Final Report

Jiho Kim, Garth Slaney, Eddie Kim

CPSC 471

Submission Date: December 9, 2020

University of Calgary

Abstract:

Covid-19 has greatly impacted the way consumers shop, as they look for ways to minimize the number of trips outside of the household. This has led to a demand for digitalization of store's inventories as consumers want to minimize their exposure to other people. However, this poses unique challenges for small businesses as they have been struggling to adapt to the increasing demand for consumers wanting to access their inventories with minimal outside exposure. Based on a real-life grocery store the objective of this project was to create an online inventory system for a small grocery store that digitalizes their data and automates their transactions. This will allow the grocery store to manage their inventory easily and it will also provide accurate information to the clients and suppliers.

The aim of the solution is to achieve a better inventory management system that can be accessed and modified anywhere. This will save money and time as manual inventory tracking will no longer be required. Furthermore, this will allow consumers and store owners to have more accurate information regarding the products. The database saves virtually and is accessible from the website. The website allows consumers to view products and make product purchases. In addition, the website has an admin login feature that allows the store owner to view financial records in addition to making and tracking purchases made from suppliers.

Design Section:

Our system has two different users, a store owner and a customer. The store owner is able to view financial records and make supplier purchases. To view the financial records the store owner logs in to the system. The system will then call a POST API to check whether the user has entered incorrect credentials. The store owner then enters the month and year of the records he wishes to see. The system then calls a GET API using the user input to retrieve the money made from customer purchases that month, the monthly cost of running the store (rent, electricity, etc.) and the money spent that month on supplier purchases. The store owner is provided with a catalogue from the suppliers which can then be used by the store owner to make a purchase from the supplier. The supplier catalogue will be consistently updated by the suppliers. All purchases made from the supplier are logged under supplier purchase.
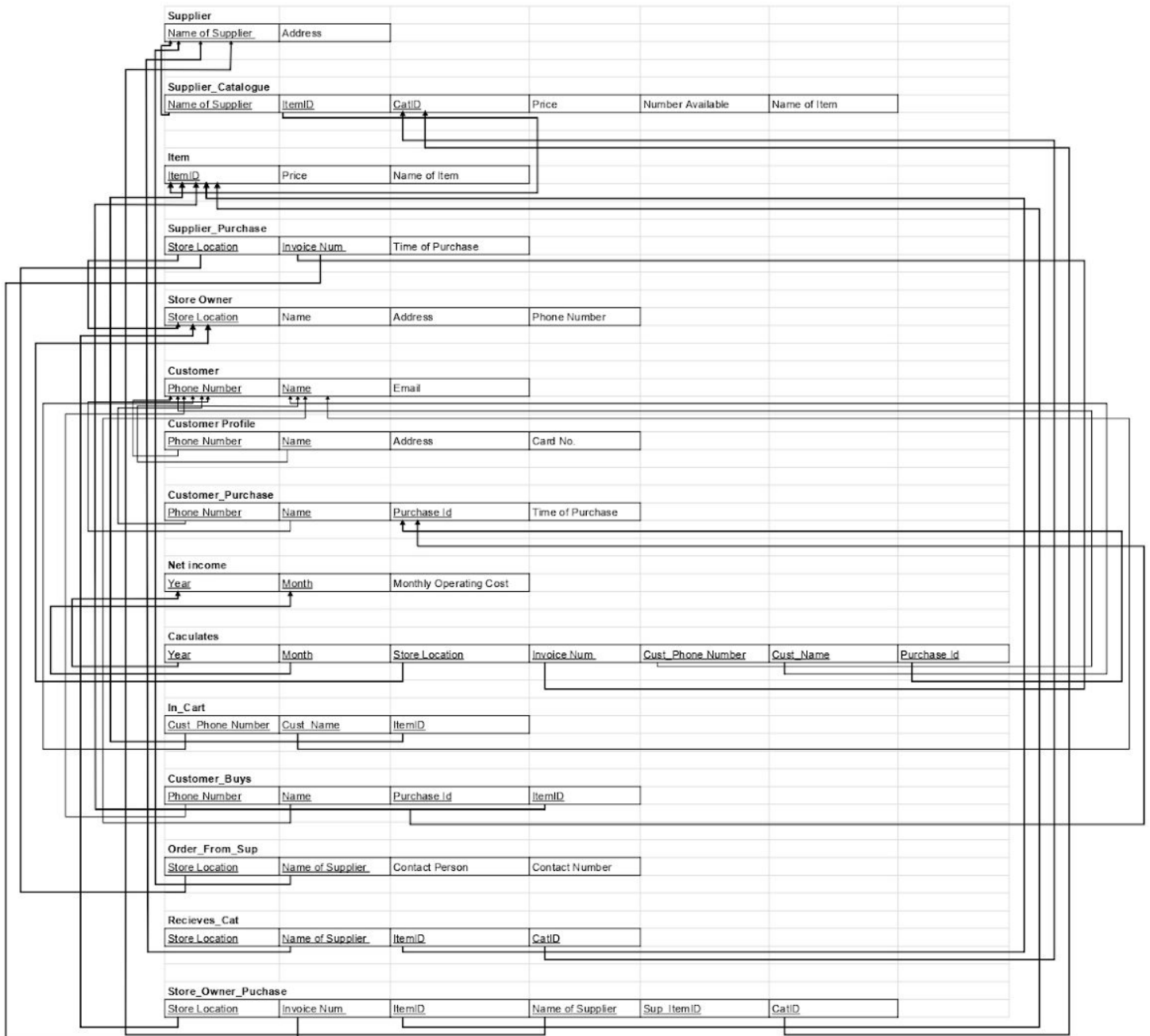
Customers are able to login/register into the system (using a POST API) by entering their phone number and name. The list of items are retrieved using a GET API and customers will then be able to add items to their cart. When the customer adds all their desired items to the cart, they will then be able to make a purchase of those items. Customers are also able to register by providing their phone number, name and email. Once registered, the customers are then provided with the option to store their credit card information and their address for ease of purchase.

