DATABASE LAB

PROJECT FINAL DELIVERABLE

Name: Amna Shahid, Fatima Qurban.
Roll.no: 21I-1148, 21I-1195.
Company Name: InvenTrack
(IMS - Inventory Management System)
Company Logo:

INVENTRACK

Project Title: Inventory Management System.

Introduction:
An Inventory Management System (IMS) is a software application or system that helps businesses efficiently track, manage, and control their inventory. Real time monitoring of products, management of purchases and sales orders, and insurance of stock levels to an optimal level are the features of this inventory management system.

Scope:
The scope of this project encompasses product Management, Inventory Tracking, purchase orders and supplier management, sales order management, point of sales, Return and Refunds, Reports and Analytics and Access Control.

Objectives:
1. To optimize the inventory levels in order to meet the customer demands.
2. To minimize the costs associated with warehouses, and product records.
3. To enhance the quality of the supply chain with increased coordination with suppliers and distributors.
4. To keep a check on the remaining product stock in order to prevent any loss.
5. To decrease the chances of wrong decisions regarding stock, as the real time access to the data will be provided.
6. To keep a check on any mismanagement and theft in the inventory as the record of each product will be kept.
7. To minimize the human resources, which will result in decreased cost.
8. To enhance customer service and satisfaction, the less the delay in orders due to inaccurate inventory records, the greater will be the customer satisfaction.

Problem Statement:
InvenTrack is facing some issues regarding stock and operational costs due to the manual management system of the company. In order to deal with such issues and to increase the efficiency of the management, an automated inventory management system is needed.

Description:
Manual management has resulted in increasing issues regarding the stocks and the costs. InvenTrack's system lacks real-time visibility of products and stocks which causes a major increase in operational costs. In order to cope up with these challenges an efficient inventory management system is needed which gives a real-time view of the inventory and enhances the data accuracy. Secure management and tracking of products and orders will be some major features of the automated inventory management system. It will help to improve the costs and increase the ability to meet the customers' needs on time.

Business Process:

The business process of our Inventory Management System(InvenTrack) are following;

- **Product Management:** Adding, updating, and deleting products in the inventory. Tracking product details such as name, description, price, and stock levels.
- **Inventory Tracking:** Managing stock replenishment and reordering based on predefined thresholds.
- **Purchase Orders:** Creating and managing purchase orders for restocking products
- **Supplier Management:** Tracking supplier information and purchase history.
- **Sales Order Management:** Processing customer sales orders. Managing order fulfillment.
- **Point of Sale (POS):** Handling transactions. Generating sales receipts and customer receipts.
- **Order Returns:** Handling product returns, and updating it back to inventory
- **Employee Management:** Managing employee information and shifts.
- **Reporting and Analytics:** Generating reports on sales and inventory.
- **Security and Access Control:** Implementing role-based access control to restrict user access to sensitive data. Like it requires the manager's login before the salesman could deal with inventory.
- **Customer Management:** Tracking customer purchase history.

Functional and Nonfunctional Requirements of Inventory Management System

1. Product Management:
   - **Functional Requirements:**

The Inventrack (Inventory Management System) shall provide the employee/manager with the ability to add new products with details such as name, description, price, and stock levels.
The IMS shall provide the manager with the ability to update existing product information.
The IMS shall allow the manager to delete products from the inventory.

   - **Non-functional Requirements:**

The IMS should support multiple product categories.
The IMS should store the product data securely and easily retrieve it.

2. Inventory Tracking:

- **Functional Requirements:**

The IMS shall have the ability to track stock levels for products.
The manager shall set predefined thresholds for reordering.
The IMS shall automatically generate restocking alerts when stock falls below thresholds.

- **Non-functional Requirements:**

The IMS should give real-time stock level updates.
The IMS should provide a reliable notification system for reordering.

3. Purchase Orders

- **Functional Requirements:**

The IMS shall provide the manager/ employee with the ability to create and manage purchase orders and purchase order transactions.

- **Non-functional Requirements:**

The IMS should have efficient order management and tracking.

5. Supplier Management:

- **Functional Requirements:**

The IMS shall have the ability to track supplier information.
The IMS shall have the ability to record purchase history.

- **Non-functional Requirements:**

The IMS should ensure the secure storage of supplier data.

4. Sales Order Management:

- **Functional Requirements:**

The IMS shall provide the manager/employee with the ability to process customer sales orders.
The IMS shall have the ability to manage order fulfillment.

- **Non-functional Requirements:**

The IMS will have the ability of quick order processing to meet customer expectations.
Reliable order fulfillment and shipping procedures.

5. Point of Sale (POS):

- **Functional Requirements:**

The IMS shall handle transactions.
The IMS shall have the ability to generate sales receipts and customer receipts.

- **Non-functional Requirements:**

The IMS will ensure fast transaction processing at the POS.
The IMS should have a stable and user-friendly interface for cashiers.

6. Returns:

- **Functional Requirements:**

The IMS shall handle product returns and add products back to inventory.

- **Non-functional Requirements:**

The IMS should be able to process returns efficiently to ensure customer satisfaction.

7. Employee Management:

- **Functional Requirements:**

The IMS shall be able to manage employee information.
The IMS shall provide the manager the ability to manage employee shifts in dealing inventory

- **Non-functional Requirements:**

The IMS should have secure storage of employee data.
The IMS should be able to process payroll accurately and timely.

8. Reporting and Analytics:

- **Functional Requirements:**

The IMS shall generate various reports on sales orders and inventory.

- **Non-functional Requirements:**

The IMS should have the ability for efficient report generation and data analysis.
The IMS should ensure data accuracy and consistency for reporting.

9. Security and Access Control:

- **Functional Requirements:**

The IMS shall have the ability to implement role-based access control.

- **Non-functional Requirements:**

The IMS should secure user authentication and authorization.
The IMS should restrict access to sensitive data based on roles.

10. Customer Management:

- **Functional Requirements:**

The IMS shall have the ability to track customer purchase history.

- **Non-functional Requirements:**

The IMS should store customer data securely.
The IMS should ensure efficient retrieval of customer purchase history.

