    120.00 |       180.00 | 2024-11-15 15:00:41 |
|      9 | Printer    |       25 | All-in-one printer          | Electronics |     80.00 |       120.00 | 2024-11-15 15:00:41 |
|     10 | Webcam     |       90 | HD webcam                   | Electronics |     30.00 |        50.00 | 2024-11-15 15:00:41 |
+--------+------------+----------+-----------------------------+-------------+-----------+--------------+---------------------+
10 rows in set (0.00 sec)

mysql>
```

16)Lists Annual Sales Revenue and the Revenue Increase Compared to the Previous Year



17)Fetch the Most Sold Item in the Last 3 Months with Its Quantity and Revenue



18)Listing Feature.

We created two tables:

1)Product Features Table

```
CREATE TABLE ProductFeatures (
    FeatureID INT AUTO_INCREMENT PRIMARY KEY,
    ItemID INT NOT NULL,
    FeatureKey VARCHAR(255) NOT NULL,
    FeatureValue VARCHAR(255) NOT NULL,
    FOREIGN KEY (ItemID) REFERENCES Inventory(ItemID)
);
```

2)**Product Filters Table**

```
CREATE TABLE ProductFilters (
    FilterID INT AUTO_INCREMENT PRIMARY KEY,
    ItemID INT NOT NULL,
    FilterKey VARCHAR(255) NOT NULL,
    FilterValue VARCHAR(255) NOT NULL,
    FOREIGN KEY (ItemID) REFERENCES Inventory(ItemID)
);
```

The tables are populated with sample data:
ProductFeatures:

```
INSERT INTO ProductFeatures (ItemID, FeatureKey, FeatureValue)
VALUES
-- Laptop
(1, 'Brand', 'Apple'),
(1, 'Model Name', 'MacBook Pro'),
```

(1, 'Screen Size', '14.2 Inches'),

(1, 'Color', 'Silver'),

(1, 'Hard Disk Size', '512 GB'),

(1, 'CPU Model', 'Apple M3'),

(1, 'Ram Memory Installed Size', '8 GB'),

-- Smartphone

(2, 'Brand', 'Samsung'),

(2, 'Model Name', 'Galaxy S23'),

(2, 'Screen Size', '6.7 Inches'),

(2, 'Color', 'Phantom Black'),

(2, 'Storage', '128 GB'),

(2, 'Battery Capacity', '4500 mAh'),

(2, 'Camera', '108 MP'),

-- Monitor

(6, 'Brand', 'Dell'),

(6, 'Screen Size', '24 Inches'),

(6, 'Resolution', '1920x1080'),

(6, 'Refresh Rate', '144Hz'),

(6, 'Panel Type', 'IPS'),

(6, 'Color', 'Black'),

-- Printer

(9, 'Brand', 'HP'),

(9, 'Printer Type', 'All-in-One'),

(9, 'Color Capability', 'Color'),

(9, 'Connectivity', 'Wi-Fi'),

(9, 'Paper Size', 'A4'),

(9, 'Print Speed', '20 ppm'),

(9, 'Color', 'White');

```
mysql> select * from productfeatures;
+-----------+--------+--------------------------+---------------+
| FeatureID | ItemID | FeatureKey               | FeatureValue  |
+-----------+--------+--------------------------+---------------+
|         1 |      1 | Brand                    | Apple         |
|         2 |      1 | Model Name               | MacBook Pro   |
|         3 |      1 | Screen Size              | 14.2 Inches   |
|         4 |      1 | Color                    | Silver        |
|         5 |      1 | Hard Disk Size           | 512 GB        |
|         6 |      1 | CPU Model                | Apple M3      |
|         7 |      1 | Ram Memory Installed Size | 8 GB         |
|         8 |      2 | Brand                    | Samsung       |
|         9 |      2 | Model Name               | Galaxy S23    |
|        10 |      2 | Screen Size              | 6.7 Inches    |
|        11 |      2 | Color                    | Phantom Black |