Grocery Store Inventory System ER Diagram
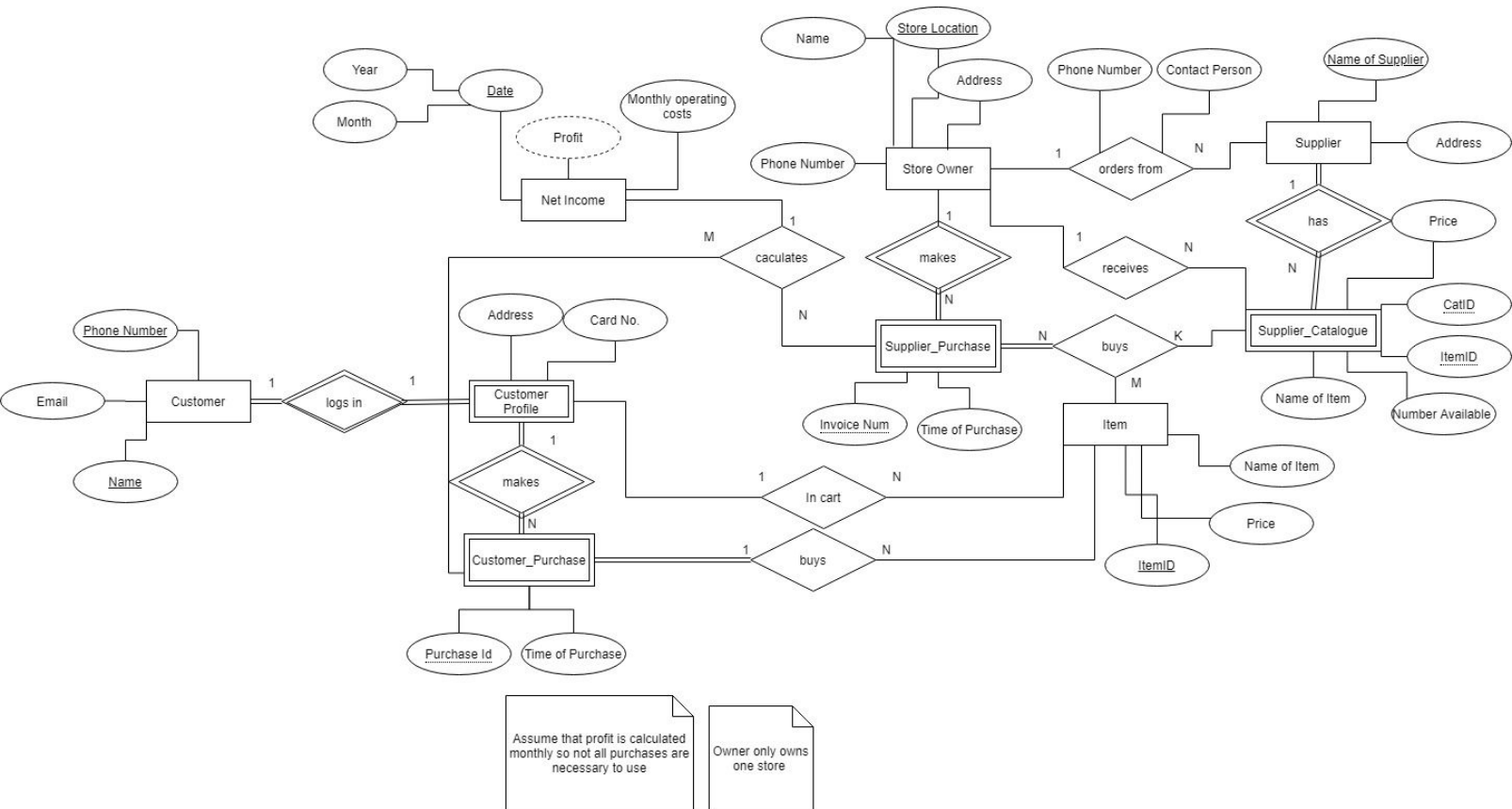
- The only change made in implementation is the removal of the calculates

  relationship as we instead used stored procures to generate total income.