Entity Data Dictionary:

Entity Name	Description	Aliases
Product	Represent	prod
Category	Categorizes products on the basis of their usability and usage	catg
SalesOrder	Manages sales orders from customer	sOrder
OrderReturn	Records product returns	oReturn

Employee	Manages employee details and role.	emp
Customer	Manages customer names, IDs and details.	cust
purchaseOrder	Manages purchase orders and suppliers	prchsOdr
Stock	Manages product stock in the inventory	stock
Supplier	Manages supplier names, IDs and details.	supplier
Stock Report	Manages the reports of stocks, their generation dates and types.	sr

Attributes Data Dictionary:

Entity or Relationship Name	Attributes	Description	Data Type and Length	Nulls	Multi-valued
Product	prodID (PK)	Stores ID of each product.	N 6	No	No
	prodDscrp	Describes each product.	S 100	No	No
	prodPrice	Stores prices of each product	$ 12 (Money)	No	No
	prodName	Has a unique name for each product.	S 12	No	No
	catgID (FK)	Each product category has a separate ID.	N 4	No	No

	custID (FK)	ID of a specific customer.	N 5	No	No
	stockID	ID of a specific stock.	N 5		
Category	catgName	Each product category has a unique name.	S 12	No	No
	catgID(PK)	Each product category has a separate ID.	N 4	No	No
SalesOrder	orderID (PK)	Stores the ID of the sales order.	N 6	No	No
	custID (FK)	Stores the ID of the customer.	N 5	No	No
	tAmount	Total amount of sales made.	$ 12 (Money)	No	No
	orderDate	The date on which a particular product sales was ordered.	D (date)	No	No
	prodID (FK)	Stores ID of each product.	N 6	No	No
OrderReturn	returnReason	Reason for a product to be returned.	N 20	No	No
	returnID (PK)	ID of product returned.	N 6	No	No

	oReturnDate	Date on which the order was returned.	D (date)	No	No
	sOrderID (FK)	ID of sales order being returned.	N 6	No	No
Employee	empShift	Shift details of an employee.	D (datetime)	No	No
	employeeID (PK)	ID of each specific employee.	N 4	No	No
	empFName (composite)	First name of the employee.	S 10	No	No
	empLName (composite)	Last name of the employee.	S 10	No	No
	ManagerID	Name to the manager who himself belongs to the employee	int	No	No
	empEmail	Records email of a specific employee.	S 225	No	No
	empPassword	Records password of each employee to login the system.	S 225	No	No
Customer	custName	Unique name of a customer.	S 12	No	No

	custID (PK)	ID of a specific customer.	N 5	No	No
	custPhoneNo	Contact of a customer.	S 14	No	No
	custEmail	Every customer has an email address.	S 225	No	No
	custPassword	A password is unique for every customer.	S	No	No
PurchaseOrder	supplierID (FK)	ID of each specific supplier.	N 4	No	No
	prchsOdrID (PK)	ID of a specific purchase order.	N 5	No	No
	stockID (FK)	ID of a specific stock.	N 4	No	No
	prchsDate	Date on which a purchase was made.	D (date)	No	No
	pOPayMethod	Method of payment for a purchase.	S 225	No	No
Stock	stockID (PK)	ID of a stock.	N 5	No	No
	stockLevel	Level of stock (including the products available)	int	No	No
	stockThreshold L-evel	Minimum level of stock that should	int	No	No

		be maintained.			
	purchaseDate	Date on which a stock is purchased.	D (date)	No	No
Supplier	supplierPhone No	Contact of each supplier.	S 14	No	No
	supplierName	Name of a specific supplier.	S 12	No	No
	supplierID (PK)	ID of each specific supplier.	N 4	No	No

Relationship-Entity Data Dictionary:

Entity Name	Multiplicity	Relationship	Relationship Description	Multiplicity	Entity Name
Product	1..*	belongs	each product has some category	1..1	Category
Product	1..*	has	Product are sold in a salesorder	1..*	SalesOrder
Product	1..1	has	Products are available in a stock	1..*	Stock
Stock	1..1	has	Stocks are purchase from supplier in purchase order	1..*	PurchaseOrder
Stock	1..*	supplies	Stocks are supplied by suppliers	1..*	Supplier

Stock	1..1	has	Stock has a report for details.	1..1	StockReport
SalesOrder	1..1	has	Salesorder can be returned in if there is any reason	0..1	OrderReturn
Employee	1..1	process	Employee deals with salesOrder	1..*	SalesOrder
SalesOrder	1..*	Belongs to	Each Customer has salesorder	1..1	Customer

FlatTable (prodID, prodDscrp, prodPrice, stockLevel, prodName, catgID, catgName, returnID, returnReason, oReturnDate, , tPaymentMethod, employeeID, empShift, empFName, empLName, empEmail, empPassword StockID, stockLevel, stockThresholdLevel, supplierPhoneNo, supplierName, supplierID, purchaseDate, prchsOrdrID,pOPaymentMethod, custID, custName, custPhoneNo, custEmail, custPassword sOrderID, orderStatus, salesAmount, orderDate, sOPaymentMethod)

Normalization:

1st NORMAL FORM: (there is no multivalued attribute in our ERD.)

FlatTable (prodID, prodDscrp, prodPrice, stockLevel, prodName, catgID, catgName, returnID, returnReason, oReturnDate, , tPaymentMethod, employeeID, empShift, empFName, empLName, StockID, stockLevel, stockThresholdLevel, supplierPhoneNo, supplierName, supplierID, prchsOdrStatus, totalAmount, purchaseDate,

prchsOrdrID,pOPaymentMethod, custID, custName, custPhoneNo, sOrderID, orderStatus, salesAmount, orderDate, sOPaymentMethod)

2nd NORMAL FORM: (Each primary key has been assigned its dependant attributes and extracted from the flat table)

Product:

prodID	prodDescp	prodPrice	prodName	custID (FK)	stockID (FK)	catgID (FK)

Category:

catgName	catgID

SalesOrder

orderID	tAmount	orderDate	prodID(FK)	CustID(FK)

OrderReturn

returnID	returnReason	oReturnDate	sOrderID (FK)

Employee

employeeID	empShift	empFName	empLName	empEmail	empPassword	ManagerID

Customer

custID	custName	custEmail	custPassword	custPhoneNo

PurchaseOrder

prchsID	supplierID (FK)	pOPayMethod	prchsDate	stockID(FK)

