SUPPLY CHAIN MANAGEMENT DASHBOARD FOR CONSTRUCTION COMPANY

Member 1	Manu Narayan Hegde	PES1UG22CS335
Member 2	Mohammed Furqaan	PES1UG22CS352

Project Description:

Product Perspective:
The SCM dashboard is an application designed specifically for the construction industry. The Dashboard provides a centralised solution for managing supply chains across multiple construction projects and companies. It operates independently, with no reliance on existing systems, and is intended for use by various user types, including company admins, suppliers,to optimise procurement, inventory, and resource management.

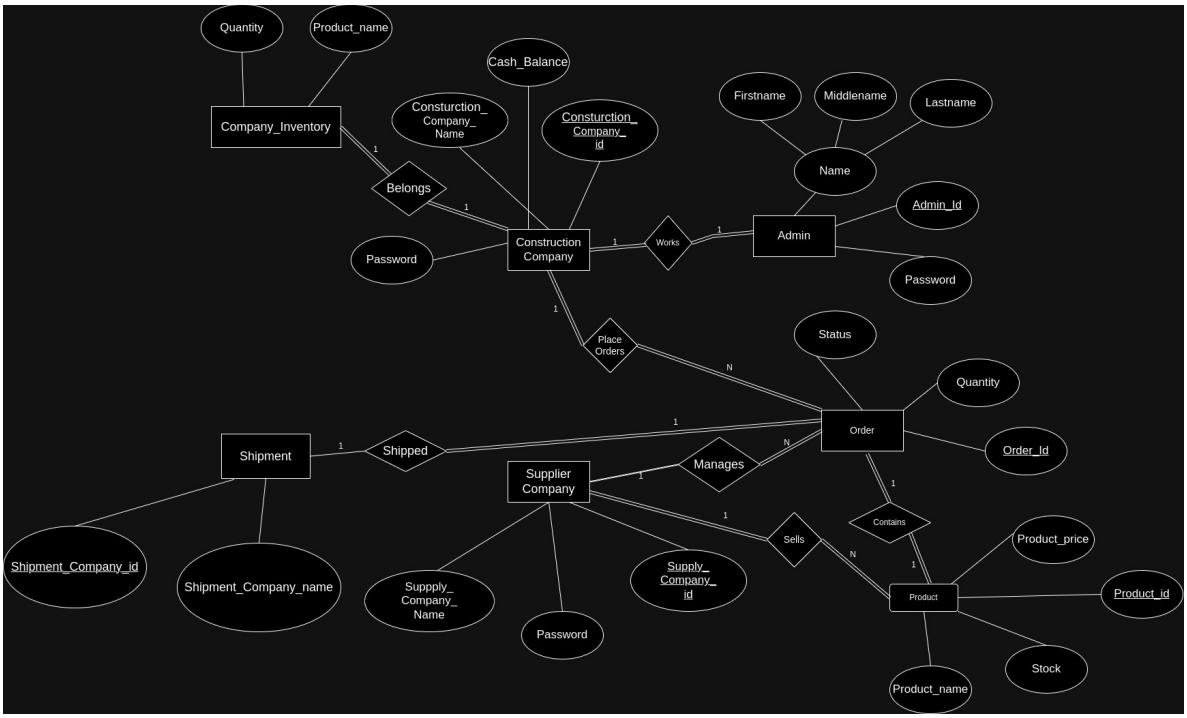
The SCM dashboard enables the following key functions:
User Authentication: Secure login with role-based access.
Multi-Company Management: Support for managing multiple companies with segregated data.
Inventory Tracking: Real-time management of construction materials and stock levels. Order Management: Placing, tracking, and processing orders.
Supplier Management: Managing supplier information and evaluating performance.

Project Scope:
The SCM dashboard is a web-based platform designed for the construction industry to manage inventory, orders, and supplier relationships. Its purpose is to optimise procurement, resource allocation, and supply chain operations across multiple construction projects. By Supporting multi-company environments, the dashboard enables construction firms to collaborate efficiently with suppliers, reducing delays and improving project delivery. Thesoftware aligns with corporate goals by increasing operational efficiency, reducing costs, and ensuring timely project execution.

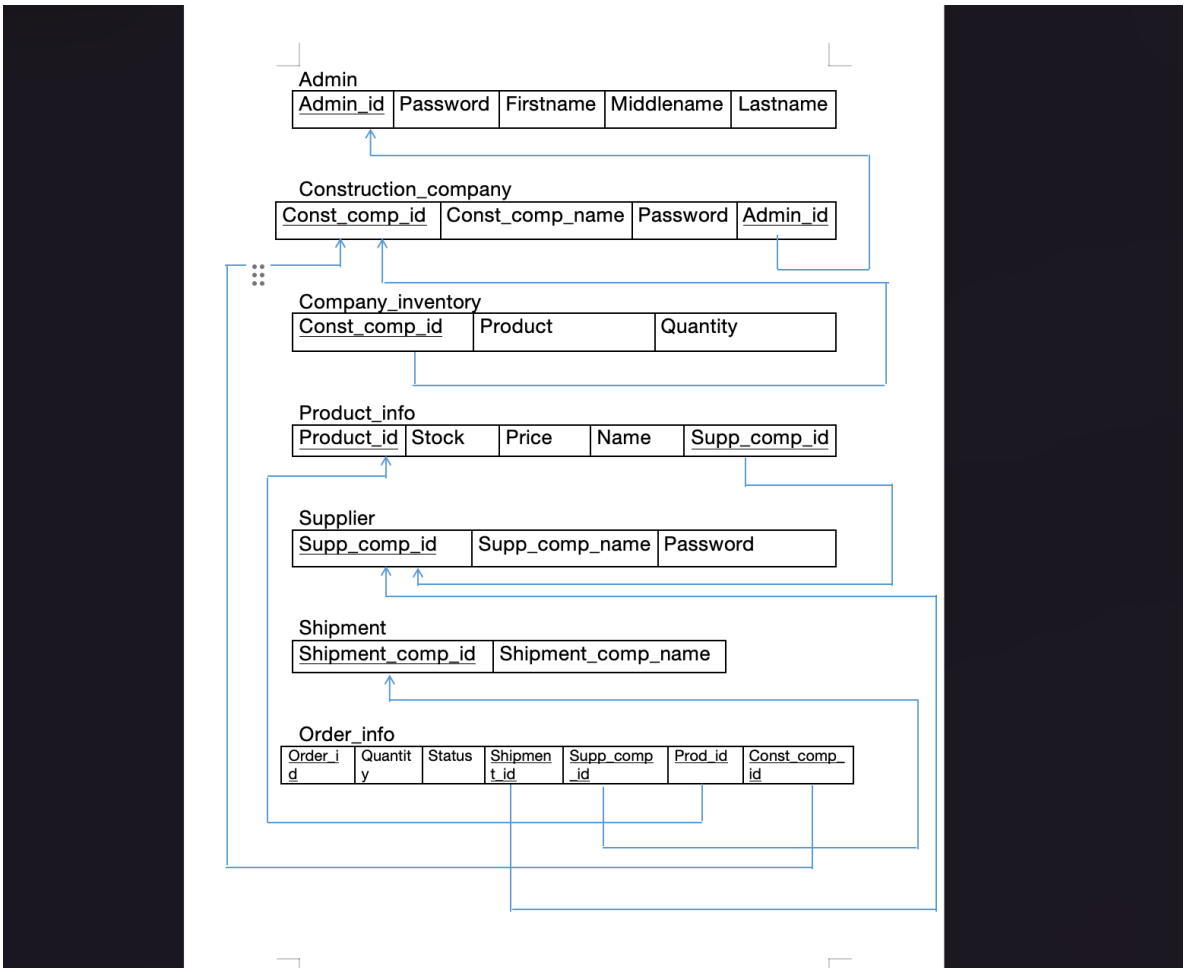
User Requirement Specification:

List of Tools used:
Python, Visual Studio code , MySQL, SQL, Streamlit

ER DIAGRAM:


Quantity
Product_name
Cash_Balance
Company_Inventory
Consturction_ Company_ Name
Consturction_ Company_ id
Firstname
Middlename
Lastname
Name
Belongs
Admin_Id
Password
Construction Company
Works
Admin
Password
Place Orders
Status
Quantity
Order
Order_Id
Shipment
Shipped
Supplier Company
Manages
Contains
Sells
Product_price
Shipment_Company_id
Shipment_Company_name
Suppply_ Company_ Name
Supply_ Company_ id
Password
Product
Product_id
Stock
Product_name

Relational Schema:



Triggers:

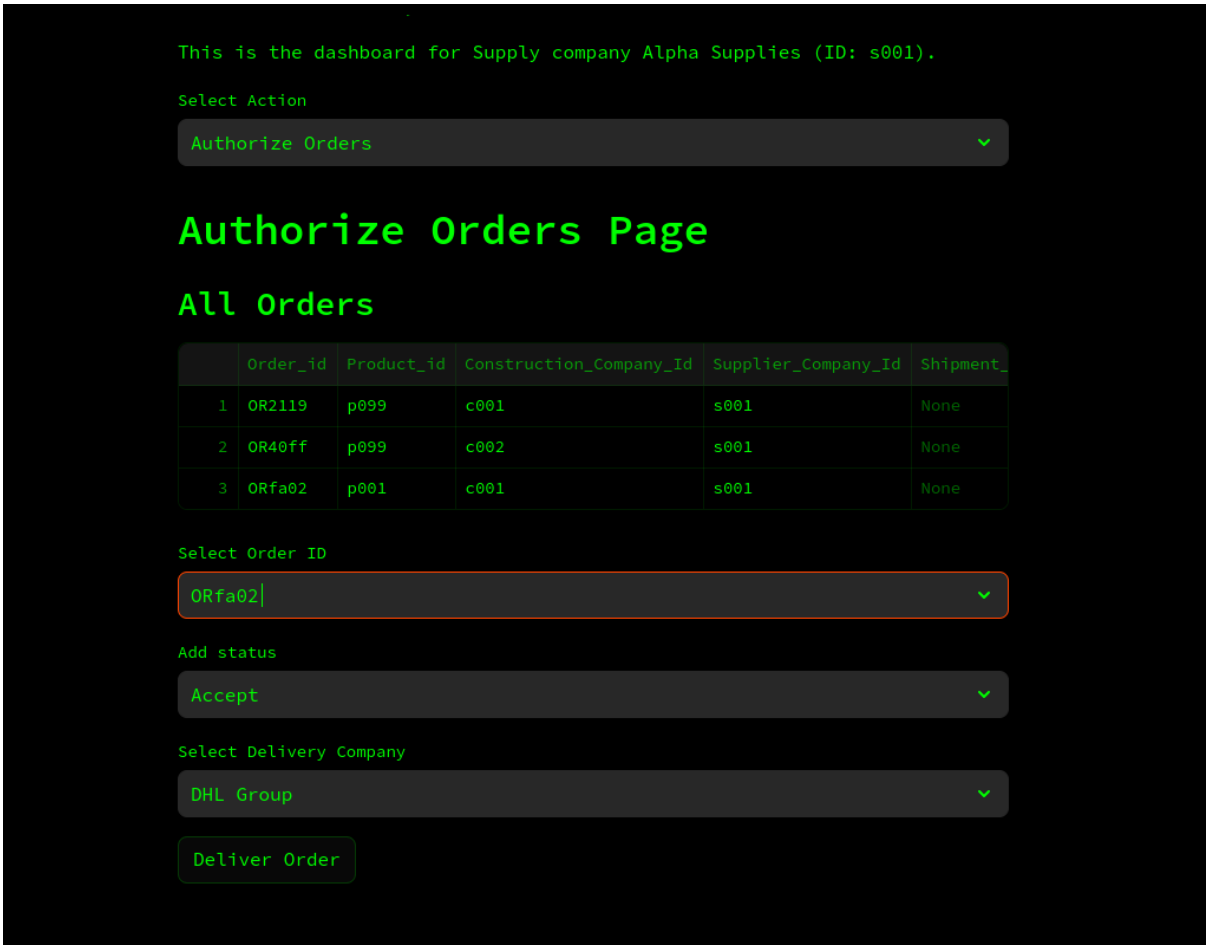
This trigger is used to restock a product if the quantity ordered exceeds the current inventory available in the supplier company's stock.

```
DROP TRIGGER IF EXISTS update_stock_on_order_accept;

CREATE TRIGGER update_stock_on_order_accept
       AFTER UPDATE ON Order_info
       FOR EACH ROW
       BEGIN
           DECLARE current_stock INT;

           IF NEW.Status = 'Accepted' THEN
               -- Check current stock
               SELECT Stock INTO current_stock
               FROM Product_info
               WHERE Product_id = NEW.Product_id;

               -- Restock if necessary
               IF current_stock < NEW.Quantity THEN
                   UPDATE Product_info
                   SET Stock = current_stock + (NEW.Quantity * 2)
                   WHERE Product_id = NEW.Product_id;
               END IF;

               -- Deduct the ordered quantity from stock
               UPDATE Product_info
               SET Stock = Stock - NEW.Quantity
               WHERE Product_id =
               NEW.Product_id;
           END
       IF;
       END;
```

