E-COMMERCE DATABASE MANAGEMENT SYSTEM

Problem Statement:

1. Lack of proper market platforms for vendors. Also, with the increase in cost of living, finding the right location to set up a store is either difficult or expensive to afford.

2. With covid restrictions and busy life style, customers require a platform to make purchases in a more fast and efficient way.

Solution:

Nations are developing faster and so as the technology is also developing. The advanced Smartphones, Computer Systems, Web Access speed, etc.. has made it easy for the users to execute their task in a short interval of time and scope of ecommerce achieve high growth in the future. E-commerce shopping made the online shopping experience safe and secure with an additional layer of security.

E-Commerce Database Management System (EC-DBMS) is a system that is designed to store, process, retrieve and analyze information concerned with the administrative and management of sales activity done by a customer online sitting at home. The project also aims in providing a stable online platform to connect with the customers and also provide better services based on the reviews and ratings provided by the user.

TABLES CREATED

Table: address

Columns:

addressid	int(11) AI PK
houseno	int(11)
street	varchar(50)
custid	int(11)

Table: cart

Columns:

cartid	int(11) AI PK
dtcreated	date
custid	int(11)

Related Tables:

Target	customers (custid → custid)
On Update	RESTRICT
On Delete	RESTRICT

Table: cartproduct

Columns:

cartproductid	int(11) AI PK
vendorproductid	int(11)
quantity	int(11)
cartid	int(11)

Table: category

Columns:

categoryid	int(11) AI PK
categoryname	varchar(80)

Table: city

Columns:

cityid	int(11) AI PK
cityname	varchar(50)

Table: courier

Columns:

courierid	int(11) AI PK
couriername	varchar(50)
contact	varchar(30)

Table: customers

Columns:

custid	int(11) AI PK
FirstName	varchar(50)
LastName	varchar(50)
DOB	date
Email	varchar(50)
contact	char(10)
pwd	varchar(20)

Table: earnings

Columns:

vendorid	int(11)
earning	int(11)
vendorproductid	int(11) PK

Table: highsold

Columns:

vendorproductid	int(11)
highsoldid	int(11) AI PK
vendorid	int(11)

Table: orderedproducts

Columns:

orderedproductid	int(11) AI PK
vendorproductid	int(11)
orderid	int(11)
quantity	int(11)

Table: orders

Columns:

orderid	int(11) AI PK
custid	int(11)
orderdt	date
addressid	int(11)
vendorcourierid	int(11)
trackingid	varchar(30)

Table: product

Columns:

productid	int(11) AI PK
productname	varchar(20)
categoryid	int(11)

Table: province

Columns:

provinceid	int(11) AI PK
proname	varchar(50)

Table: state

Columns:

stateid	int(11) AI PK
statename	varchar(50)

Table: vendor

Columns:

vendorid	int(11) AI PK
vendorname	varchar(50)
address	text
email	varchar(50)
vpass	varchar(30)
contact	varchar(30)

Table: vendorcourier

Columns:

vendorcourierid	int(11) AI PK
vendorid	int(11)
courierid	int(11)

Table: vendorproduct

Columns:

vendorproductid	int(11) AI PK
vendorid	int(11)
productid	int(11)
price	int(11)
quantity	int(11)
descr	varchar(10(

VIEWS CREATED

View: cartdisplay

Columns:

productname	varchar(20)
price	int(11)
productid	int(11)
vendorproductid	int(11)

View: orderedprod

Columns:

vendorproductid	int(11)
vendorid	int(11)
productid	int(11)
vendorcourierid	int(11)

View: productdisplay

Columns:

productname	varchar(20)
price	int(11)
descr	varchar(100)
productid	int(11)
categoryid	int(11)
quantity	int(11)

View: venearning

Columns:

productname	varchar(20)
earning	int(11)
vendorid	int(11)

View: venproddisplay

Columns:

productname	varchar(20)
price	int(11)
descr	varchar(100)
vendorproductid	int(11)
vendorid	int(11)