Stock

stockID	stockLevel	stockThresholdLevel	prodID(FK)	purchaseDate

Supplier

supplierID	supplierName	supplierPhoneNo

3rd NORMAL FORM: (Transitive dependency has been removed)

Product:

prodID	prodDescp	prodPrice	prodName	catgID (FK)	custID (FK)	stockID (FK)

Category:

catgName	catgID

SalesOrder

sOrderID	tAmount	orderDate	prodID(FK)	CustID(FK)

SalesOrderhasProduct:

orderNo	sOrderID (FK)	prodID (FK)

OrderReturn

returnID	returnReason	oReturnDate	sOrderID (FK)

Employee

employeeID	empShift	empFName	empEmail	empPassword	empLName	ManagerID

Customer

custID	custEmail	custPassword	custName	custPhoneNo

PurchaseOrder

prchsID	supplierID (FK)	pOPayMethod	prchsDate	stockID(FK)

Stock

stockID	stockLevel	stockThresholdLevel	purchaseDate	prodID(FK)

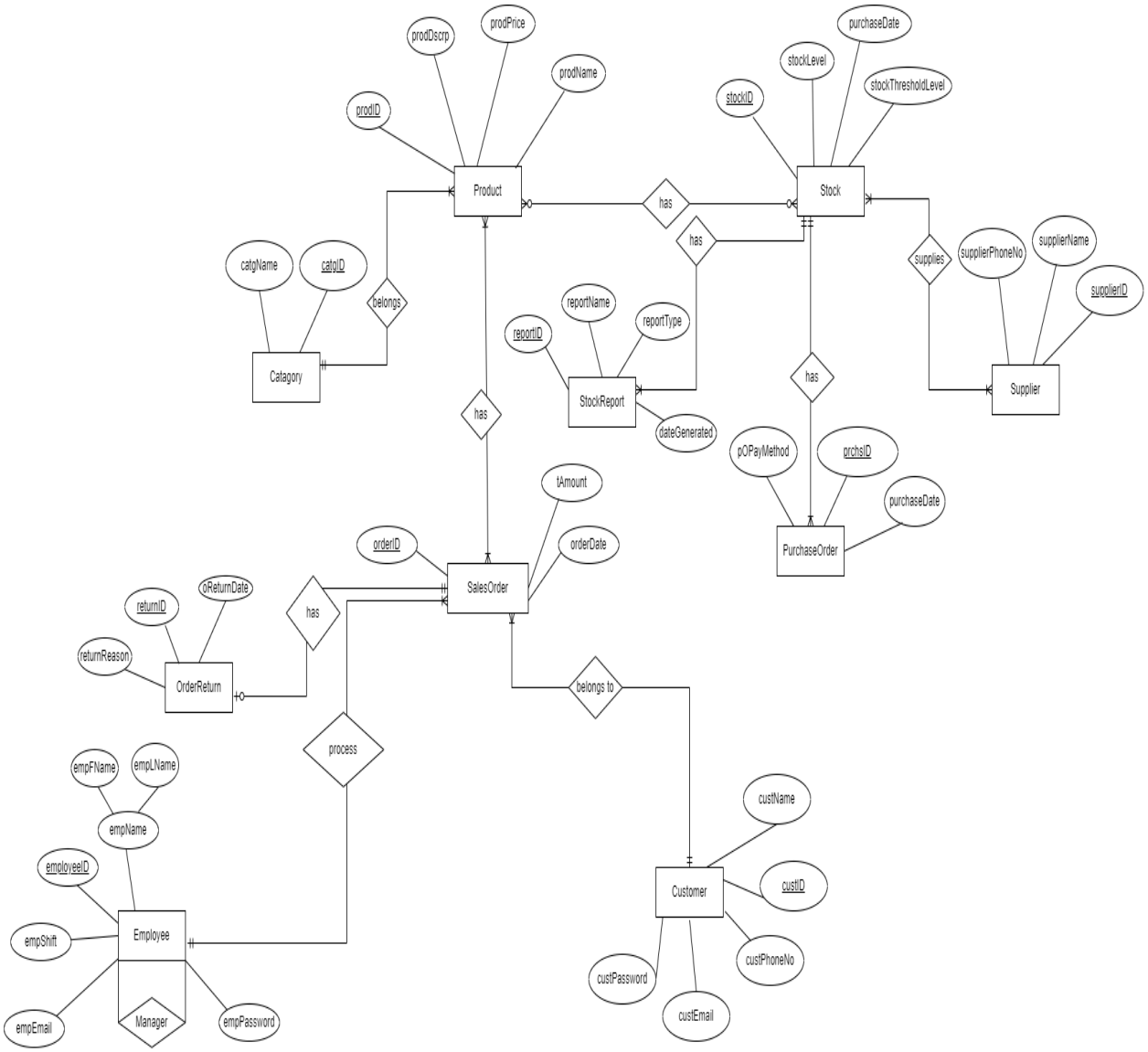
Supplier

supplierID	supplierName	supplierPhoneNo

SuppliersuppliesStock:

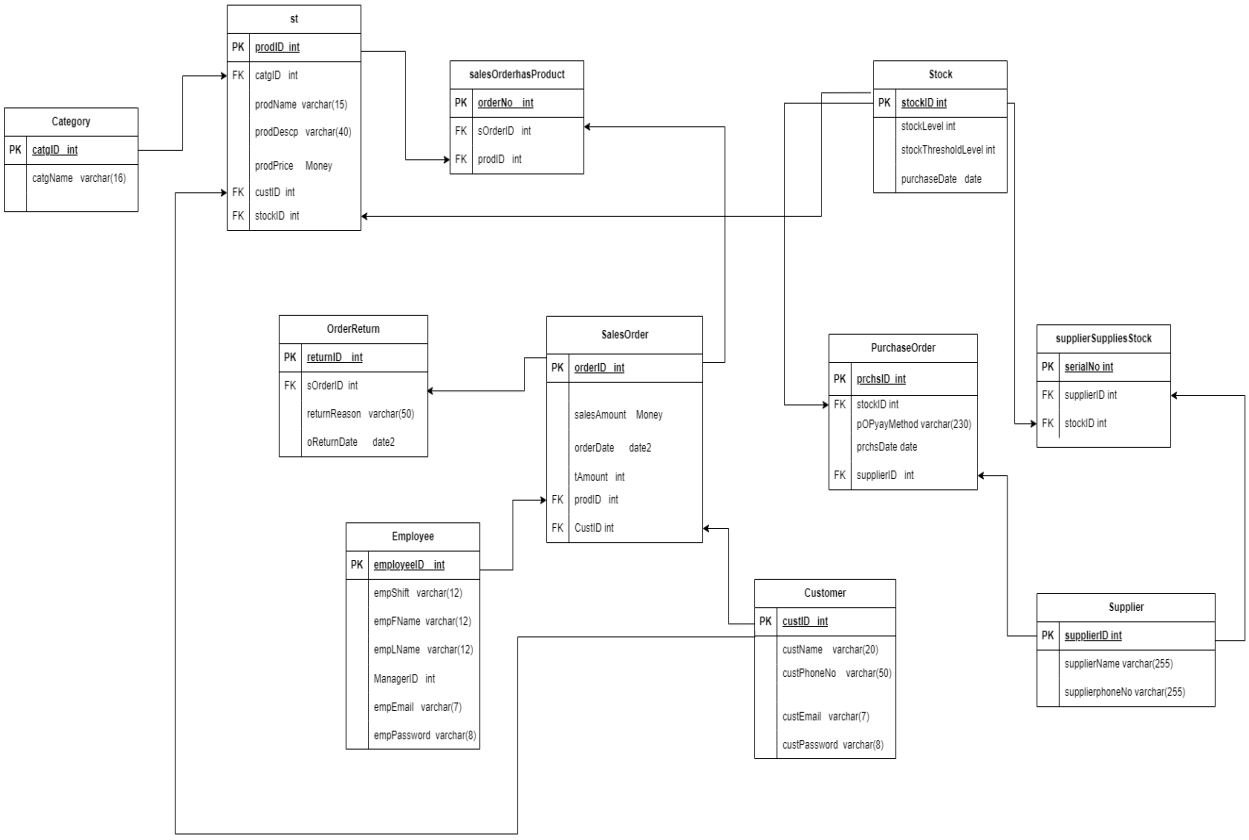
serialNo	supplierID (FK)	prodID (FK)

Entity Relationship Dlagram:



Normalized Entity Relationship Diagram:

Inventory Management System



Relational Schema:

Product:

prodID	prodDescp	prodPrice	prodName	catgID (FK)	custID (FK)	stockID (FK)

Category:

catgName	catgID

SalesOrder

orderID	orderDate	tAmount	custID (FK)	prodID (FK)

SalesOrderhasProduct:

orderNo	sOrderID (FK)	prodID (FK)

OrderReturn

returnID	returnReason	oReturnDate	sOrderID (FK)

Employee

employeeID	empShift	empFName	empLName	ManagerID	empEmail	empPassword

Customer

custID	custName	custPhoneNo	custEmail	custPassword

PurchaseOrder

prchsId	prchsDate	pOPayMethod	stockID (FK)	supplierID (FK)

Stock

stockID	stockLevel	thresholdLevel	purchaseDate

StockReport

reportID	reportName	dateGenerated	stockID (FK)

Supplier

supplierID	supplierName	supplierPhoneNo

SuppliersuppliesStock:

serialNo	supplierID (FK)	stockID (FK)

Database of InvenTrack (Inventory Management System)

```
CREATE DATABASE InvenTrack5;
USE InvenTrack5;

CREATE TABLE Employee (
  employeeID INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
  empShift VARCHAR(12),
  empFName VARCHAR(12) NOT NULL,
  empLName VARCHAR(12) NOT NULL,
  empEmail VARCHAR(255) NOT NULL,
  empPassword VARCHAR(255) NOT NULL,
  ManagerID INT
);

CREATE TABLE Customer (
  custID INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
  custName VARCHAR(20) NOT NULL,
  custPhoneNo VARCHAR(50) NOT NULL,
  custEmail VARCHAR(255) NOT NULL,
  custPassword VARCHAR(255) NOT NULL
);

CREATE TABLE Stock (
  stockID INT  IDENTITY(1,1) NOT NULL PRIMARY KEY,
  stockLevel INT,
  thresholdLevel INT,
  purchaseDate INT,
);

CREATE TABLE StockReport (
  reportID INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
  reportName VARCHAR(25),
  dateGenerated DATE,
  stockID INT,
  FOREIGN KEY (stockID) REFERENCES dbo.Stock(stockID) ON DELETE CASCADE ON
UPDATE CASCADE
);
```

```
CREATE TABLE Category (
  catgID INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
  catgName VARCHAR(16) NOT NULL
);

CREATE TABLE Product (
  prodID INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
  stockID INT,
  catgID INT,
  custID INT,
  prodName VARCHAR(15) NOT NULL,
  prodDescrp VARCHAR(40),
  prodPrice MONEY,
  FOREIGN KEY (catgID) REFERENCES Category(catgID) ON DELETE SET NULL ON
UPDATE CASCADE,
  FOREIGN KEY (stockID) REFERENCES Stock(stockID) ON DELETE SET NULL ON
UPDATE CASCADE
  FOREIGN KEY (custID) REFERENCES Customer(custID) ON DELETE SET NULL ON
UPDATE CASCADE,
);

CREATE TABLE Supplier (
  supplierID INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
  supplierName VARCHAR(225) NOT NULL,
  supplierphoneNo VARCHAR(255) NOT NULL
);

CREATE TABLE SupplierSuppliesStock (
  serialNo INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
  supplierID INT,
  stockID INT,
  FOREIGN KEY (supplierID) REFERENCES Supplier(supplierID) ON DELETE CASCADE ON
UPDATE CASCADE,
  FOREIGN KEY (stockID) REFERENCES Stock(stockID) ON DELETE CASCADE ON
UPDATE CASCADE
);

CREATE TABLE PurchaseOrder (
  prchsID INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
  stockID INT,
  supplierID INT,
  pOPayMethod VARCHAR(255),
  prchsDate DATE,
```