Before:

```
mysql> select * from Product_info;
+------------+--------------+---------------+--------------------+-------+
| Product_id | Product_name | Product_price | Supplier_Company_id | Stock |
+------------+--------------+---------------+--------------------+-------+
| p001       | CEMENT       |        500.00 | s001               |  8980 |
| p002       | BRICKS       |        250.00 | s001               |  9500 |
| p003       | IRON         |        300.00 | s002               |  9950 |
| p004       | GOLD         |       1000.00 | s002               | 10000 |
| p005       | BRONZE       |        100.00 | s003               | 10000 |
| p006       | SILVER       |        400.00 | s003               | 10000 |
| p007       | COPPER       |        150.00 | s004               | 10000 |
| p008       | STEEL        |        350.00 | s004               |  9900 |
| p099       | NAIL         |        100.00 | s001               |  9990 |
+------------+--------------+---------------+--------------------+-------+
9 rows in set (0.00 sec)
```

After:

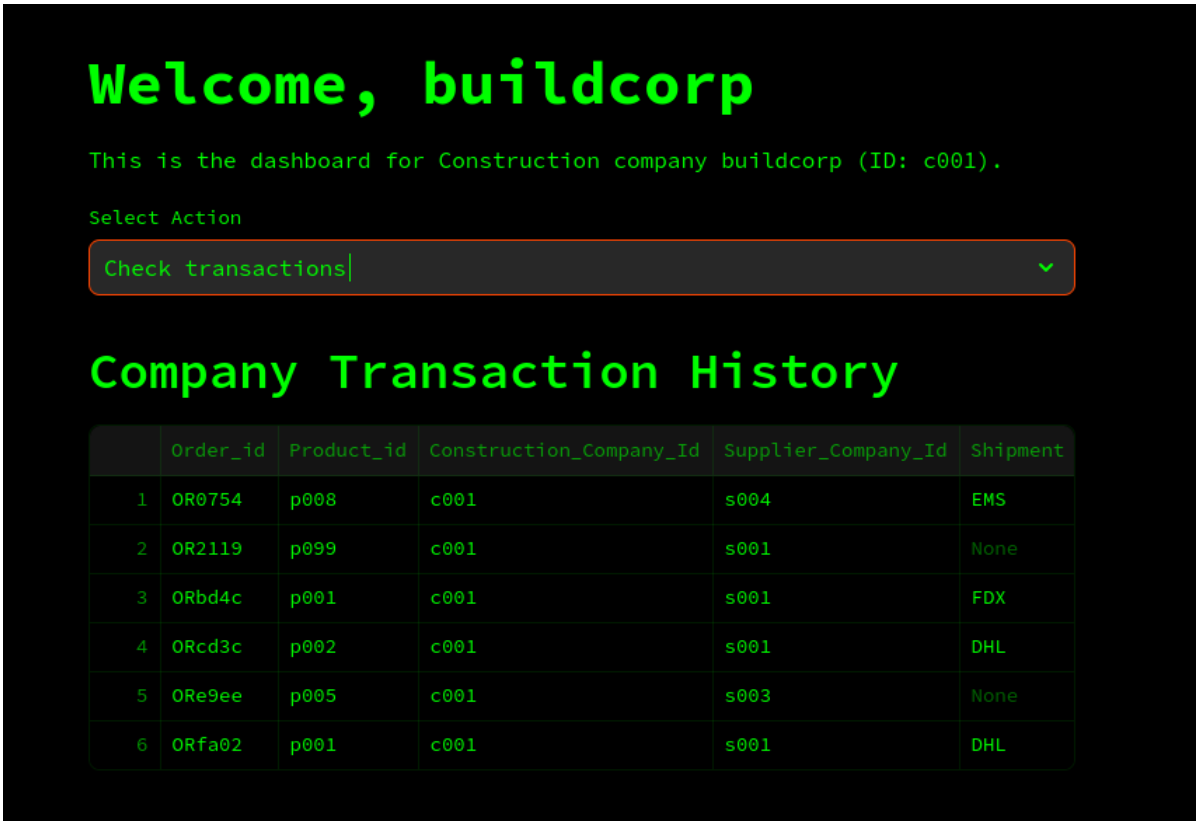



```
mysql> select * from Product_info;
+------------+--------------+---------------+----------------------+-------+
| Product_id | Product_name | Product_price | Supplier_Company_id | Stock |
+------------+--------------+---------------+----------------------+-------+
| p001       | CEMENT       |        500.00 | s001                 | 18980 |
| p002       | BRICKS       |        250.00 | s001                 |  9500 |
| p003       | IRON         |        300.00 | s002                 |  9950 |
| p004       | GOLD         |       1000.00 | s002                 | 10000 |
| p005       | BRONZE       |        100.00 | s003                 | 10000 |
| p006       | SILVER       |        400.00 | s003                 | 10000 |
| p007       | COPPER       |        150.00 | s004                 | 10000 |
| p008       | STEEL        |        350.00 | s004                 |  9900 |
| p099       | NAIL         |        100.00 | s001                 |  9990 |
+------------+--------------+---------------+----------------------+-------+
9 rows in set (0.00 sec)
```

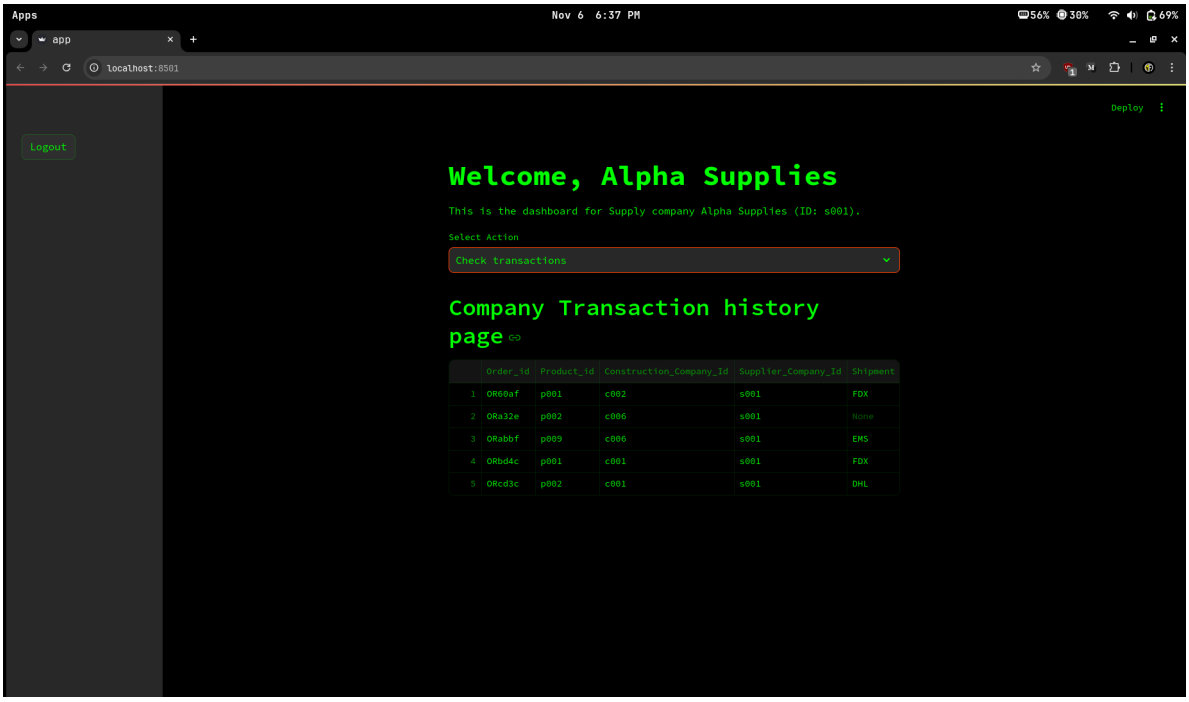
Procedure:

This procedure gives all the transactions made by the construction company or supplier company respectively.

```
drop procedure if exists get_order_info;

create procedure if not exists

get_order_info(
                    IN in_company_id VARCHAR(50),
                    IN company_type ENUM('CONSTRUCTION', 'SUPPLIER')
                )
                BEGIN
                    if company_type = 'CONSTRUCTION' THEN
                        SELECT * FROM Order_info WHERE
Construction_Company_Id = in_company_id;
                    elseif company_type = 'SUPPLIER' THEN
                        SELECT * FROM Order_info WHERE
Supplier_Company_Id = in_company_id;
                    end if;
                end
```



	Order_id	Product_id	Construction_Company_Id	Supplier_Company_Id	Shipment
1	OR0754	p008	c001	s004	EMS
2	OR2119	p099	c001	s001	None
3	ORbd4c	p001	c001	s001	FDX
4	ORcd3c	p002	c001	s001	DHL
5	ORe9ee	p005	c001	s003	None
6	ORfa02	p001	c001	s001	DHL


Apps
Nov 6 6:37 PM
56% 30% 69%
app
localhost:8501
Deploy
Logout
Welcome, Alpha Supplies
This is the dashboard for Supply company Alpha Supplies (ID: s001).
Select Action
Check transactions
Company Transaction history page
Order_id Product_id Construction_Company_Id Supplier_Company_Id Shipment
1 OR60af p001 c002 s001 FDX
2 ORa32e p002 c006 s001 None
3 ORabbf p009 c006 s001 EMS
4 ORbd4c p001 c001 s001 FDX
5 ORcd3c p002 c001 s001 DHL

Create Tables:

Creating Admin Table that stores all the information of admin of the construction company.

```
# Admin Table
CREATE TABLE IF NOT EXISTS Admin (
            Admin_id VARCHAR(50) PRIMARY KEY,
            Firstname VARCHAR(50) NOT NULL,
            Middlename VARCHAR(50),
            Lastname VARCHAR(50) NOT NULL,
            Password VARCHAR(255) NOT NULL
        )
```

```
mysql> select * from Admin;
+----------+-----------+------------+----------+--------------------------------------------------------------------------------------------------------------+
| Admin_id | Firstname | Middlename | Lastname | Password                                                                                                     |
+----------+-----------+------------+----------+--------------------------------------------------------------------------------------------------------------+
| a001     | john      |            | doe      | $argon2id$v=19$m=65536,t=3,p=4$rYAQFyeSzbMZd0PzHEW0Vw$Ikd8sUVipTKodVCBl1FbaEQWiFDCMJ3uzA0hHRdsg/c |
| a002     | manu      |            | hegde    | $argon2id$v=19$m=65536,t=3,p=4$worgQoRoq03r/vjtoqqIFA$4GjwEYGNq7nd5+9orr3qDY+Ufol9+LWQbuoi/WSYDQI |
| a003     | jane      | mary       | doe      | $argon2id$v=19$m=65536,t=3,p=4$tKbN8J0+R+IeOAlthdGlRg$313zLWFyg9d7KzE6i9OKtBT0Qqg+wyJ9wbalbeOY1Pc |
| a004     | alice     |            | johnson  | $argon2id$v=19$m=65536,t=3,p=4$qPeAf1khWv4vOYJOGHIaMw$GY9UQ/vHOZ4lmqIb+unDgcbN7S+6m9mCt8PSrqb0zow |
| a005     | mohammed  |            | furqaan  | $argon2id$v=19$m=65536,t=3,p=4$FKbr1M/4SLbAMD1CpPlheQ$ClbwJF/Isks8SALErIegkbMoSh1O5tKhoFPy8VuclE4 |
| a006     | bob       |            | brown    | $argon2id$v=19$m=65536,t=3,p=4$bXlA4Wsht4MnZFFF4TMh5w$AdbhMvP1aa4nFrXuxwJnnAYIXFXfkdJ4ihJH+wDiopg |
+----------+-----------+------------+----------+--------------------------------------------------------------------------------------------------------------+
6 rows in set (0.00 sec)
```

Creating Construction company table that stores all the construction companies.

```
# Construction_Company Table
CREATE TABLE IF NOT EXISTS Construction_Company (
            Construction_Company_id VARCHAR(50) PRIMARY KEY,
            Construction_Company_name VARCHAR(100) NOT NULL UNIQUE,
            Cash_Balance DECIMAL(20, 2) DEFAULT 10000000,
            Admin_id VARCHAR(50),
            FOREIGN KEY (Admin_id) REFERENCES Admin(Admin_id)
        )
```

```
mysql> select * from Construction_Company;
+-------------------------+---------------------------+--------------+----------+
| Construction_Company_id | Construction_Company_name | Cash_Balance | Admin_id |
+-------------------------+---------------------------+--------------+----------+
| c001                    | buildcorp                 |  15000010.00 | a001     |
| c002                    | ABC Constructions         |  10500000.00 | a002     |
| c003                    | XYZ Builders              |  10000000.00 | a003     |
| c004                    | summit builders           |  10000000.00 | a004     |
| c005                    | peaky blinders            |  89985000.00 | a005     |
| c006                    | beta builders             |   9999500.00 | a006     |
+-------------------------+---------------------------+--------------+----------+
6 rows in set (0.00 sec)
```

Supplier Company Table:

Creating Supplier company table that stores all the Supplier companies.

```
cursor.execute("""
       CREATE TABLE IF NOT EXISTS Supplier_Company (
           Supplier_Company_id VARCHAR(50) PRIMARY KEY,
           Supplier_Company_name VARCHAR(100) NOT NULL UNIQUE,
           Password VARCHAR(255) NOT NULL
       )
   """)
```

```
mysql> select * from Supplier_Company;
+--------------------+---------------------+--------------------------------------------------------------------------------------------------------------+
| Supplier_Company_id | Supplier_Company_name | Password                                                                                                     |
+--------------------+---------------------+--------------------------------------------------------------------------------------------------------------+
| s001               | Alpha Supplies      | $argon2id$v=19$m=65536,t=3,p=4$YaYhWNR/TK6gXKN7BlQeZQ$A3OLkpsd34pmXdyvkds57KaF5dIIi8NGNFrbsycnC34 |
| s002               | Beta Suppliers      | $argon2id$v=19$m=65536,t=3,p=4$s+9XWSI0uWu3lfJeJfCMmg$H+bw1rv45+HrVaCRLdslaEK9xtxZ7rJ4hkP0W7O3AC4 |
| s003               | Gamma Suppliers     | $argon2id$v=19$m=65536,t=3,p=4$ZR2vsC3YBOf2ZZdhq6OY6w$FYAiLMD262alXnUbTnt+qV8iZpUVIbMjXfL0IY3J9iI |
| s004               | Epsilon Suppliers   | $argon2id$v=19$m=65536,t=3,p=4$JaySU/C8McO1M6R0B2iDow$DUawvxL7eMW8oEDug27ZnOYcBsremlckIC8MbH7bSlQ |
| s005               | Delta Suppliers     | $argon2id$v=19$m=65536,t=3,p=4$BPVr3FEVxaHMdZEeeaI0Iw$N6ObVtJWL7DRa4oeTfGbZCOfvJ5PwM1gHdfitBuu+JM |
+--------------------+---------------------+--------------------------------------------------------------------------------------------------------------+
5 rows in set (0.00 sec)
```

Shipment Company Table:

Creating Shipment company table that stores all the shipment companies for delivery of orders

```
CREATE TABLE IF NOT EXISTS Shipment_Company (
           Shipment_Company_id VARCHAR(50) PRIMARY KEY,
           Shipment_Company_name VARCHAR(100) NOT NULL
           UNIQUE
       )
```

```
mysql> select * from Shipment_Company;
+---------------------+-----------------------+
| Shipment_Company_id | Shipment_Company_name |
+---------------------+-----------------------+
| DHL                 | DHL Group             |
| EMS                 | Express Mail Service  |
| FDX                 | FedEx Corp            |
| TNT                 | TNT Express           |
| UPS                 | United Parcel Service |
+---------------------+-----------------------+
5 rows in set (0.00 sec)
```

Product info table:
Stores list of all products sold by the Supplier companies.

```
CREATE TABLE IF NOT EXISTS Product_info (
            Product_id VARCHAR(50) PRIMARY
            KEY,
            Product_name VARCHAR(100) NOT NULL,
            Product_price DECIMAL(12, 2) NOT
            NULL,Supplier_Company_id VARCHAR(50), Stock INT
            DEFAULT 10000,
            FOREIGN KEY (Supplier_Company_id) REFERENCES
Supplier_Company(Supplier_Company_id)
        )
```

```
mysql> select * from Product_info;
+------------+--------------+---------------+---------------------+-------+
| Product_id | Product_name | Product_price | Supplier_Company_id | Stock |
+------------+--------------+---------------+---------------------+-------+
| p001       | CEMENT       |        500.00 | s001                | 18980 |
| p002       | BRICKS       |        250.00 | s001                |  9500 |
| p003       | IRON         |        300.00 | s002                |  9950 |
| p004       | GOLD         |       1000.00 | s002                | 10000 |
| p005       | BRONZE       |        100.00 | s003                | 10000 |
| p006       | SILVER       |        400.00 | s003                | 10000 |
| p007       | COPPER       |        150.00 | s004                | 10000 |
| p008       | STEEL        |        350.00 | s004                |  9900 |
| p099       | NAIL         |        100.00 | s001                |  9990 |
+------------+--------------+---------------+---------------------+-------+
9 rows in set (0.00 sec)
```

Order Info table:

This table stores all the placed orders with their order status and shipment delivery company.

```
CREATE TABLE IF NOT EXISTS Order_info (
            Order_id VARCHAR(50) PRIMARY
            KEY,
            Product_id VARCHAR(50),
            Construction_Company_Id VARCHAR(50),
            Supplier_Company_Id VARCHAR(50),
            Shipment_Company_Id VARCHAR(50) DEFAULT NULL,
            Quantity INT NOT NULL,
            Cost DECIMAL(12, 2),
            Status ENUM('Pending', 'Accepted', 'Rejected') DEFAULT
'Pending',
            FOREIGN KEY (Product_id) REFERENCES
Product_info(Product_id),
            FOREIGN KEY (Construction_Company_Id) REFERENCES
Construction_Company(Construction_Company_id),
            FOREIGN KEY (Supplier_Company_Id) REFERENCES
Supplier_Company(Supplier_Company_id),
            FOREIGN KEY (Shipment_Company_Id) REFERENCES
Shipment_Company(Shipment_Company_id)
        )
```

Order_id	Product_id	Construction_Company_Id	Supplier_Company_Id	Shipment_Company_Id	Quantity	Cost	Status
OR0754	p008	c001	s004	EMS	100	35000.00	Accepted
OR1c7b	p003	c005	s002	TNT	50	15000.00	Accepted
OR2119	p099	c001	s001	NULL	50	5000.00	Pending
OR40ff	p099	c002	s001	NULL	500	50000.00	Pending
OR60af	p001	c002	s001	FDX	1000	500000.00	Accepted
OR914b	p006	c004	s003	NULL	1000	400000.00	Pending
ORa32e	p002	c006	s001	NULL	1000	250000.00	Rejected
ORabbf	p009	c006	s001	EMS	10	500.00	Accepted
ORbd4c	p001	c001	s001	FDX	20	10000.00	Accepted
ORc062	p004	c002	s002	NULL	10	10000.00	Rejected
ORcd3c	p002	c001	s001	DHL	500	125000.00	Accepted
ORd216	p001	c001	s001	NULL	10	5000.00	Pending
ORe9ee	p005	c001	s003	NULL	500	50000.00	Pending
ORf79e	p007	c005	s004	NULL	100	15000.00	Rejected
ORfa02	p001	c001	s001	DHL	10000	5000000.00	Accepted

15 rows in set (0.00 sec)