Implementation Section:

Supplier

Name of Supplier	Address

Supplier_Catalogue

Name of Supplier	ItemID	CatID	Price	Number Available	Name of Item

Item

ItemID	Price	Name of Item

Supplier_Purchase

Store Location	Invoice Num	Time of Purchase

Store Owner

Store Location	Name	Address	Phone Number

Customer

Phone Number	Name	Email

Customer Profile

Phone Number	Name	Address	Card No.

Customer_Purchase

Phone Number	Name	Purchase Id	Time of Purchase

Net income

Year	Month	Monthly Operating Cost

Caculates

Year	Month	Store Location	Invoice Num	Cust_Phone Number	Cust_Name	Purchase Id

In_Cart

Cust_Phone Number	Cust_Name	ItemID

Customer_Buys

Phone Number	Name	Purchase Id	ItemID

Order_From_Sup

Store Location	Name of Supplier	Contact Person	Contact Number

Recieves_Cat

Store Location	Name of Supplier	ItemID	CatID

Store_Owner_Puchase

Store Location	Invoice Num	ItemID	Name of Supplier	Sup_ItemID	CatID

- The only change made was removing calculates and instead having stored procedures in the database find and calculate net income.

We selected the WebHost database integration and more specifically phpMyAdmin running 10.3.16-MariaDB.

SQL Transactions:

```
CREATE DEFINER=`id15457855_cpscfinal`@`%` PROCEDURE `AddCustomer`(IN `I_Phone_Number` INT, IN `I_Name` VARCHAR(255), IN `I_Email` VARCHAR(255)) NOT DETERMINISTIC CONTAINS SQL SQL SECURITY DEFINER

BEGIN

    INSERT INTO Customer VALUES (I_Phone_Number, I_Name, I_Email);

END

CREATE DEFINER=`id15457855_cpscfinal`@`%` PROCEDURE `AddCustomerProfile`(IN `I_Phone_Number` INT, IN `I_CName` VARCHAR(255), IN `I_Address` VARCHAR(255), IN `I_Card_Num` INT) NOT DETERMINISTIC MODIFIES SQL DATA SQL SECURITY DEFINER

BEGIN

    INSERT INTO Customer_Profile VALUES (I_Phone_Number, I_CName, I_Address, I_Card_Num);

END

CREATE DEFINER=`id15457855_cpscfinal`@`%` PROCEDURE `AddCustomerPurchase`(IN `I_Phone_Number` INT, IN `I_Name` VARCHAR(255), IN `I_PurchaseId` INT) NOT DETERMINISTIC MODIFIES SQL DATA SQL SECURITY DEFINER

BEGIN

    INSERT INTO Customer_Purchase(Phone_Number, Name, PurchaseId) VALUES (I_Phone_Number, I_Name, I_PurchaseId);

END
```

```
CREATE DEFINER=`id15457855_cpscfinal`@`%` PROCEDURE `AddItem`(IN `I_Id`
INT, IN `I_Price` INT, IN `I_NameItem` VARCHAR(255)) NOT DETERMINISTIC
MODIFIES SQL DATA SQL SECURITY DEFINER

BEGIN

        INSERT INTO Item VALUES (I_Id, I_Price, I_NameItem);

END

CREATE DEFINER=`id15457855_cpscfinal`@`%` PROCEDURE `AddNetIncome`(IN
`I_Years` INT, IN `I_Months` INT, IN `I_Monthly_Cost` INT) NOT DETERMINISTIC
MODIFIES SQL DATA SQL SECURITY DEFINER

BEGIN

        INSERT INTO NetIncome VALUES (I_Years,I_Months, I_Monthly_Cost);

END

CREATE DEFINER=`id15457855_cpscfinal`@`%` PROCEDURE `AddStoreOwner`(IN
`I_Store_Location` VARCHAR(255), IN `I_Name` VARCHAR(255), IN `I_Address`
VARCHAR(255), IN `I_Phone_Number` VARCHAR(255)) NOT DETERMINISTIC
MODIFIES SQL DATA SQL SECURITY DEFINER

Begin

        Insert Into Store_Owner VALUES(I_Store_Location, I_Name, I_Address,
        I_Phone_Number);

End

CREATE DEFINER=`id15457855_cpscfinal`@`%` PROCEDURE `AddSupplier`(IN
`I_NameSup` VARCHAR(255), IN `I_Address` VARCHAR(255)) NOT DETERMINISTIC
MODIFIES SQL DATA SQL SECURITY DEFINER

BEGIN

        INSERT INTO Supplier VALUES(I_NameSup, I_Address);

END

CREATE DEFINER=`id15457855_cpscfinal`@`%` PROCEDURE
`AddSupplierCatalogue`(IN `I_NameSup` VARCHAR(255), IN `I_ItemID` INT, IN
`I_CatID` INT, IN `I_Price` INT, IN `I_NumAv` INT, IN `I_NameItem` VARCHAR(255))
NOT DETERMINISTIC MODIFIES SQL DATA SQL SECURITY DEFINER

BEGIN
```

```sql
        INSERT INTO Supplier_Catalogue VALUES(I_NameSup, I_ItemID, I_CatID,
        I_Price, I_NumAv,I_NameItem);

END

CREATE DEFINER=`id15457855_cpscfinal`@`%` PROCEDURE
`AddSupplierPurchase`(IN `I_Store_Location` VARCHAR(255), IN `I_InvoiceNum` INT,
IN `I_TimePurchase` DATE) NOT DETERMINISTIC MODIFIES SQL DATA SQL
SECURITY DEFINER

BEGIN

        INSERT INTO Supplier_Purchase VALUES (I_Store_Location, I_InvoiceNum,
        I_TimePurchase);

END

CREATE DEFINER=`id15457855_cpscfinal`@`%` PROCEDURE `AddToCart`(IN
`I_Cust_Phone` INT, IN `I_Cust_Name` VARCHAR(255), IN `I_ItemID` INT) NOT
DETERMINISTIC MODIFIES SQL DATA SQL SECURITY DEFINER

BEGIN

        INSERT INTO In_Cart VALUES (I_Cust_Phone, I_Cust_Name, I_ItemID);

END

CREATE DEFINER=`id15457855_cpscfinal`@`%` PROCEDURE
`CaculateMoneyEarned`(IN `Years` INT, IN `Months` INT) NOT DETERMINISTIC
READS SQL DATA SQL SECURITY DEFINER

BEGIN

SELECT SUM(I.Price)
        From I as Item, Cp as Customer_Purchase, Cb as Customer_Buys
        Where Cp.PurchaseId = Cb.PurchaseId and Cb.ItemId = I.ItemID and
Years=YEAR(Cp.TimeofPurchase) and Months = MONTH(Cp.TimeofPurchase) ;

 END

CREATE DEFINER=`id15457855_cpscfinal`@`%` PROCEDURE
`CaculateMoneySpent`(IN `Years` INT, IN `Months` INT) NOT DETERMINISTIC
READS SQL DATA SQL SECURITY DEFINER

BEGIN

SELECT SUM(I.Price)
        From I as Item, Sp as Supplier_Purchase, Stor as Store_Owner_Purchase
```