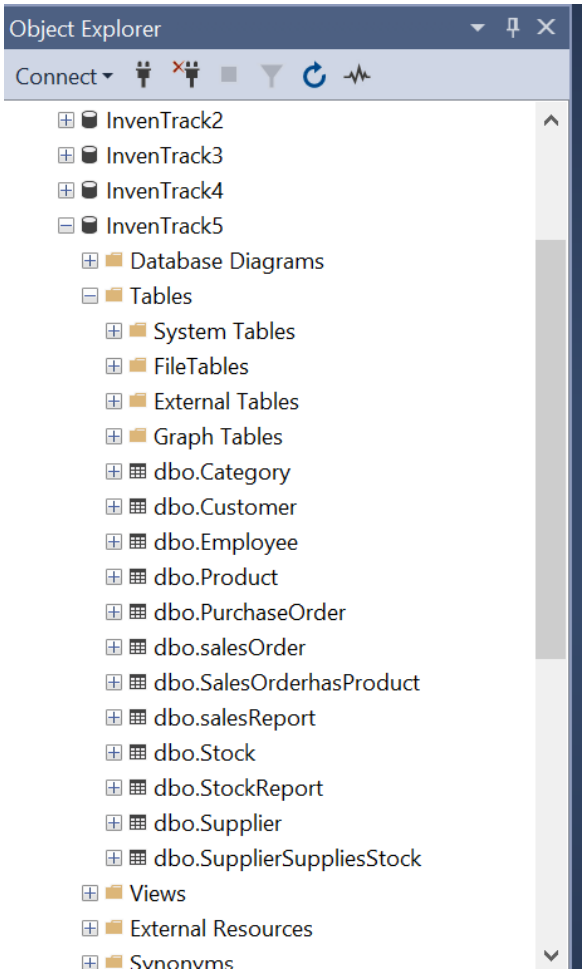
```
  FOREIGN KEY (supplierID) REFERENCES Supplier(supplierID) ON DELETE CASCADE ON
UPDATE CASCADE,
  FOREIGN KEY (stockID) REFERENCES Stock(stockID) ON DELETE SET NULL ON
UPDATE CASCADE
);
CREATE TABLE salesOrder(
orderID INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
custID INT,
prodID INT,
tAmount MONEY,
orderDate DATETIME DEFAULT GETDATE(),
FOREIGN KEY (custID) REFERENCES Customer(custID) ON DELETE NO ACTION ON
UPDATE NO ACTION,
FOREIGN KEY (prodID) REFERENCES Product(prodID) ON DELETE NO ACTION ON
UPDATE NO ACTION
);
```

Data For DataBase of InvenTrack

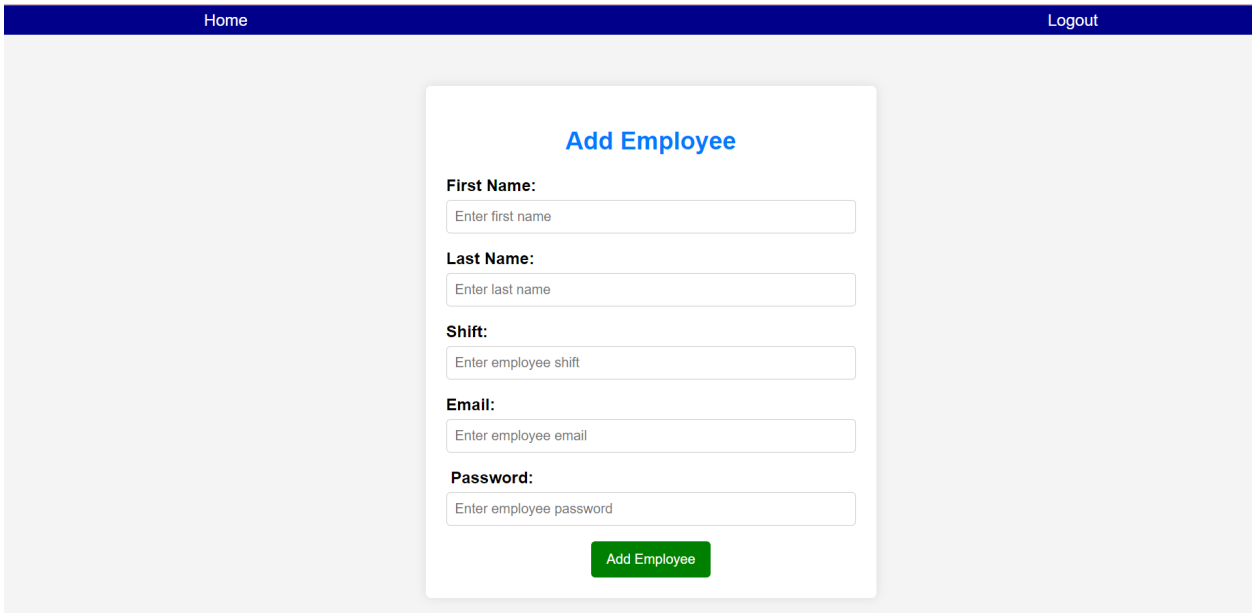
```
INSERT INTO Customer(custName, custPhoneNo, custEmail, custPassword)
VALUES ('Farhan Ali', '549-1-6233', 'farhan95@gmail.com', 'f398A'),
('Samia Junaid', '491-2-1432', 'junaid.samia@gmail.com', 'Samia28J'),
('Farman Haider', '592-1-3354', 'fHaider@gmail.com', 'FH456'),
('Amna Shahid', '312-3-1265', 'amna@gmail.com', 'amna24');

INSERT INTO Category(catgName)
VALUES ('Electronics');

INSERT INTO Product(stockID, catgID, prodName, prodDescrp, prodPrice, quantity)
VALUES (1, 1, 'Laptop', 'High-performance laptop', 1200.00, 1),
(1, 1, 'Mobile', 'High-performance mobile', 800.00, 4);

INSERT INTO StockReport (reportName, dateGenerated, stockID)
VALUES ('June Report', '2023-7-1', 1),
('August Report', '2023-9-1', 2);

INSERT INTO Supplier ( supplierName, supplierphoneNo) VALUES ( 'ABC Supplier',
'987-654-3210');
INSERT INTO Supplier ( supplierName, supplierphoneNo) VALUES ( 'DEF Supplier',
'123-456-3210');
INSERT INTO Supplier ( supplierName, supplierphoneNo) VALUES ( 'GHI Supplier',
'012-654-3210');
```

