Project Name

GROCERY MANAGEMENT

TEAM MEMBERS:

J.DEVISREE CHANDANA(AI23BTECH11008)

K.JAGADEESH(AI23BTECH11012)

S.RAMA HARSHINI(AI23BTECH11028)

S.VENKATA SAILAJA(AI23BTECH11029)

Overview

This project where various customers can order their groceries online from the seller they wish to and compare and choose them of their interest .It is also useful for sellers to sell their grocery items to their customers.

Goals

1.Easy to order the items and also check for various possible options in customer point of view

2.Seller can sell their product at any time irrespective of market ups and downs.

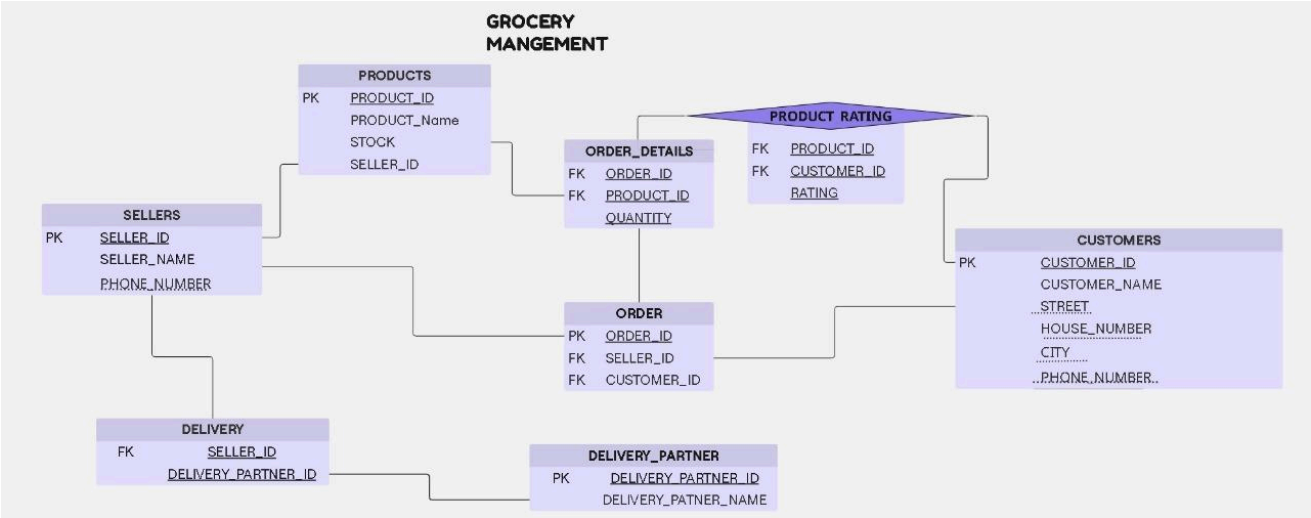
Specifications

In this project we use

- css and html for the frontend
- mysql for the database
- django for the backend.

Database Design:

ER Diagram:



Functional Dependencies and Normalisation:

- All the functional dependencies in schema are Trivial .
- They are in BCNF.

Schema:

Tables:

1.customers:

customer_id - int -PRIMARY KEY

customer_name - varchar(50)

2.Customer_ph_no:

customer_id - int

phone_number - varchar(20)

3.customer_addresses:

customer_id - int

street - varchar(15)

house_no - varchar(15)

city - varchar(50)

4.sellers:

seller_id - int - PRIMARY KEY

seller_name - varchar(40)

5.seller_ph_no:

seller_id - int - PRIMARY KEY

phone_number - varchar(20)- PRIMARY KEY

6.products:

product_id -int -PRIMARY KEY

product_name - varchar(40)

stock -float

seller_id - int

7.product_rating:

product_id- int

customer_id-int

rating -float

8.orders:

order_id -int -PRIMARYKEY

Customer_id -int

seller_id -int

9. order_details:

order_id - int

product_id - int

quantity_ordered - float

10.delivery:

Seller_id- int - PRIMARY KEY

delivery_partner_id -int -PRIMARY KEY

delivery_partner_name -varchar(50)

11.delivery_Partner:

delivery_partner_id -int-PRIMARY KEY

delivery_partner_name-VARCHAR(250)

Stored Procedures:

1.new_cust:

In this whenever a new customer is logged in we create a newId from him and save his details into database.

2.place_order:

We will create a order in database with help of customer_id,product_id and quantity

3.new_seller:

In this whenever a new seller is logged in we create a newId from him and save his details into the database.

4.new_product:

In this whenever a new product is added a newId is created and saved into the database.G

Triggers:

We will use 3 triggers here

1.after_order_insert:

To check if there is sufficient quantity before placing the order.

2.before_order_insert:

To update the quantity in the database after placing the order.

3.before_product_rating:

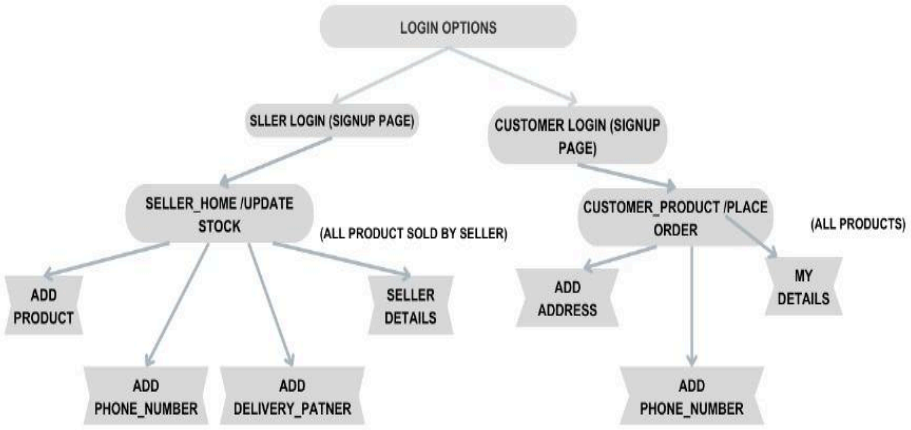
It checks whether a order for a product is placed before allowing the customer to give a product rating.

Indexing:

Index - seller_id_idx on the table products for the column seller_id, we have added this, because we would query on table products, with key seller_id, as each time the seller logins, for efficiency.

Index- product_id_idx on table product_rating for the column product_id, we have added this because each time a customer logs in, a join operation between products and product_rating for average rating of each product.

Functionality:



THANK YOU