PROCEDURE

Name: earnings

The name of the routine is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

DDL:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `earnings`()
BEGIN
update earnings set earning=earning+(0.8*(SELECT price from vendorproduct v where
 v.vendorproductid=earnings.vendorproductid))
where earnings.vendorproductid=(SELECT vendorproductid from orderedproducts  order by orderedproductid desc limit 1);
END
```

noofstocks

The name of the routine is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `noofstocks`()
BEGIN
    SELECT categoryname,COUNT(p.categoryid) FROM product p,category c WHERE  p.categoryid=c.categoryid
    GROUP BY p.categoryid;
END
```

TRIGGERS

```
show triggers;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Trigger	Event	Table	Statement	Timing	Created	sql_mode	D
vendorproduct_AFTER_INSERT	INSERT	orderedproducts	BEGIN UPDATE vendorproduct SET quantity=qu...	AFTER	2022-06-06 22:13:39.28	NO_ZERO_IN_DATE,NO_ZERO_DATE,NO_ENGI...	ro
highsold_AFTER_INSERT	INSERT	orderedproducts	BEGIN INSERT INTO highsold(vendorproductid,...	AFTER	2022-06-07 22:43:17.87	NO_ZERO_IN_DATE,NO_ZERO_DATE,NO_ENGI...	ro
earning	INSERT	orderedproducts	BEGIN update earnings set earning=earning+1...	AFTER	2022-06-07 23:07:41.00	NO_ZERO_IN_DATE,NO_ZERO_DATE,NO_ENGI...	ro

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Trigger	Event	Table	Statement	Timing	Created	sql_mode	D
vendorproduct_AFTER_INSERT	INSERT	orderedproducts	BEGIN UPDATE vendorproduct SET quantity=qu...	AFTER	2022-06-06 22:13:39.28	NO_ZERO_IN_DATE,NO_ZERO_DATE,NO_ENGI...	ro
highsold_AFTER_INSERT	INSERT	orderedproducts	BEGIN INSERT IN				
earning	INSERT	orderedproducts	BEGIN update ea				