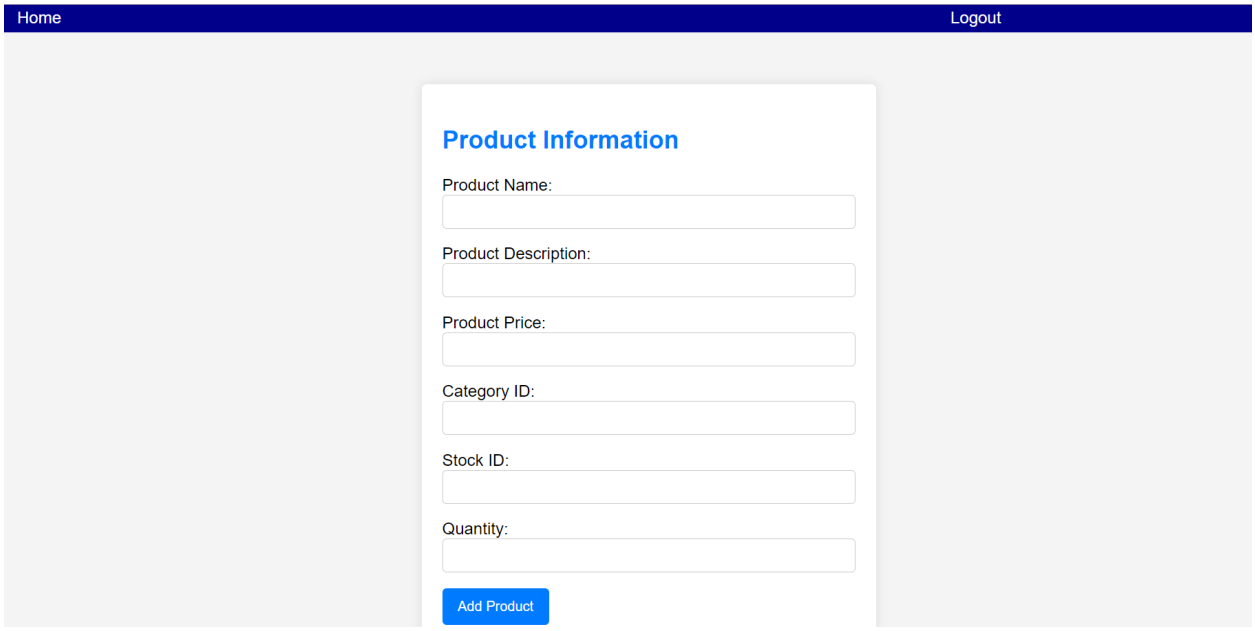
INSERT INTO Employee (empFName, empLName, empEmail, empPassword, ManagerID) VALUES ('Jane', 'Smith', 'jane@gmail.com','jane123',NULL);

INSERT INTO SupplierSuppliesStock (serialNo, supplierID, prodID) VALUES (1, 1, 1);



Frontend Pages of InvenTracks:


Home
Logout
Add Employee
First Name:
Enter first name
Last Name:
Enter last name
Shift:
Enter employee shift
Email:
Enter employee email
Password:
Enter employee password
Add Employee


Home
Logout
Product Information
Product Name:
Product Description:
Product Price:
Category ID:
Stock ID:
Quantity:
Add Product

Home Logout

Customer Record

Customer ID	Customer Name	Product ID	Product Category	Product Description	Product Price	Product Stock Level
1	Farhan ALi	2	Electronics	High-performance laptop	1200.0000	16
2	Tom Shaw	3	Electronics	High-performance mobile	800.0000	16