Where Sp.InvoiceNum = Stor.InvoiceNum and Stor.ItemId = I.ItemID and Years=YEAR(Sp.Time_of_Purchase) and Months = MONTH(Sp.Time_of_Purchase) ;

END

CREATE DEFINER=`id15457855_cpscfinal`@`%` PROCEDURE `GetCartItems`(IN `phone` INT, IN `name` VARCHAR(255)) NOT DETERMINISTIC CONTAINS SQL SQL SECURITY DEFINER

BEGIN

Select ItemID From In_Cart Where Cust_Phone_Number = phone and Cust_Name = name;

END

CREATE DEFINER=`id15457855_cpscfinal`@`%` PROCEDURE `GetCustomerProfiles`() NOT DETERMINISTIC CONTAINS SQL SQL SECURITY DEFINER

BEGIN

Select * from Customer_Profile;

END

CREATE DEFINER=`id15457855_cpscfinal`@`%` PROCEDURE `GetCustomerPurchases`() NOT DETERMINISTIC READS SQL DATA SQL SECURITY DEFINER

BEGIN

Select * from Customer_Purchase;

END

CREATE DEFINER=`id15457855_cpscfinal`@`%` PROCEDURE `GetCustomers`() NOT DETERMINISTIC READS SQL DATA SQL SECURITY DEFINER

BEGIN

Select * from Customer;