|        12 |      2 | Storage                  | 128 GB        |
|        13 |      2 | Battery Capacity         | 4500 mAh      |
|        14 |      2 | Camera                   | 108 MP        |
|        15 |      6 | Brand                    | Dell          |
|        16 |      6 | Screen Size              | 24 Inches     |
|        17 |      6 | Resolution               | 1920x1080     |
|        18 |      6 | Refresh Rate             | 144Hz         |
|        19 |      6 | Panel Type               | IPS           |
|        20 |      6 | Color                    | Black         |
|        21 |      9 | Brand                    | HP            |
|        22 |      9 | Printer Type             | All-in-One    |
|        23 |      9 | Color Capability         | Color         |
|        24 |      9 | Connectivity             | Wi-Fi         |
|        25 |      9 | Paper Size               | A4            |
|        26 |      9 | Print Speed              | 20 ppm        |
|        27 |      9 | Color                    | White         |
+-----------+--------+--------------------------+---------------+
27 rows in set (0.01 sec)
```

ProductFilters:

INSERT INTO ProductFilters (ItemID, FilterKey, FilterValue)

VALUES

-- Laptop

(1, 'RAM', '8GB'),

(1, 'RAM', '16GB'),

(1, 'Storage', '512GB'),

(1, 'Storage', '1TB'),

-- Smartphone

(2, 'Storage', '64GB'),

(2, 'Storage', '128GB'),

(2, 'Color', 'Phantom Black'),

(2, 'Color', 'Cream'),

-- Monitor

(6, 'Screen Size', '24 Inches'),

(6, 'Screen Size', '32 Inches'),

(6, 'Refresh Rate', '60Hz'),

(6, 'Refresh Rate', '144Hz'),

-- Printer

(9, 'Color Capability', 'Color'),

(9, 'Color Capability', 'Black & White'),

(9, 'Connectivity', 'Wi-Fi'),(9, 'Connectivity', 'Ethernet');

```
mysql> select * from productfilters;
+----------+--------+------------------+---------------+
| FilterID | ItemID | FilterKey        | FilterValue   |
+----------+--------+------------------+---------------+
|        1 |      1 | RAM              | 8GB           |
|        2 |      1 | RAM              | 16GB          |
|        3 |      1 | Storage          | 512GB         |
|        4 |      1 | Storage          | 1TB           |
|        5 |      2 | Storage          | 64GB          |
|        6 |      2 | Storage          | 128GB         |
|        7 |      2 | Color            | Phantom Black |
|        8 |      2 | Color            | Cream         |
|        9 |      6 | Screen Size      | 24 Inches     |
|       10 |      6 | Screen Size      | 32 Inches     |
|       11 |      6 | Refresh Rate     | 60Hz          |
|       12 |      6 | Refresh Rate     | 144Hz         |
|       13 |      9 | Color Capability | Color         |
|       14 |      9 | Color Capability | Black & White |
|       15 |      9 | Connectivity     | Wi-Fi         |
|       16 |      9 | Connectivity     | Ethernet      |
+----------+--------+------------------+---------------+
16 rows in set (0.02 sec)
```

Steps to execute for amazon listing feature:

Clone the github repository:
https://github.com/abharathkumarr/inventorymanagement/tree/main

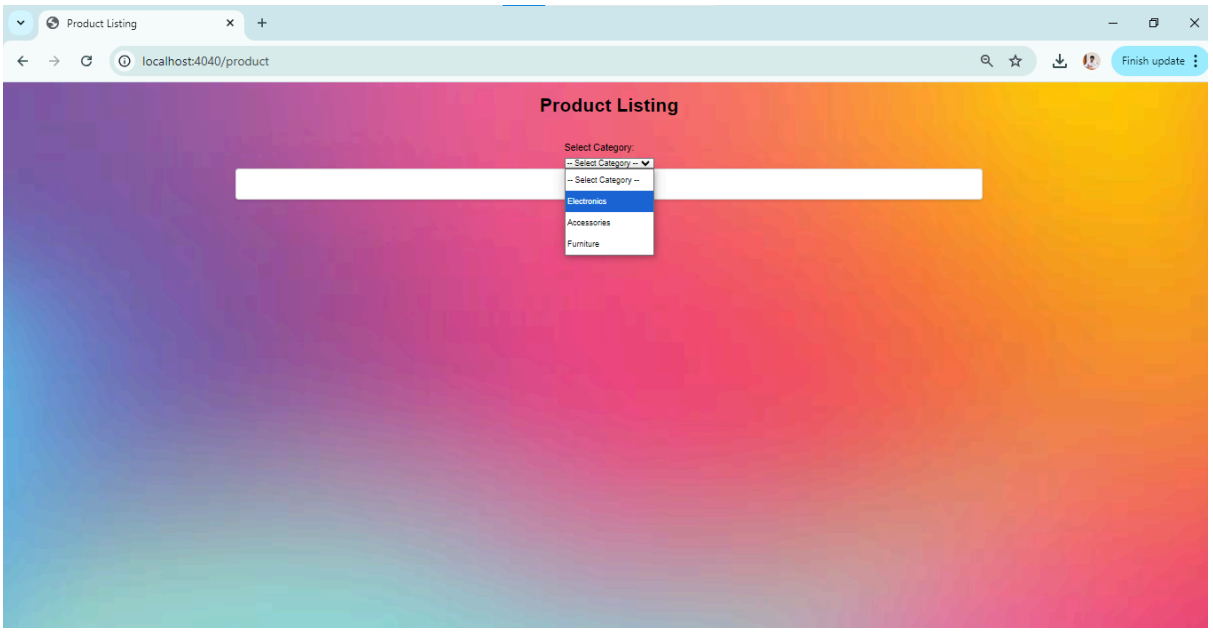
create tables and insert above data into your database.

