Retail Store Management System

an Online Retail Store

Built as a course project for
CSE202: Fundamentals of Database Management Systems

Harsh Vishwakarma (2022205)
Kartik (2022238)
Lakshya Dayma (2022268)

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY **DELHI**

Deadline 1

Defining Project Scope and Requirements

PROJECT OVERVIEW

We are aiming to make an online grocery management system. The online store will be a B2C ecommerce platform that allows customers to browse and purchase products online with various discounts. Different payment methods will be linked. In terms of data, we will be storing customer information like name, phone number, email and address. Data regarding products will be stored like descriptions, price, reviews and ratings. The key stakeholders are the business owners, customers, suppliers, and delivery partners.

PROJECT SCOPE

The grocery management system aims to facilitate :

- Effective acquiring of groceries, distribution of groceries by supplier.
- Effective helpline contact for any query.
- Different payment methods will be linked for payment.
- Delivery partners of delivery of different products.
- Easy browsing and purchase of product by the customers.

CUSTOMER ORDER MANAGEMENT

- Register for the app and login (Name, contact details, delivery address, Email)
- Browse and view the available products.
- Add product to the cart one by one and display the total sum.
- Select the item, quantity, delivery date(default today) for orders
- 2 click order(ONLINE)
- New deals must be shown for special occasions.
- Multiple payment processes (linked bank account(optional)).
- Track the order to show the current status of the delivery.
- Customer care contact.
- There must be a feedback option for the delivered orders.

CONSTRAINTS

A customer(with specific email-id or mobile no.) can have only one account in the store application.

Multiple customers can purchase the same products, and conversely, one customer can acquire multiple product.

There can be multiple sellers for the same product,this would be a many-to-many relationship.

Single cart per account and one account can have many payment methods stored in the account.

The order should be placed only after payment is confirmed. The cart should be emptied once when order is placed.

Deadline 2

Conceptual and Relational Models

The link to the Pdf is attached here :
“https://drive.google.com/file/d/1jg0UU71Vl8JmY1rlEvZDeIqT-5pDNwYG/view?usp=sharing”

Deadline 3

Database schema and indexes creation with integrity constraints and data insertion

The required sql script is attached in the folder.

-

Deadline 4

Embedded SQL Queries and Triggers

Embedded SQL Queries

Select all customers who have placed orders:

```
SELECT DISTINCT c.*
FROM Customer c
JOIN Orders o ON c.CustomerID = o.CustomerID;
```

Update the quantity of a product in stock:

```
UPDATE products
SET quantity_in_stock = quantity_in_stock - 5
WHERE product_id = 3;
```

Insert a new deal with a discount:

```
INSERT INTO Deals (DealCode, Discount, Descriptions)
VALUES ('DEAL11', 15.00, '15% discount on electronics');
```

Delete a product and its associated deals:

```
DELETE FROM Cart WHERE ProductID = 7;
DELETE FROM ItemsDeals WHERE ProductID = 7;
```

Select orders along with customer information and delivery status:

```
SELECT o.OrderID, c.CustomerName, o.OrderDate, o.DeliveryDate, d.Name AS
DeliveryPartner,
CASE WHEN o.CurrentStatus = 1 THEN 'Pending'
WHEN o.CurrentStatus = 2 THEN 'Delivered'
ELSE 'Unknown' END AS Status
FROM Orders o
```

```
JOIN Customer c ON o.CustomerID = c.CustomerID
LEFT JOIN DeliveryPartner d ON o.DeliveryPartnerID = d.DeliverypartnerID;
```

Calculate total revenue generated from orders:

```
SELECT SUM(TotalSum) AS TotalRevenue
FROM DeliveryInfo;
```

Select products along with their suppliers:

```
SELECT i.itemName AS Product, s.SupplierName AS Supplier
FROM Items i
JOIN ItemsSuppliers isup ON i.ProductID = isup.ProductID
JOIN Supplier s ON isup.SupplierID = s.SupplierID;
```