END

CREATE DEFINER=`id15457855_cpscfinal`@`%` PROCEDURE `GetItems`() NOT DETERMINISTIC READS SQL DATA SQL SECURITY DEFINER

BEGIN

```sql
        Select * from Item;

END

CREATE DEFINER=`id15457855_cpscfinal`@`%` PROCEDURE `GetMonthlyCost`(IN `Years` INT, IN `Months` INT) NOT DETERMINISTIC NO SQL SQL SECURITY DEFINER

BEGIN

SELECT NI.Monthly_Operating_Cost
        FROM NI as NetIncome
                WHERE NI.Year = Years and NI.Month = Months;

END

CREATE DEFINER=`id15457855_cpscfinal`@`%` PROCEDURE `GetNetIncomes`() NOT DETERMINISTIC READS SQL DATA SQL SECURITY DEFINER

BEGIN

        Select * from NetIncome;

END

CREATE DEFINER=`id15457855_cpscfinal`@`%` PROCEDURE `GetStoreOwner`() NOT DETERMINISTIC READS SQL DATA SQL SECURITY DEFINER

BEGIN

        Select * from Store_Owner;

END

CREATE DEFINER=`id15457855_cpscfinal`@`%` PROCEDURE `GetSupplierCatalogues`() NOT DETERMINISTIC READS SQL DATA SQL SECURITY DEFINER

BEGIN

        Select * from Supplier_Catalogue;

END

CREATE DEFINER=`id15457855_cpscfinal`@`%` PROCEDURE `GetSupplierPurchases`() NOT DETERMINISTIC READS SQL DATA SQL SECURITY DEFINER

BEGIN
```

```sql
        Select * from Supplier_Purchase;

END

CREATE DEFINER=`id15457855_cpscfinal`@`%` PROCEDURE `GetSuppliers`() NOT
DETERMINISTIC READS SQL DATA SQL SECURITY DEFINER

BEGIN

        Select * from Supplier;

END

CREATE DEFINER=`id15457855_cpscfinal`@`%` PROCEDURE
`MakeCustomerPurchase`(IN `Cust_N` VARCHAR(255), IN `Cust_P` INT, IN
`PurchaseId` INT) NOT DETERMINISTIC MODIFIES SQL DATA SQL SECURITY
DEFINER

BEGIN

 -- Keep track of last and first rows
DECLARE lastRows INT DEFAULT 0;
DECLARE startRows INT DEFAULT 0;

-- Find how many items in cart
SELECT COUNT(*) FROM In_Cart INTO lastRows;
SET startRows=0;

-- Make customerpurchase
Call AddCustomerPurchase(Cust_P, Cust_N, PurchaseId);

WHILE startRows < lastRows DO
        IF (Cust_N = (SELECT Cust_Name FROM In_Cart LIMIT startRows ,1) and
Cust_P =(SELECT Cust_Phone_Number FROM In_Cart LIMIT startRows ,1))
                THEN INSERT INTO Customer_Buys VALUES (Cust_P, Cust_N,
PurchaseId,(SELECT ItemId FROM In_Cart LIMIT startRows ,1));
        END IF;
        SET startRows= startRows+1;
END WHILE;


CREATE DEFINER=`id15457855_cpscfinal`@`%` PROCEDURE
`RemoveFromCart`(IN `Cust_P` INT, IN `Cust_N` VARCHAR(255)) NOT
DETERMINISTIC NO SQL SQL SECURITY DEFINER

BEGIN
```
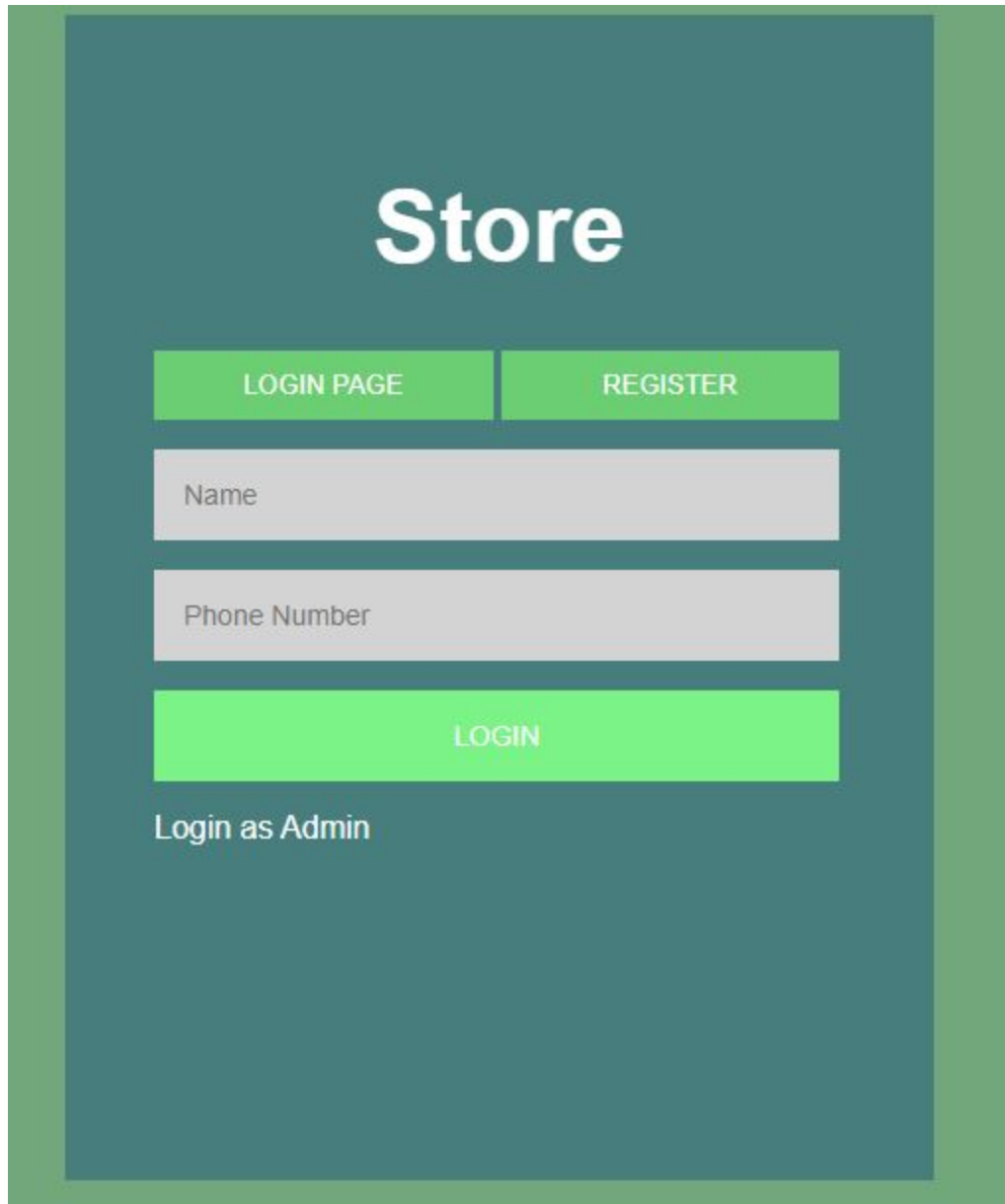
```
    DELETE FROM In_Cart WHERE Cust_Phone_Number = Cust_P and
    Cust_Name = Cust_N;

END
```