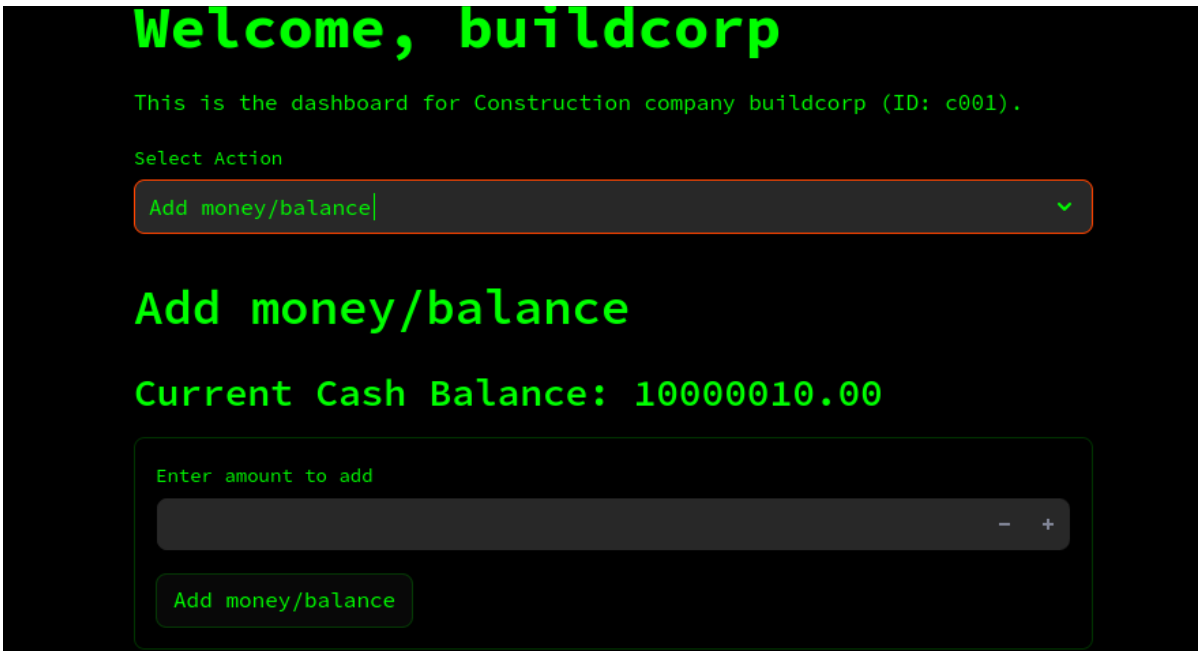
Company Inventory Table:
Stores all the items owned by the Construction company

```
Construction_Company_Id VARCHAR(50) NOT NULL,
           Product_name VARCHAR(100) NOT NULL,
           Quantity INT NOT NULL,
           FOREIGN KEY (Construction_Company_Id) REFERENCES
Construction_Company(Construction_Company_id)
```

```
mysql> select * from Company_Inventory;
+--------------------------+--------------+----------+
| Construction_Company_Id | Product_name | Quantity |
+--------------------------+--------------+----------+
| c001                     | BRICKS       |      500 |
| c002                     | CEMENT       |     1000 |
| c001                     | CEMENT       |    10010 |
| c005                     | IRON         |       50 |
| c001                     | STEEL        |      100 |
| c006                     | NAIL         |       10 |
+--------------------------+--------------+----------+
6 rows in set (0.00 sec)
```

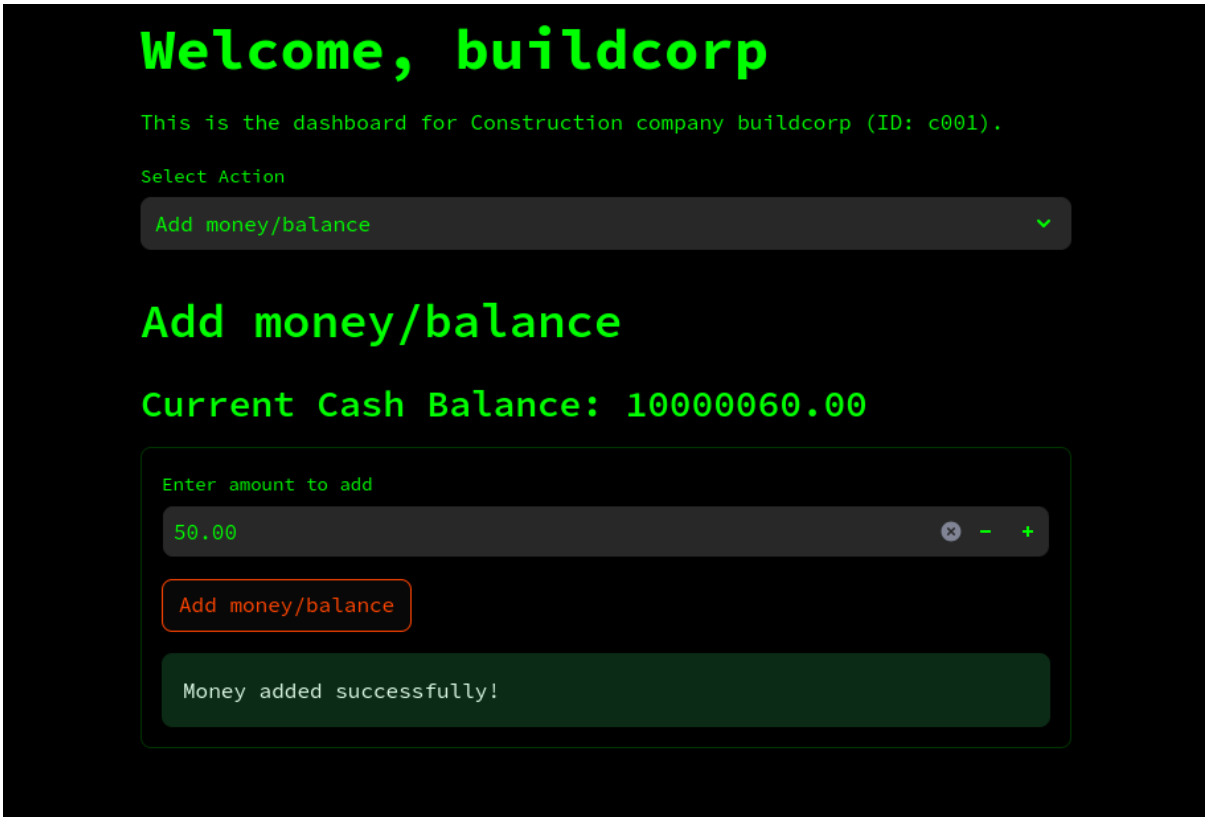
Read Operations:
This is used to display the cash balance of the construction company

```
SELECT Cash_Balance FROM Construction_Company WHERE
Construction_Company_id = %s
```