Go to Vscode and from api folder: run below command: node app.js

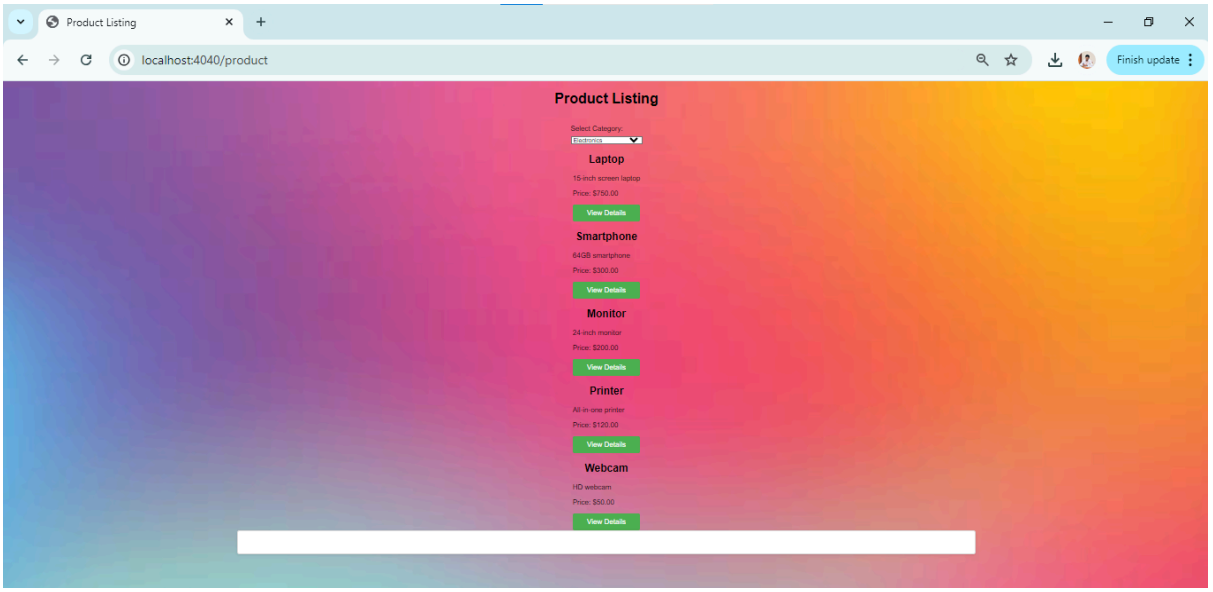
Ex:C:\Users\Checkout\Desktop\cs157aproject\invm\api> node app.js

Go to:http://localhost:4040/product