Select orders with payment information:

```
SELECT o.OrderID, p.Amount, p.PaymentMethod, p.payment_date
FROM Orders o
JOIN Payment p ON o.OrderID = p.OrderID;
```

Update delivery instructions for a specific order:

```
UPDATE Cart
SET DeliveryInstruction = 'Leave at the doorstep'
WHERE CustomerID = 4 AND ProductID = 4;
```

Select deals applied to each product:

```
SELECT i.itemName AS Product, d.DealCode, d.Discount
FROM Items i
LEFT JOIN ItemsDeals id ON i.ProductID = id.ProductID
LEFT JOIN Deals d ON id.DealCode = d.DealCode;
```

Deadline 5

Embedding SQL in Java and Creation of Triggers

Triggers

The following triggers are identified according to the use cases and implemented in the database:

Trigger 1:

```
DELIMITER //
CREATE TRIGGER update_quantity_in_stock
AFTER INSERT ON Orders
FOR EACH ROW
BEGIN
   DECLARE ordered_quantity INT;
   -- Get the ordered quantity from the Orders table
   SELECT Quantity INTO ordered_quantity
   FROM Orders
   WHERE OrderID = NEW.OrderID;
   -- Update the quantity in stock in the Items table
   UPDATE Items
   SET QUANTITY = QUANTITY - ordered_quantity
   WHERE ProductID IN (
      SELECT ProductID
      FROM Cart
      WHERE CustomerID = NEW.CustomerID
   );
END //
DELIMITER ;
```

Trigger 2:

```
DELIMITER //
CREATE TRIGGER Account_Block_after_3_attempts AFTER INSERT ON loginattempts
FOR EACH ROW
BEGIN
   DECLARE attempts INT;
   DECLARE blocked BOOLEAN;
   -- Get the current number of attempts and block status
   SELECT Attempts, Blocked INTO attempts, blocked
   FROM LoginAttempts
   WHERE CustomerID = NEW.CustomerID;
   -- Update the number of attempts and block status
   SET attempts = attempts + 1;
   IF attempts >= 3 THEN
      SET blocked = TRUE;
   END IF;
   -- Update the LoginAttempts table
   UPDATE LoginAttempts
   SET Attempts = attempts, Blocked = blocked, LastAttempt = CURRENT_TIMESTAMP
   WHERE CustomerID = NEW.CustomerID;
END //
DELIMITER ;
```

Deadline 6

Conflicting and Non-Conflicting Database Transactions

The link for conflicting and non- conflicting transactions :
"https://docs.google.com/document/d/1HycPmrzqYd-iunNI999UZ0COd-k7YbaXm04EO2E2U4Q/edit?usp=sharing"

User Guide

Welcome to the Retail Store Management System Command-Line Interface (CLI).
This guide will help you navigate through the various functionalities of the system.

Getting Started:

Launch the CLI
Main Menu:

Upon launching the CLI, you will be presented with a main menu.
Select an option by entering the corresponding number and pressing Enter.

Main Menu Options:

Enter as Admin:

Choose this option to log in as an admin user.
You will be prompted to enter your username and password.

User Login:

Select this option to log in as a regular user.
Enter your username and password when prompted.
After three consecutive failed login attempts, your account will be temporarily blocked.
Once logged in, you can access features available to regular users.

User Register:

Use this option to register as a new user if you don't have an existing account.
Follow the instructions to complete the registration process.

Exit:
Choose this option to exit the CLI.

Incorrect Credentials:

If you enter incorrect credentials, you will be prompted to try again.
After a 3 failed attempts, your account may be temporarily blocked for security reasons.

User Registration:

Choose this option from the main menu to register as a new user.
You will be prompted to enter the following details:

- **Customer Name**
- **House/Street Number**
- **City**
- **State**
- **Pincode**
- **Email ID**
- **Phone Number**
- **Password**

Submitting Registration:
After entering all required details, the system will automatically generate a unique Customer ID based on the total number of existing customers.
Once all details are entered, the user information will be stored in the database.
If the registration is successful, you will receive a confirmation message indicating successful registration.
If there's any issue during registration, an error message will be displayed, and you may need to retry.
Return to Main Menu.