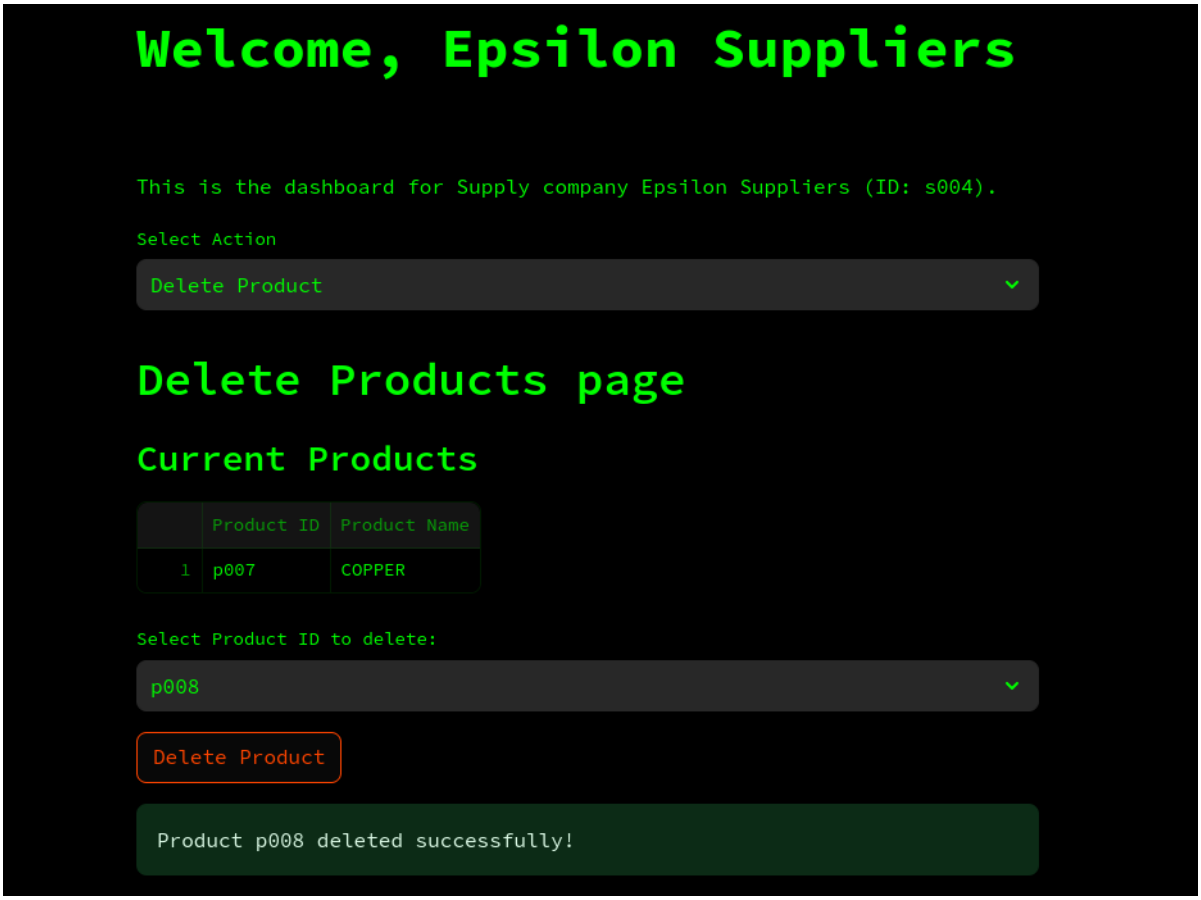
Update Operations:
Updates the Cash balance when money is added in the Construction company

```
UPDATE Construction_Company
SET Cash_Balance = Cash_Balance + %s
WHERE Construction_Company_id = %s
```



Delete Operation:
Used to delete a product which will not be sold again by the Supplier company.

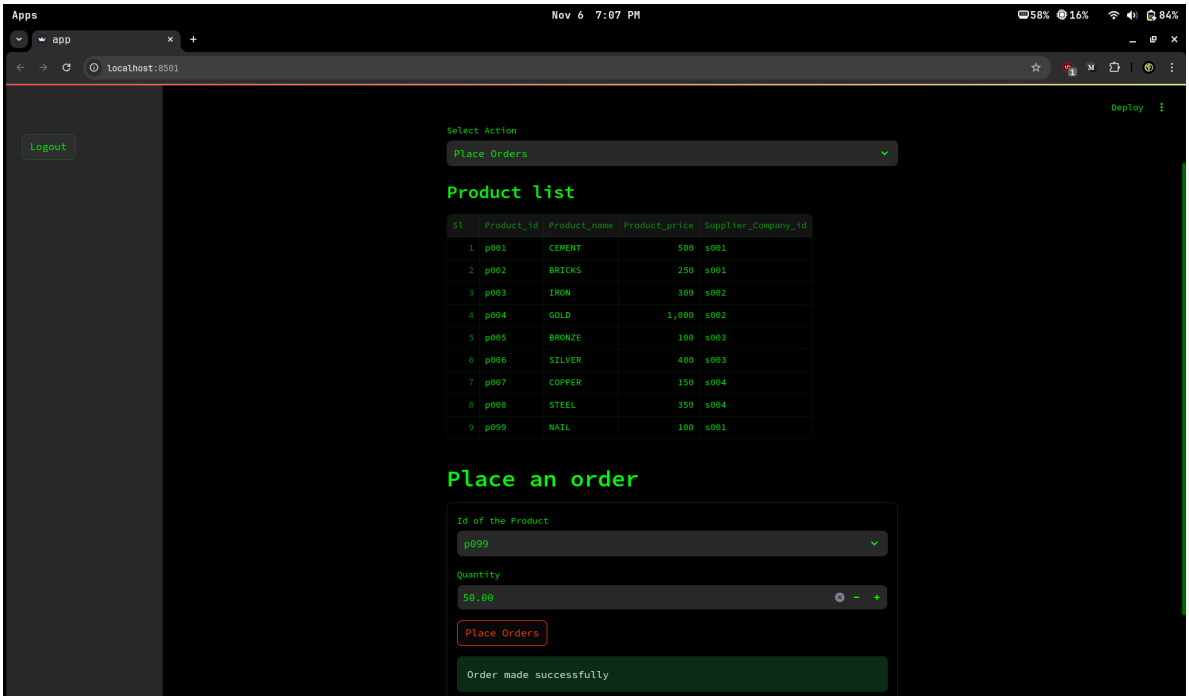
```
DELETE FROM Product_info WHERE Product_id = %s
```



Nested Query:

This is used to place orders for the Construction company selects the product and the corresponding supplier company.