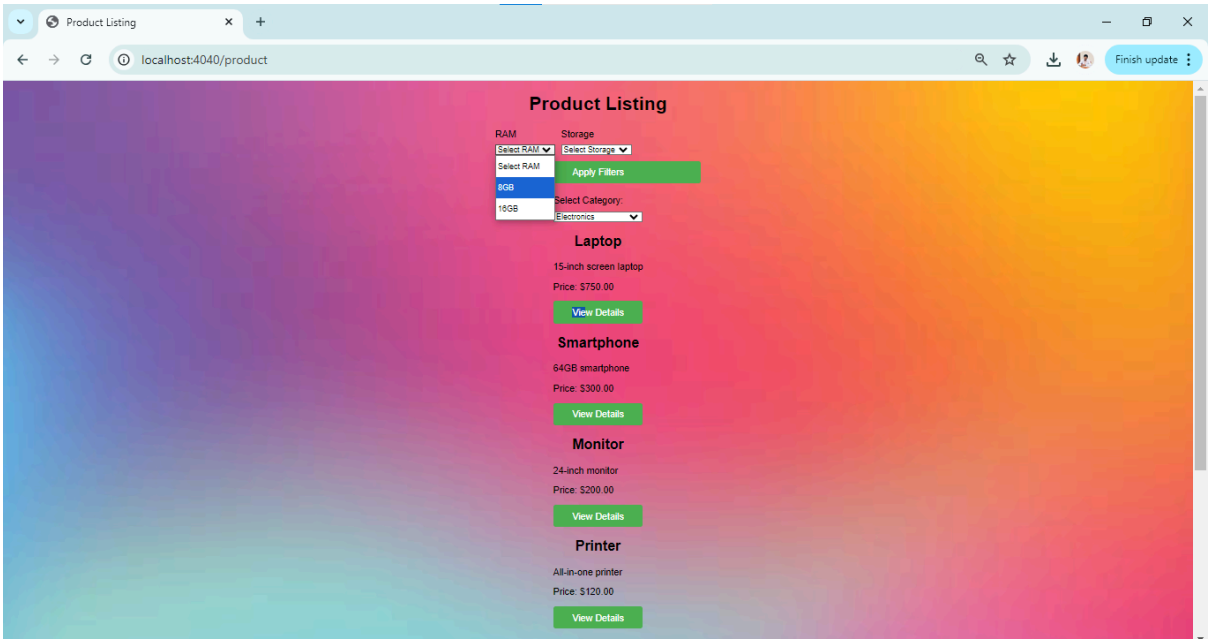
>>Select the category



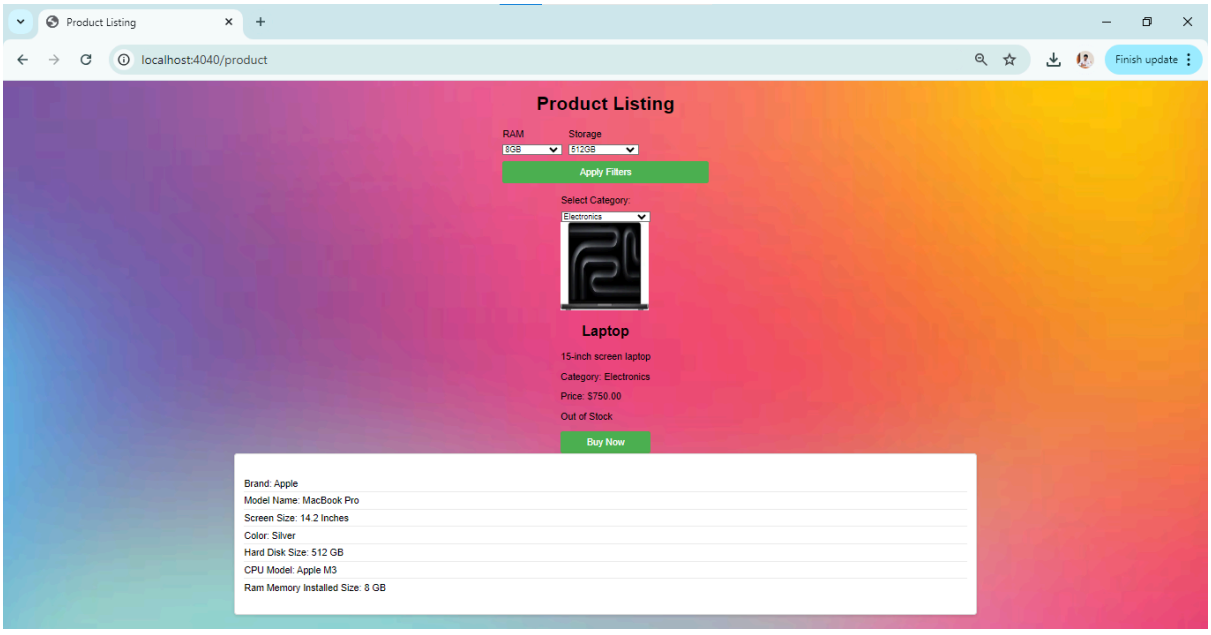
>>Under Laptop click view details



>>Apply filters: I selected Ram:8gb and storage:512gb and click apply filters.