After completing the registration process, you will automatically return to the main menu.
From here, you can explore other features of the CLI or exit the program.

Data Validation:

- The system may perform basic validation on the entered data to ensure correctness and prevent errors during registration.
- For example, the system check for valid email formats, phone number formats, or ensure that required fields are not left empty.

- Password Security: The system may implement security measures for passwords, such as enforcing a minimum length or requiring a combination of letters, numbers, and special characters.

OrderItemsApp

Welcome Message:
Upon launching the application, users are greeted with a welcome message indicating they have entered the store.
Main Menu:
Users are presented with a menu displaying options to:

- **View Items**
- **View Deals**
- **Exit the application**
- **View Items:**

When selecting the "View Items" option:
The application fetches and displays all available items from the database, including their names, quantities, and prices.
Users are then prompted to choose from additional options:

- **Add items to the cart**
- **Go back to the main menu**
- **View Deals:**

Choosing the "View Deals" option displays all available deals.
Deals are fetched from the database and presented along with their respective details, such as deal code, discount percentage, description, and associated item name.

Add Item To Cart

It presents a menu with options:

- **Add items**
- **Remove items**
- **Display cart**
- **Place order**

Based on the user's choice, it calls corresponding methods.
addItem
Users can add items to the cart by specifying the item name and quantity.
It retrieves item details from the database based on the provided name.
If the item is found and the requested quantity is available, it adds the item to the cart.

It calculates and displays the updated total amount after adding the item.

displayCart
Displays the items currently in the cart along with their quantities and total amounts.
If the cart is empty, it notifies the user.

calculateTotalAmount
Calculates the total amount of the items in the cart based on their prices and quantities.

removeItem
Allows users to remove items from the cart.
It displays the items in the cart and prompts the user to enter the name and quantity of the item they want to remove.
If the item is found and the requested quantity is valid, it updates the cart accordingly.
It notifies the user of the success or failure of the removal process.

placeOrder
This method facilitates placing an order for the items in the cart.
It calculates the total amount of the order and prompts the user to select a payment method.
Based on the chosen payment method, it either processes the payment online or confirms cash on delivery.
After payment confirmation, it retrieves a random delivery partner from the database and informs the user about the delivery partner.
Finally, it exits the program.
If a delivery partner is found, it returns the name of the delivery partner.

payOnline
This method simulates online payment by prompting the user to enter the total amount to be paid.
If the amount entered by the user matches the total amount of the order, it confirms successful payment and places the order.
If the amount is insufficient, it informs the user that the order cannot be placed due to insufficient payment.

updateItemQuantities
This method updates the quantities of items in the database after an order is successfully placed.

ADMIN

- **viewAllCustomers**

This method retrieves and displays all customers' details from the Customer table in the database.

The customer details are displayed in a formatted manner.

CUSTOMER ID:|| NAME:|| CITY:||STATE:||PINCODE:||EMAIL:||PHONE NUMBER:

- **Top 5 Customers with the Most Orders**
  Lists the top 5 customers who have placed the most orders.
- **Last Order by Customer by Date**
  Displays the last order date for each customer.
- **Customers with the Highest Total Purchase Amount**
  Lists the top 5 customers with the highest total purchase amount.
- **Customers with the Highest Average Order Amount**
  Shows the top 5 customers with the highest average order amount.
- **Top 5 Customers with the Most Items Purchased**
  Lists the top 5 customers who have purchased the most items.
- **Customers who have Availed the Most Deals**
  Displays the top 5 customers who have availed the most deals.
- **Exit to main menu**

Inventory Management Functionalities:

- **Add New Item**

Allows adding a new item to the inventory with details such as name, quantity, price, description, reviews, and ratings.

- **Update Item Details**

Enables updating the details of an existing item in the inventory, including quantity, price, description, reviews, and ratings.

- **Remove Item**

Allows removing an item from the inventory based on its Product ID.

These functionalities empower the admin to effectively manage the inventory by adding new items, updating existing item details, and removing items as needed, ensuring accurate and up-to-date inventory records.