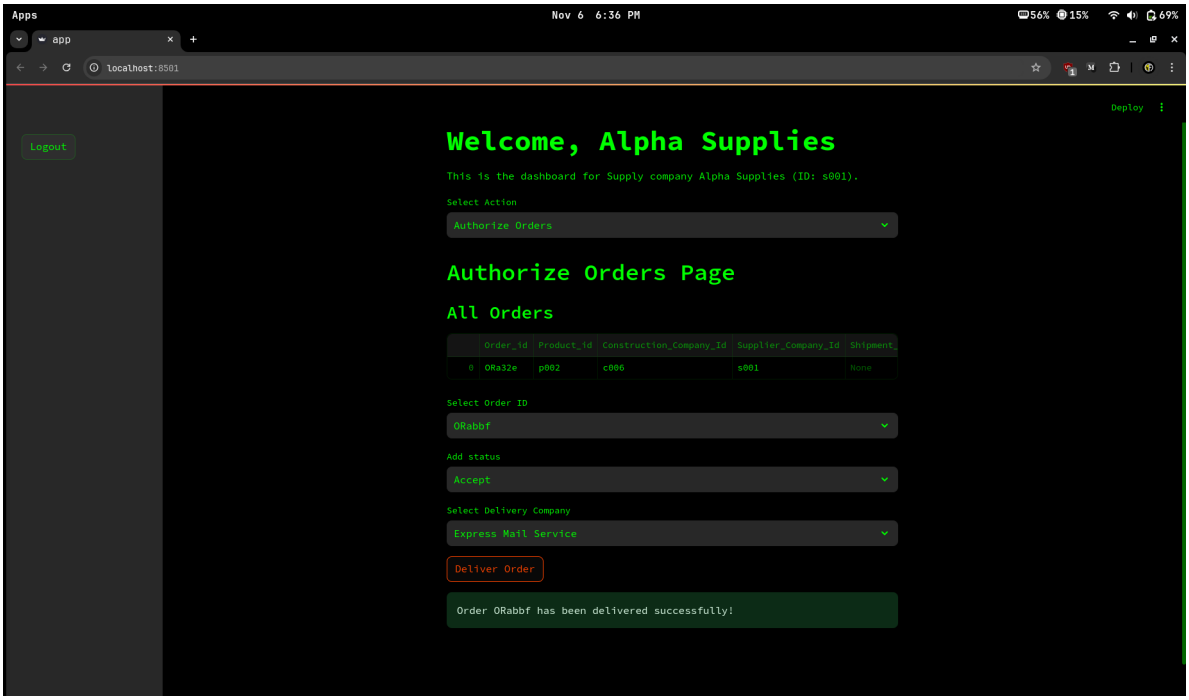
```
WITH ProductDetails AS (
    SELECT
        Supplier_Company_id,
        Product_price
    FROM Product_info
    WHERE Product_id =
    %s


),
OrderDetails AS (
    SELECT
        (pd.Product_price * %s) AS total_price,
        cc.Cash_Balance
    FROM ProductDetails pd
    JOIN Construction_Company cc ON
cc.Construction_Company_id = %s
)
SELECT
    pd.Supplier_Company_id,
    od.total_price,
    od.Cash_Balance
FROM ProductDetails pd, OrderDetails od
```



Join Query:

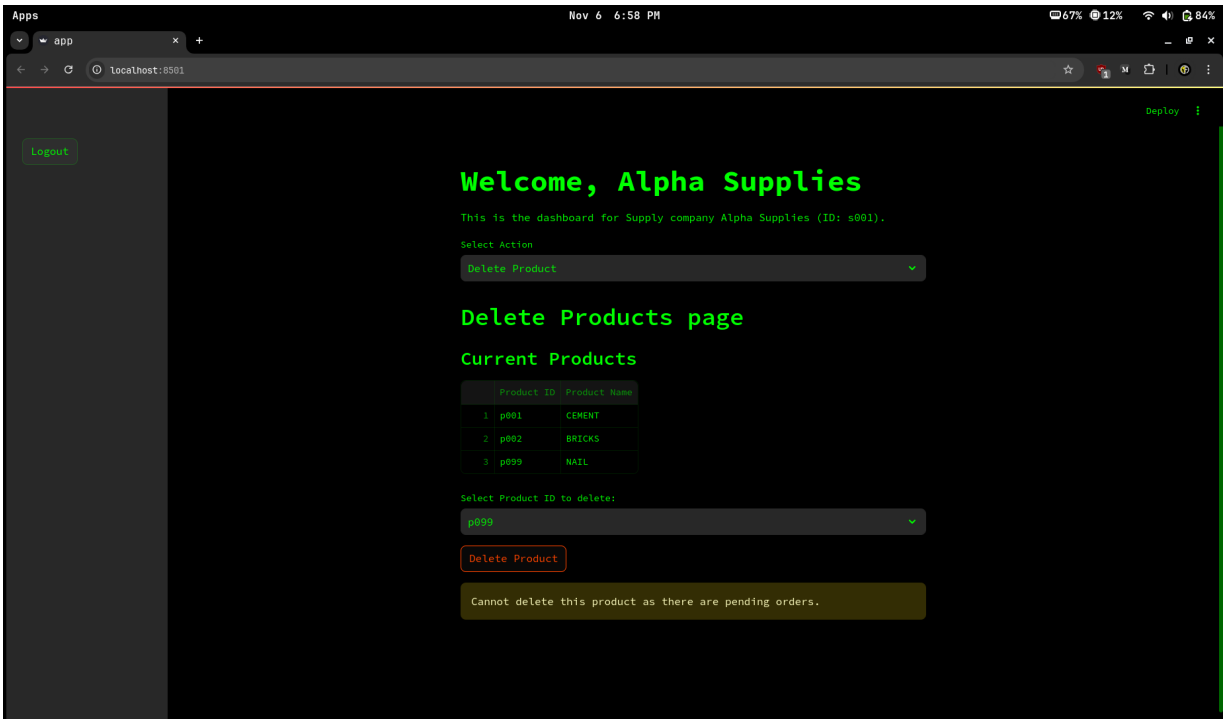
This is used in the supplier company to Authorize orders which can be accept or reject

```
SELECT o.Product_id, o.Quantity, o.Cost, o.Construction_Company_Id,
p.Stock
FROM Order_info o
JOIN Product_info p ON o.Product_id =
p.Product_id WHERE o.Order_id = %s AND o.Status =
'Pending' AND o.Supplier_Company_Id = %s
```



Aggregated Query:
This is used to display all the products that are being sold by the Supplier company to delete it.

```
SELECT COUNT(*)
FROM Order_info
WHERE Product_id = %s AND Status = 'Pending'
```



GitHub repo link: https://github.com/Manuhegde1309/SCM-construction-company