>>Result page:



Code Repository

Github Link:

https://github.com/sanbancan/CS157A-dbms-inventory-management.git

Queries Link:

https://github.com/sanbancan/CS157A-dbms-project--inventory-management/blob/main/invm/invm/api/routers/queries.js

Code Repository for Listing Feature:

Github Link:

https://github.com/abharathkumarr/inventorymanagement/tree/main

Project Details

Tech stack:

Database – mysql

Backend – node js , express js

Frontend – html, css, js

Steps to Execute:

> Clone the github repository and download the code.
> create tables and insert the tables data into your database.

>Install npm modules with command: npm install

>Go to Vscode and from api folder: run below command: **node app.js**

Ex:C:\Users\Checkout\Desktop\cs157aproject\invm\api> node app.js

>Go to:http://localhost:4040

CODE PART(BACKEND)

Connection to db (mysql) in js – conn.js

The **mysql2** library is required to enable communication with the MySQL database. This library is faster and supports promises, making it suitable for asynchronous database operations.

Purpose of conn: This function establishes the connection to the MySQL database.

Database Configuration:

- host: Specifies the database server address (localhost for local development).
- user: The username to connect to the database (root in this case).
- password: The password for the user (root123 in this case).
- database: The name of the database to use (inventory_management in this case).

Promise-Based Connection:

- The .promise() method is called on the connection. This transforms the connection into a promise-based object, allowing you to use async/await for cleaner asynchronous code.

Error Handling:

- If an error occurs while connecting to the database, it is caught in the catch block, and an error message is logged.

Backend code (app.js)

This code defines a Node.js server using the **Express.js** framework, serving as the backend for a web application. Below is an explanation of the components and each route:

General Setup

1. **Importing Dependencies**:
    - express: The web framework for defining routes and handling HTTP requests.
    - path: Provides utilities for working with file and directory paths.
    - connection: Database connection file for executing MySQL queries.
    - queryRoutes: A separate router module for handling /queries routes.
2. **Basic Configuration**:
    - Creates an Express application (app).
    - Defines the server to listen on port 4040.
    - Configures middleware to parse JSON requests (express.json) and serve static files (express.static).

Route Explanations

1. **app.use('/queries', queryRoutes')**
    - Forwards all routes starting with /queries to the queryRoutes router.
2. **app.get('/')**
    - Redirects users from the root URL / to /home.
3. **app.get('/home')**

- Serves the main homepage (index.html) to the user.

4. **app.get('/signup')**
    - Serves the signup page (signup.html).
5. **app.get('/ok')**
    - Serves the user's home or dashboard page (home.html) after a successful login.
6. **app.get('/queries/purchase')**
    - Serves the purchase page (purchase.html).
7. **app.get('/queries/sell')**
    - Serves the sell page (sell.html).
8. **app.post('/userSignup')**
    - Accepts user registration data, validates it, and inserts the data into the users table in the database.
9. **app.post('/login')**
    - Checks if the provided email exists in the Users table and verifies the password. Returns success or failure accordingly.
10. **app.use(express.static(path.join(__dirname, '..', 'public')))**
    - Serves static assets like CSS, JavaScript, and images from the public directory.
11. **app.listen(port)**
    - Starts the server and listens on port 4040.

QUERY ROUTES (routers.js)

Route Explanations

1. **router.get('/best-selling')**
    - Fetches the top 5 best-selling items based on total quantities sold.
    - Joins the Sales and Inventory tables to get item names.
2. **router.get('/high-profit-margin')**
    - Retrieves the top 5 items with the highest profit margins (SellingPrice - CostPrice).
    - Orders results by the profit margin in descending order.
3. **router.get('/sales-per-month')**
    - Aggregates monthly sales revenue.
    - Groups sales data by month and orders results by the latest month.
4. **router.get('/sales-per-item')**
    - Fetches total revenue for each item from the Sales table.
5. **router.get('/most-profit-item')**
    - Calculates the most profitable item by joining Sales, Vendors, and Inventory tables.
    - Computes profit based on total revenue minus the cost (Quantity × Vendor price).
6. **router.get('/profit')**
    - Computes total profit across all sales.
    - Uses the Sales and Vendors tables to calculate profit margins.
7. **router.get('/salesforce')**
    - Retrieves total revenue for each vendor.
    - Groups data by vendor and uses a LEFT JOIN between Vendors and Sales tables.