Website: (link at https://cpscfinal.000webhostapp.com/index.php)

Main Page:
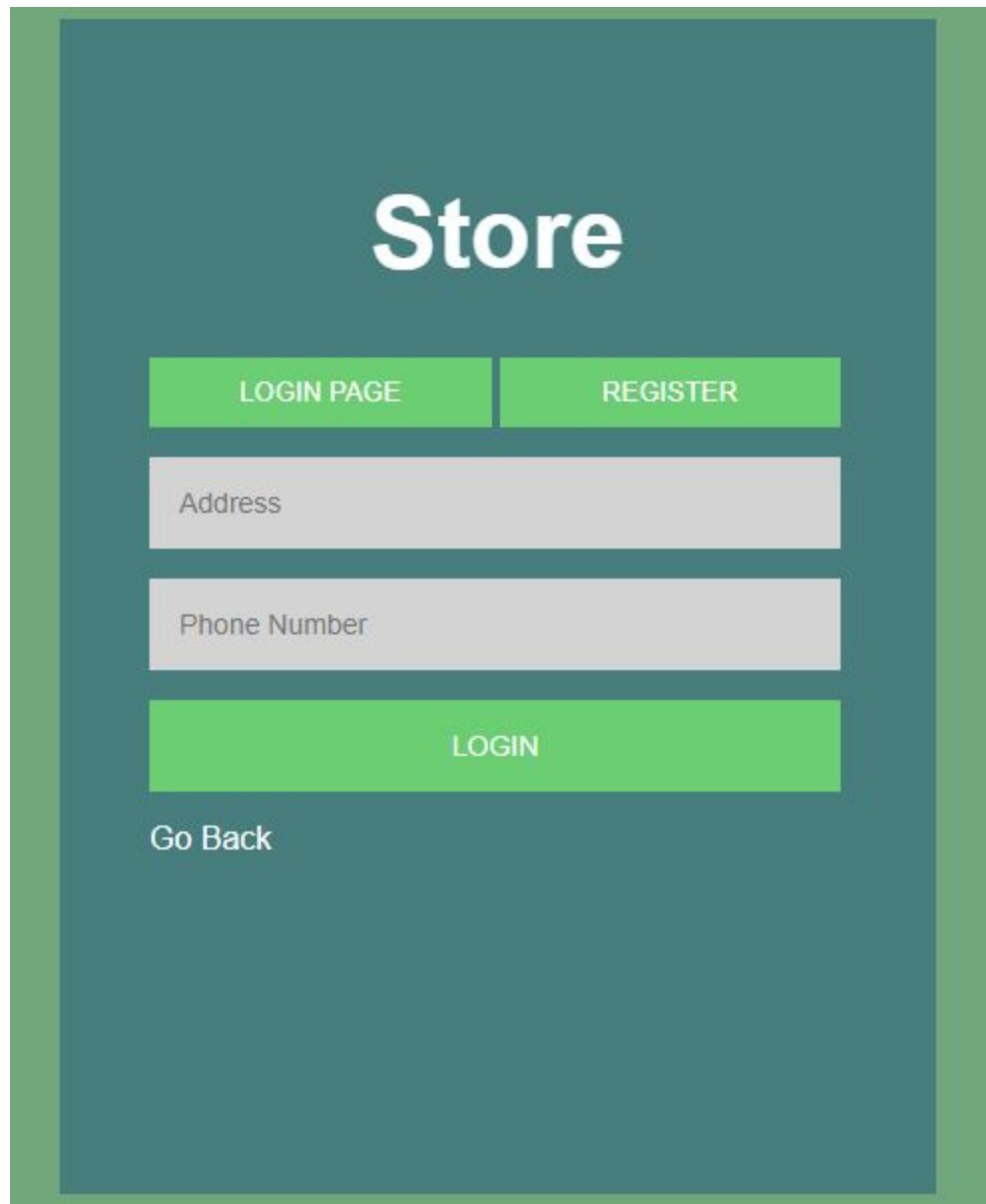
Admin Login:

Store

| LOGIN PAGE | REGISTER |

Address

Phone Number

LOGIN

Go Back

Financial Report:

Report:

2020 Report

money earned in 2020-12 is: $90
money spent in 2020-12 is: $20

LOG OUT

Register Customer:

Item Selection:

Store

→ PAY

MILK	$20
SOY MILK	$20
ALMOND MILK	$15
WATER	$15
CHIPS	$11
PLUM PUDDING	$11
DEAD SHEEP	$20

Select Items:

Store

Item	Price
MILK	$20
SOY MILK	$20
ALMOND MILK	$15
WATER	$15
CHIPS	$11
PLUM PUDDING	$11
DEAD SHEEP	$20
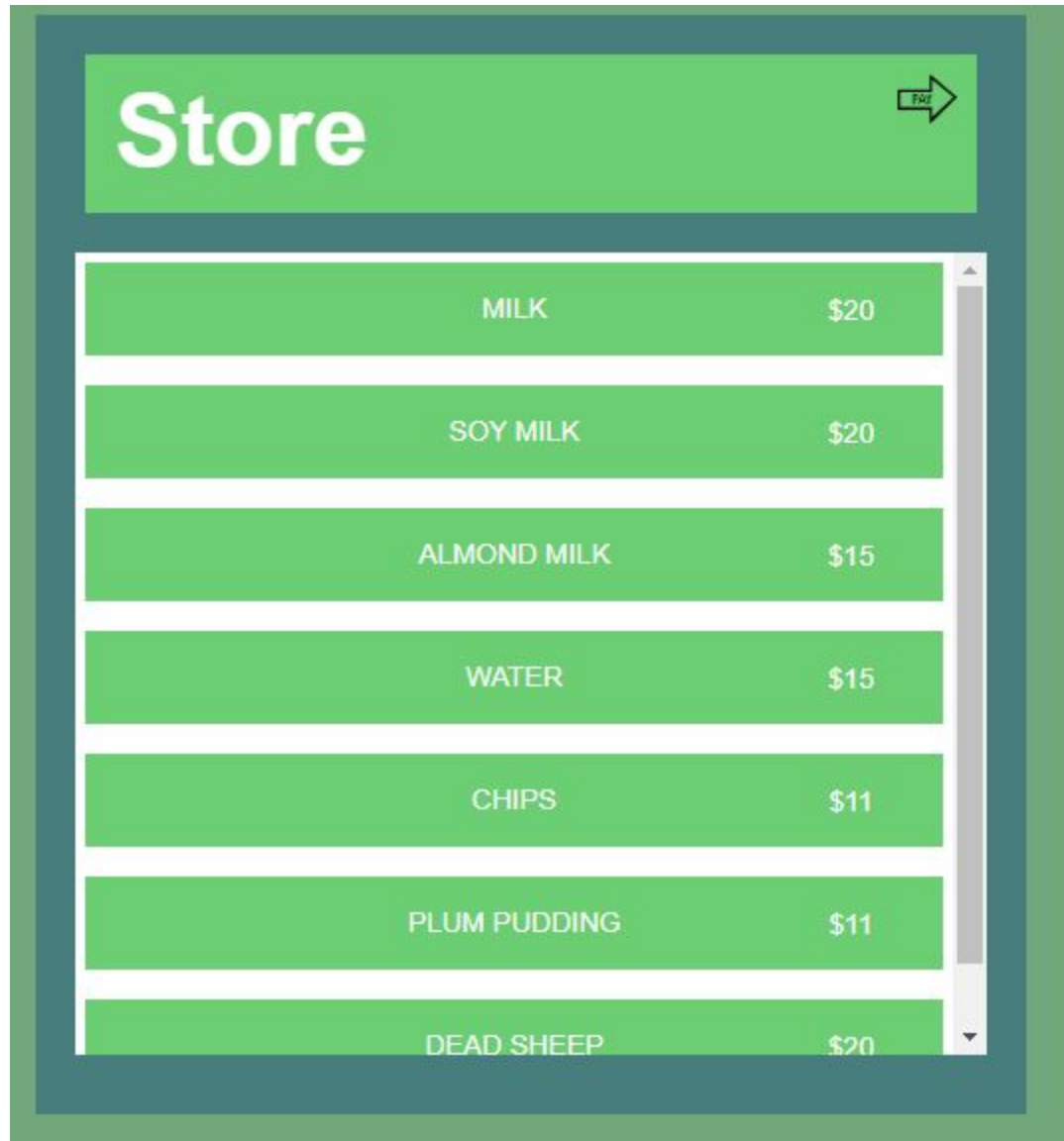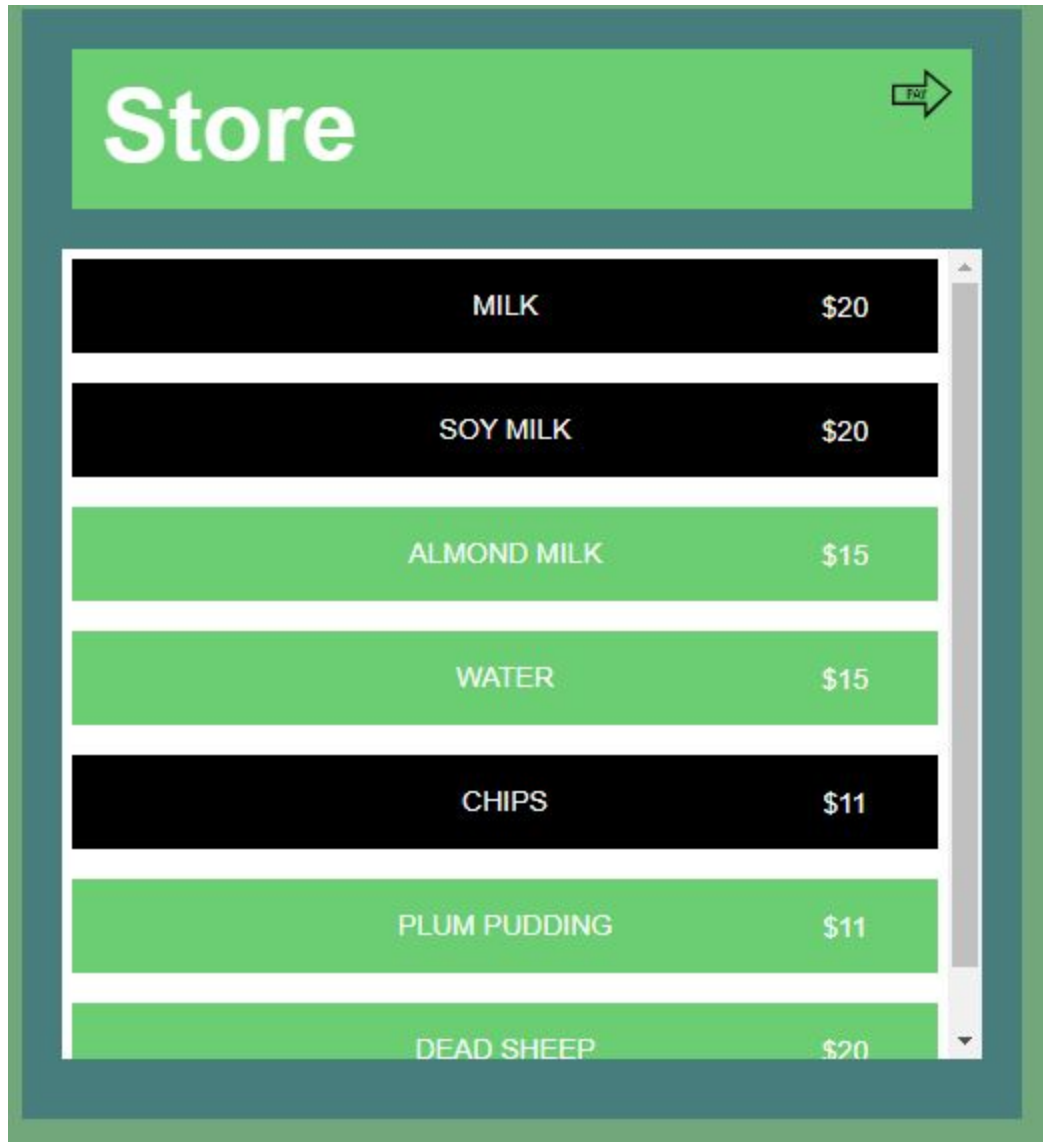
Purchase Confirmed:
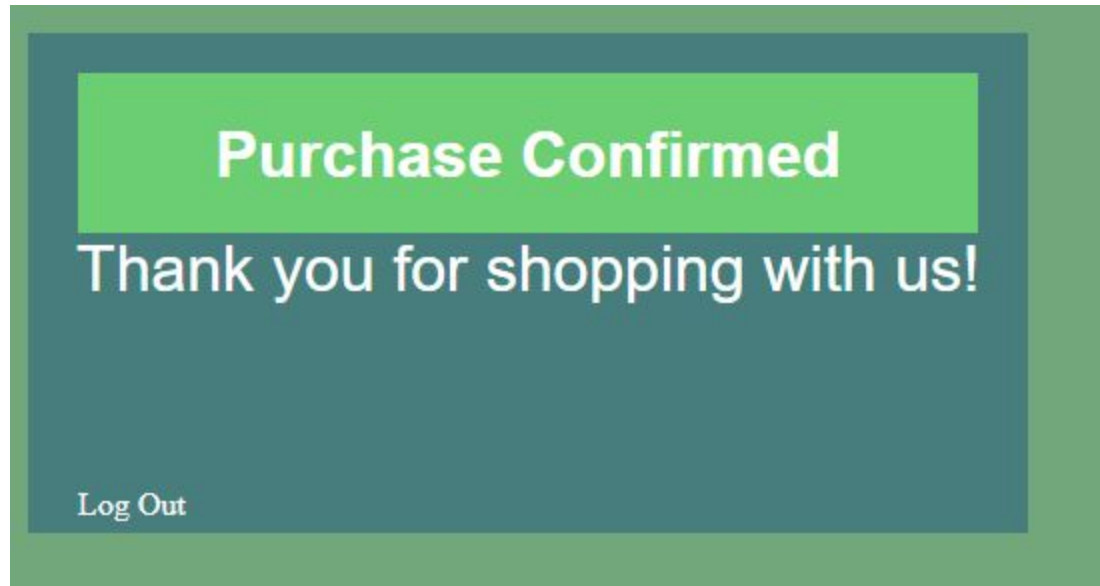
API Documentation

The documentation can be found at

https://documenter.getpostman.com/view/13702197/TVmTbuVY

Quick User Guide:

Currently, our website has the following features implemented; logging in as a customer, logging in as admin, viewing financial records as admin and registering a customer. We also have the visual implementation for adding items to a cart and making a purchase with those items after logging in as a customer. In the future, our website will also include the front end design for store owners to gather information from suppliers and make purchases from suppliers by retrieving/modifying information from the database. Our API's can be called using the Postman tool with the endpoints shown in our Postman documentation.