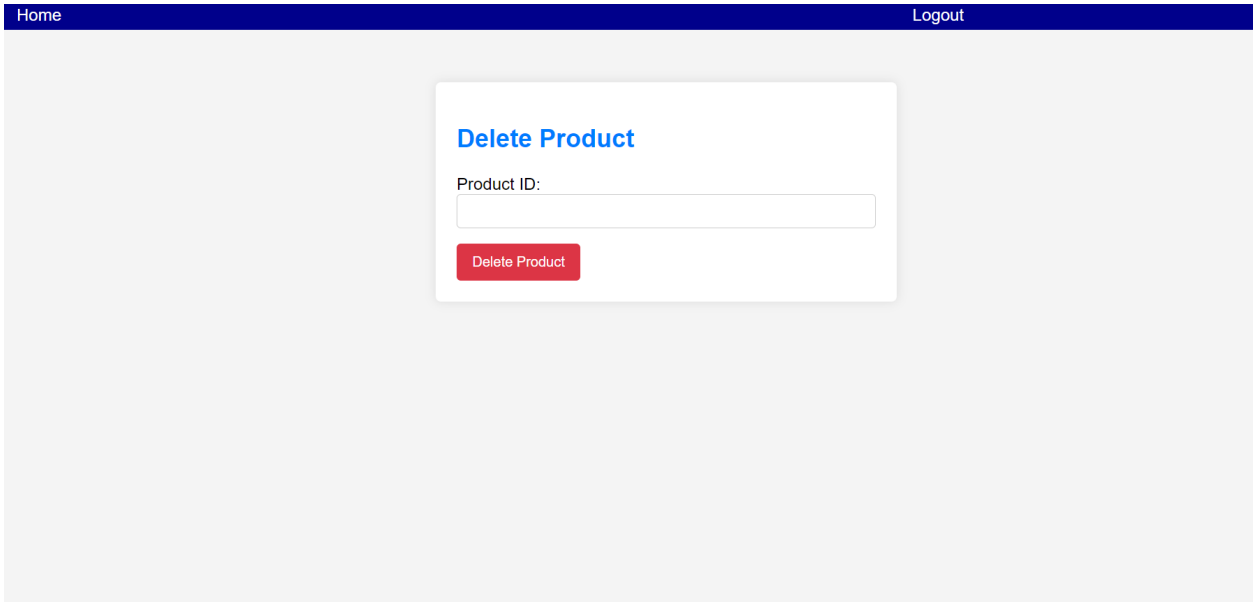
Home Logout

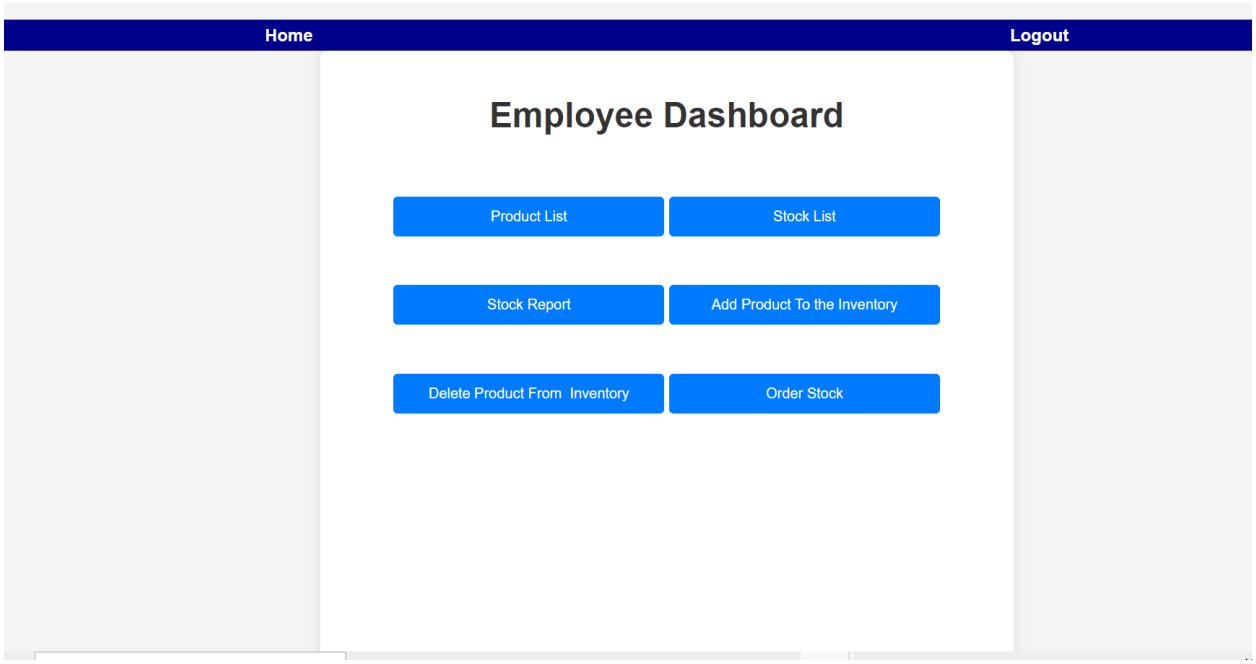
Delete Employee

Employee ID:

Enter employee ID

Delete Employee


Home
Logout
Delete Product
Product ID:
Delete Product


Home
Logout
Employee Dashboard
Product List
Stock List
Stock Report
Add Product To the Inventory
Delete Product From Inventory
Order Stock

Home Logout

Employee Record

Employee ID	Shift	First Name	Last Name	Email	Password	Manager ID
1		Jane	Smith	jane@gmail.com	jane123	
2	day	ali	durrani	ali@gmail.com	ali123	1

Login

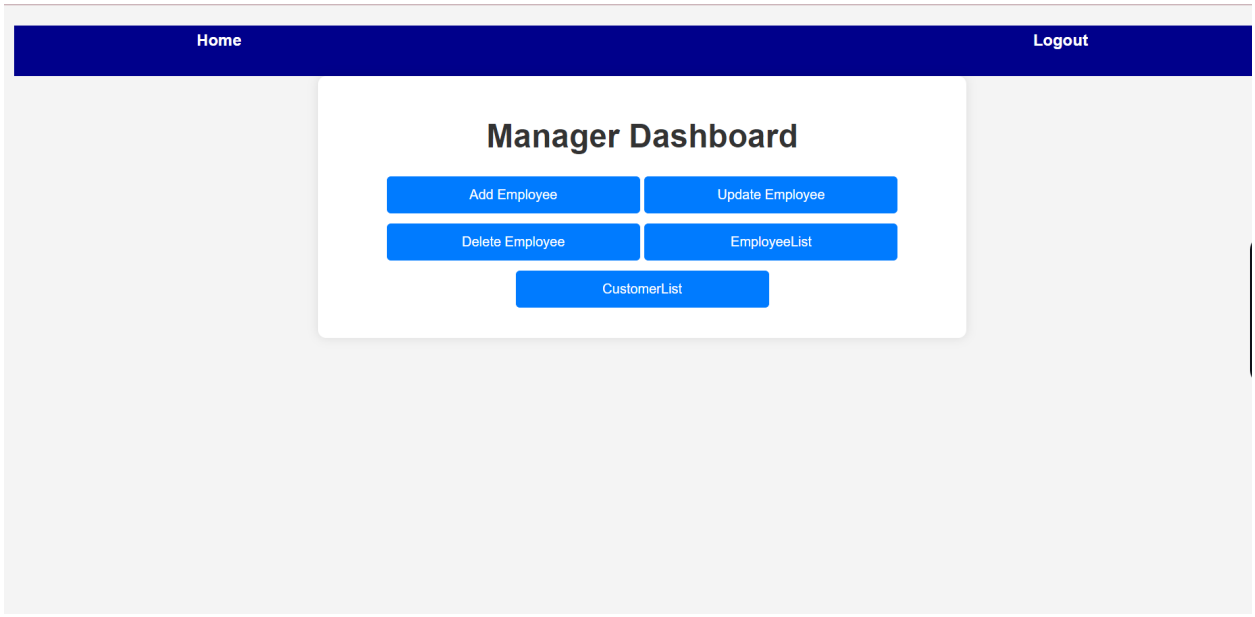
Email:

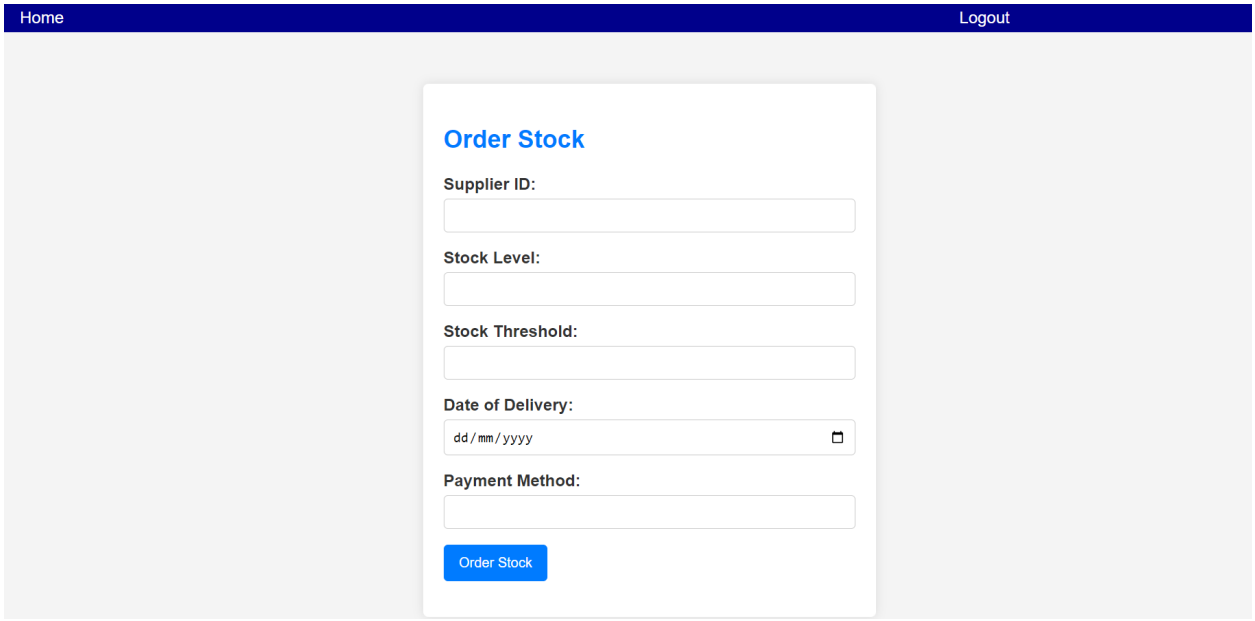
Enter your email

Password:

Enter your password

Login


Home
Logout
Manager Dashboard
Add Employee
Update Employee
Delete Employee
EmployeeList
CustomerList


Home
Logout
Order Stock
Supplier ID:
Stock Level:
Stock Threshold:
Date of Delivery:
dd/mm/yyyy
Payment Method:
Order Stock

Home Logout

Product List

Product ID	Stock ID	Category ID	Product Name	Product Description	Product Price	Product Quantity
2	1	1	Laptop	High-performance laptop	1200.0000	
3	1	1	mobile	High-performance mobile	800.0000	4
4	1	1	curler	remingtonbrand	4300.0000	10

Home Logout

Stock List

Stock ID	Stock Level	Threshold Level	Purchase Date
1	16	10	
2	20	10	
3	0	5	03/12/2023 12:00:00 am
4	0	3	01/12/2023 12:00:00 am
5	0	6	09/12/2023 12:00:00 am

Home Logout

Stock Report

Stock ID	Date Generated	Products Sold	Product Price
1	01/07/2023 12:00:00 am	Laptop	1200.0000
1	01/07/2023 12:00:00 am	mobile	800.0000
1	01/07/2023 12:00:00 am	curler	4300.0000

Home Logout

Update Employee

Employee ID:

Enter employee ID

First Name:

Enter first name

Last Name:

Enter last name

Shift:

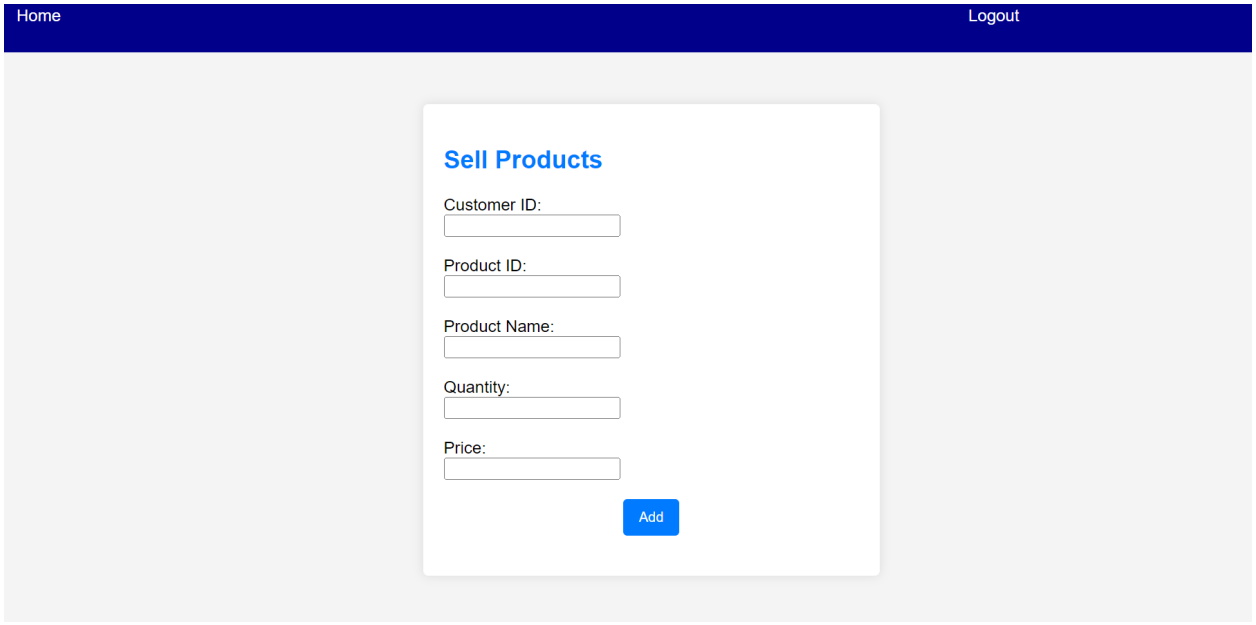
Enter employee shift

Email:

Enter employee Email

Password:

Enter employee password


Home
Logout
Sell Products
Customer ID:
Product ID:
Product Name:
Quantity:
Price:
Add

--------THE END--------