8. **router.get('/broken-items')**
    - Fetches a list of broken items from the brokenItems table.
9. **router.get('/all-items')**
    - Returns all items in the inventory with basic details (ID, name, description, price).
10. **router.post('/purchase-product')**
    - Processes a product purchase by a user.
    - Updates inventory, handles low stock replenishment, and records the purchase in the Orders table.
    - Updates total sales revenue in the Sales table.
11. **router.post('/sell-product')**
    - Processes a product sale by a user.
    - Updates inventory, calculates total sales amount, and records the sale in the Orders table.
12. **router.get('/most-items-3')**
    - Fetches items sold the most in the last three months.
    - Joins Sales and Inventory tables and aggregates quantities.
13. **router.get('/least-items-3')**
    - Retrieves items not sold at all in the last three months.
    - Uses a subquery to exclude items present in the Sales table within the timeframe.

CODE PART(FRONTEND)

Index.html

This is an **HTML page** for a login system in an **Inventory Management System**. Here's a breakdown of its structure and functionality:

---

1. Head Section

- **Meta Information:**
  - Declares the document's character set as UTF-8-(Unicode Transformation Format-8-bit) is a character encoding standard that represents characters in the Unicode standard using one to four bytes.
  - Sets viewport for responsive design on mobile devices.
- **Stylesheets:**
  - Links to a local CSS file (style.css) for custom styling.
  - Links to **Bootstrap CSS** for pre-designed responsive components.
  - Links to **FontAwesome CSS** for icons.

---

2. Navbar

- A fixed navigation bar at the top using Bootstrap:
  - Displays the system name, "Inventory Management System."
  - Keeps the navbar fixed on top while scrolling (navbar-fixed-top).

---

3. Header Title

- Contains a welcoming title ("WELCOME TO INVENTORY MANAGEMENT SYSTEM").

---

4. Login Form

- A styled form to input login credentials:
  - **Email Field:** Accepts a valid email address.
  - **Password Field:** Accepts a password.
  - **Submit Button:** Submits the form to the /login route via POST method.
- **Error Display:** An empty div to show login errors dynamically.
- **Sign Up Link:** A link directing users without an account to the /signup page.

---

5. Footer Section

- Displays contact information:
  - Phone number with a phone icon from FontAwesome.
  - Email address with an envelope icon.

---

6. JavaScript Functionality

- **Login Form Submission:**
  - Uses JavaScript to handle the form submission.
  - Prevents the default behavior (e.preventDefault()).
  - Sends the form data (email and password) as a JSON payload to the /login route.
  - **Error Handling:**
    - If the login fails, displays the error message returned by the server in the error-message div.
  - **Redirect on Success:**
    - On successful login, redirects the user to the /ok page.

---

7. External JavaScript

- **jQuery and Bootstrap JS:**
    - Supports interactive features like modals, alerts, and more from Bootstrap.

Features and Technologies Used

1. **Responsive Design:** Bootstrap ensures compatibility across various devices.
2. **Dynamic Interactions:** JavaScript provides a dynamic login experience with error handling and redirection.
3. **FontAwesome Icons:** Adds a professional look with icons for phone and email.
4. **Backend Integration:** Form submission to a backend server for processing login credentials.

home.html

Explanation of the HTML and JavaScript Code

The code creates a web page with a visually appealing **Inventory Management System**. Users can click buttons to execute specific queries or view graphical results like bar and pie charts.

HTML Structure

1. <!DOCTYPE html>

Specifies the HTML5 document type.

2. <head>

Contains metadata and links for styling and functionality.

- **Meta tags** ensure proper rendering and scaling on devices of different sizes.
- **Title**: Sets the title displayed in the browser tab.
- **CSS Styling**: Provides visual styles (explained below).

3. <body>

The main content of the webpage is within the <body> tag.

- **Container Div**:
    - Encapsulates the entire content, ensuring a clean layout.
- **Typewriter Header**:
    - Animated header text (Welcome To Inventory Management Queries !!) mimicking a typewriter effect.
- **Button Container**:
    - Holds all the action buttons, each corresponding to a query or functionality.

- **Buttons**:
  - Buttons have unique IDs to distinguish their functionalities.
  - Buttons use a custom CSS class (custom-btn btn-12) for styling.

- **Result Section**:
  - A div element with ID queryResult is used to display query results (e.g., tables, charts, or messages).

- **Canvas for Charts**:
  - A <canvas> element is dynamically created to render **Chart.js** visualizations like pie charts and bar charts.

JavaScript Code

Libraries Used

1. **Chart.js**: A library to create responsive charts.
2. **Chart.js Plugin Datalabels**: Enhances charts by adding labels directly to the data points.

Key JavaScript Functionalities

1. **fetchData() Function**
   - Sends a fetch request to the server to retrieve data for a query.
   - Clears any previously displayed results (e.g., tables or charts).
   - Calls the appropriate rendering function based on the query (e.g., displayBarChart, displayPieChart, or displayResultInTable).

2. **displayResultInTable()**
    - Displays data as a table in the queryResult container.
    - Dynamically generates a table with headers and rows from the fetched data.

3. **displayBarChart() and displayBarChart2()**
    - Render bar charts for **"Sales Per Item"** and **"Sales Per Month"** queries using Chart.js.
    - Uses labels (e.g., ItemID or Month) and corresponding values (Revenue).

4. **displayPieChart()**
    - Renders a pie chart for Salesforce revenue distribution.
    - Displays vendor names and their corresponding revenue.
    - Includes percentages directly on the pie chart using the datalabels plugin.
5. **Event Listeners**
    - Each button has an event listener for the click event.
    - On clicking, the relevant data-fetching function is triggered.
    - Example: Clicking "Best Selling Items" triggers fetchData('/queries/best-selling', 'Best Selling Items');.

CSS Details

The provided CSS creates a visually appealing, modern design:

Body Styling

- Gradient background with an **animated transition** between colors (gradientAnimation).

- Smooth animations using @keyframes.

Header (Typewriter Effect)

- Animated typing effect with a blinking caret.
- Controlled by typing and blink-caret keyframes.

Button Styling

- Custom button design (.btn-12) with hover effects.
- Buttons appear 3D with a rotation animation on hover.

Result Tables

- Tables are styled with alternating colors, padding, and borders for readability.

Chart Container

- Charts are displayed dynamically in a clean layout with a white background for contrast.

Features and Functionalities

1. **Dynamic Query Execution**:
    - Each button triggers a specific server route to fetch query results.
    - Results are displayed in different formats (tables or charts).
2. **Graphical Representation**:
    - Supports bar charts for quantitative data and pie charts for categorical distributions.
3. **Responsive Design**:
    - Layout adjusts across devices using CSS flexbox.
4. **Interactive Design**:

- Buttons have hover effects, and charts dynamically update based on data.

Purchase.html

This HTML page allows users to input details for purchasing a product, and it communicates with a backend endpoint (/queries/purchase-product) to process the purchase. Here's a breakdown of the key elements in the code:

Features:

1. **Responsive Design**:
   - The form is styled to be user-friendly and visually appealing, with a responsive layout and clean alignment.
   - It uses a centered container with box-shadow and rounded corners for a modern design.

2. **Background Styling**:
   - A background image adds visual context, with background-size: cover; ensuring it fills the screen proportionally.

3. **Form Validation**:
   - Each field (userID, itemID, quantity) is marked as required to ensure users cannot submit the form with empty fields.

4. **Frontend Logic with Fetch API**:
    - On form submission, JavaScript prevents the default behavior and instead sends a POST request to the backend using the Fetch API.
    - Input values are collected and sent as a JSON payload.

5. **User Feedback**:
    - The success or error message from the backend is displayed below the form in a user-friendly format.
    - Clear error/success indicators (error and success classes) differentiate between outcomes.

6. **Accessible and Secure Design**:
    - Labels are associated with inputs using the for attribute for better accessibility.
    - The backend endpoint is accessed over an asynchronous HTTP POST request, ensuring scalability.

sell.html

This HTML document provides a user interface for selling a product through a form. It features a visually appealing design, client-side scripting for interaction, and an integration with a backend endpoint (/queries/sell-product) to handle sell requests. Below is a breakdown of its functionality and design:

Page Features:

1. **Title and Layout**:
    - The page is titled "Sell Product."

- It uses a full-page background image (background: url(...)) that is styled to cover the viewport entirely (background-size: cover).
- A centered form container (.purchase-container) allows users to input sell details.

2. **Form Elements**:
    - **User ID Input**: A text input field where users enter their unique identifier.
    - **Item ID Input**: A text input field for specifying the item to be sold.
    - **Quantity Input**: A number input field to specify the number of items being sold.
    - **Submit Button**: A button labeled "Submit Sell" to send the sell request.

3. **CSS Styling**:
    - Clean, minimalistic design with rounded corners, box shadows, and proper spacing for readability.
    - Responsiveness through a centered container with a max-width of 400px and a box-sizing approach.
    - Color-coded messages for error (red) and success (green).

4. **Success and Error Messages**:
    - The tags with IDs success-message and error-message are placeholders for dynamic user feedback based on the backend response.

JavaScript Functionality:

- **Form Submission Handling**:
  - The submit event is intercepted with preventDefault() to avoid default browser submission behavior.
  - The form inputs are gathered and sent via the Fetch API.
- **Backend Integration**:
  - Sends a POST request to /queries/sell-product with a JSON payload containing userID, itemID, and quantity.
  - Handles server responses:
    - **Success**: Displays a success message with details from the response.
    - **Error**: Displays an error message if the response indicates a failure or if there's a network issue.

Intended Use:

1. **Frontend Workflow**:
   - A seller enters their User ID, the ID of the product they wish to sell, and the quantity.
   - Upon clicking "Submit Sell," the form sends the details to the backend.

2. **Backend Processing**:
   - The backend is expected to validate the input, update inventory, and return appropriate success/error responses.

3. **User Feedback**:

- Clear messages help the user understand whether their sell request was successful or why it failed.

---

signupPage (signup.html)

This HTML document provides a **Sign-Up Page** for an **Inventory Management System**. It features a well-structured user interface, user-friendly design, and JavaScript functionality for integration with the backend. Here's a detailed breakdown:

Key Components

1. Navbar

- Displays the system's name ("INVENTORY MANAGEMENT SYSTEM") at the top of the page.
- Fixed at the top with a dark background for clear visibility (position: sticky).

2. Page Styling

- Background:
  - A colorful blurred image (background-image).
  - Styled to cover the viewport entirely (background-size: cover).
- Fonts and Colors:
  - Fonts are modern and sans-serif.
  - Bright colors for text ensure contrast against the background.

3. Sign-Up Form

The main focus of the page, placed in a centered white container (.signup-container) for better visibility.

- **Form Fields**:

- **Username**: Text input for the user's chosen username.
- **Email**: Email input field with validation.
- **Password**: Password input (hidden characters for privacy).
- **Phone**: Tel input field for the user's phone number.

- **Submit Button**:
  - Prominent green button with a hover effect.
  - Submits the form data to the backend.
- **Error and Success Messages**:
  - Displays dynamic feedback using elements (#error-message, #success-message).
  - Errors appear in red, success messages in green.
- **Login Link**:
  - Redirects users with existing accounts to the login page.

4. Footer

- Provides contact information (phone and email) with clickable links.
- Fixed at the bottom of the page for consistent visibility.

JavaScript Functionality

The script enhances the form by handling submission asynchronously and providing real-time feedback.

1. **Form Submission**:
   - Prevents default browser submission with e.preventDefault().
   - Collects input values (username, email, password, phone).
2. **Backend Integration**:
   - Sends a POST request to /usersignup using the Fetch API with the input data as a JSON payload.

- Handles server responses:
  - **Error**: Displays error messages from the server (e.g., "Email already registered").
  - **Success**: Displays a success message (e.g., "Successfully Signed Up").

3. **Feedback**:
   - Clears previous messages (innerText = '') before displaying new feedback.

Features of the Design

1. **Responsive Layout**:
   - The form container adjusts its size to fit different screen widths (max-width: 400px).

2. **User-Focused Design**:
   - Simple navigation with clear instructions and feedback.
   - Bright, visually appealing design with easy-to-read text.

3. **Accessibility**:
   - Labels and placeholders make the form accessible.
   - Clickable links in the footer enhance usability.

Intended Use Case

The page is designed for new users to create an account in the inventory management system. It ensures:

- Easy navigation and signup.
- Error handling with real-time feedback.
- Backend integration for dynamic data processing.

REFERENCES

[1]https://www.theseus.fi/bitstream/handle/10024/752452/Le_Linh.pdf?sequence=2

[2]https://www.w3resource.com/projects/sql/sql-projects-on-inventory-management-system.php

[3]https://www.geeksforgeeks.org/how-to-design-er-diagrams-for-inventory-and-warehouse-management/

[4]https://vertabelo.com/blog/data-model-for-inventory-management-system/