```
BEGIN
UPDATE vendorproduct SET quantity=quantity-1 WHERE vendorproduct.vendorproductid=NEW.vendorproductid;
END
```

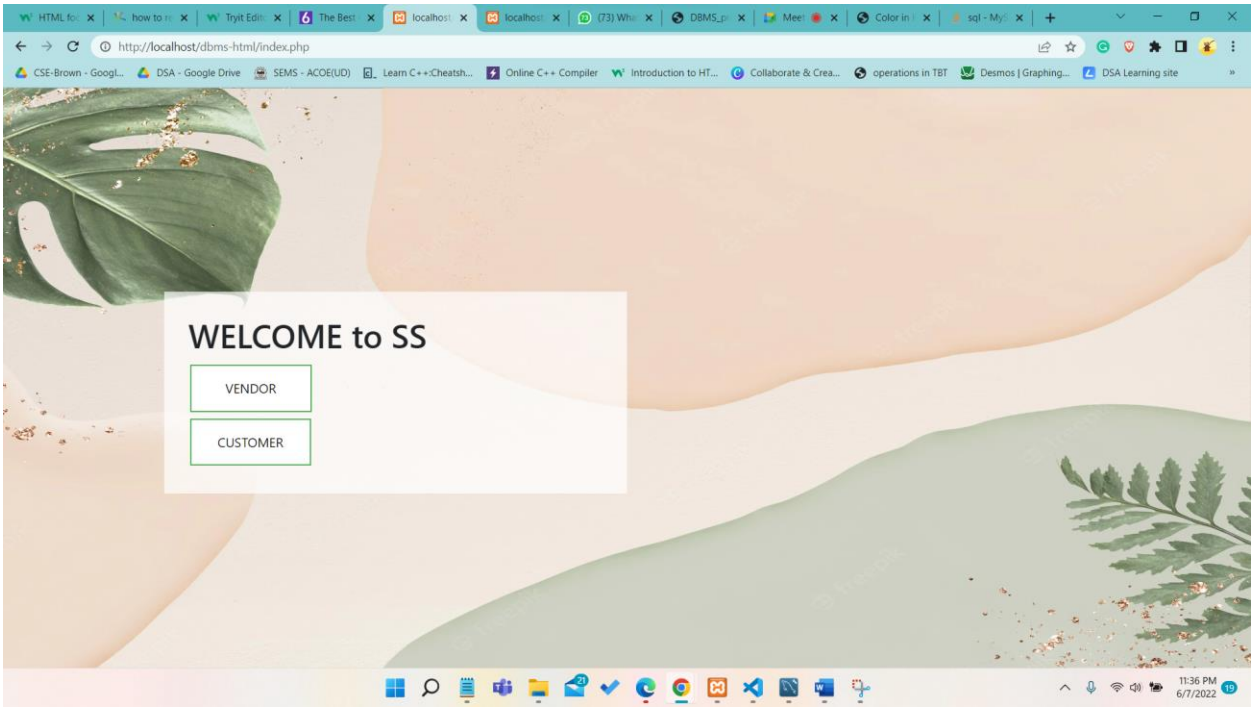
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

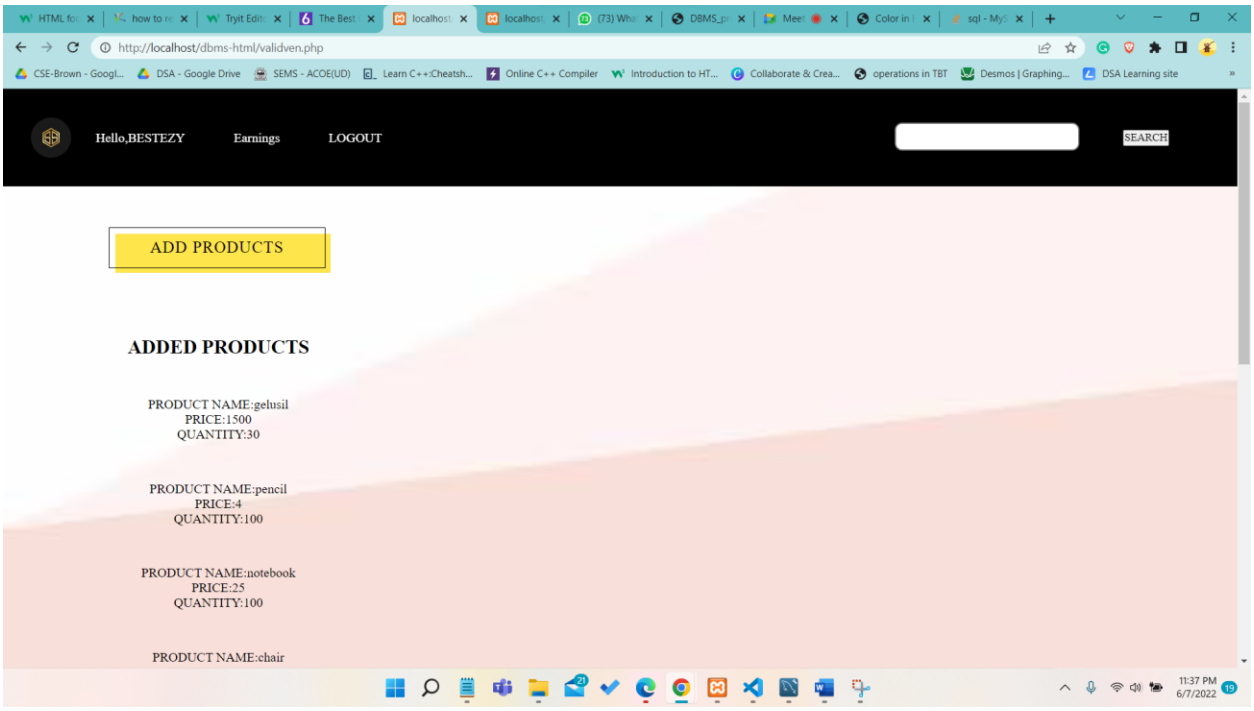
Trigger	Event	Table	Statement	Timing	Created	sql_mode	D
vendorproduct_AFTER_INSERT	INSERT	orderedproducts	BEGIN UPDATE vendorproduct SET quantity=qu...	AFTER	2022-06-06 22:13:39.28	NO_ZERO_IN_DATE,NO_ZERO_DATE,NO_ENGI...	ro
highsold_AFTER_INSERT	INSERT	orderedproducts				ENGI...	ro
earning	INSERT	orderedproducts				ENGI...	ro

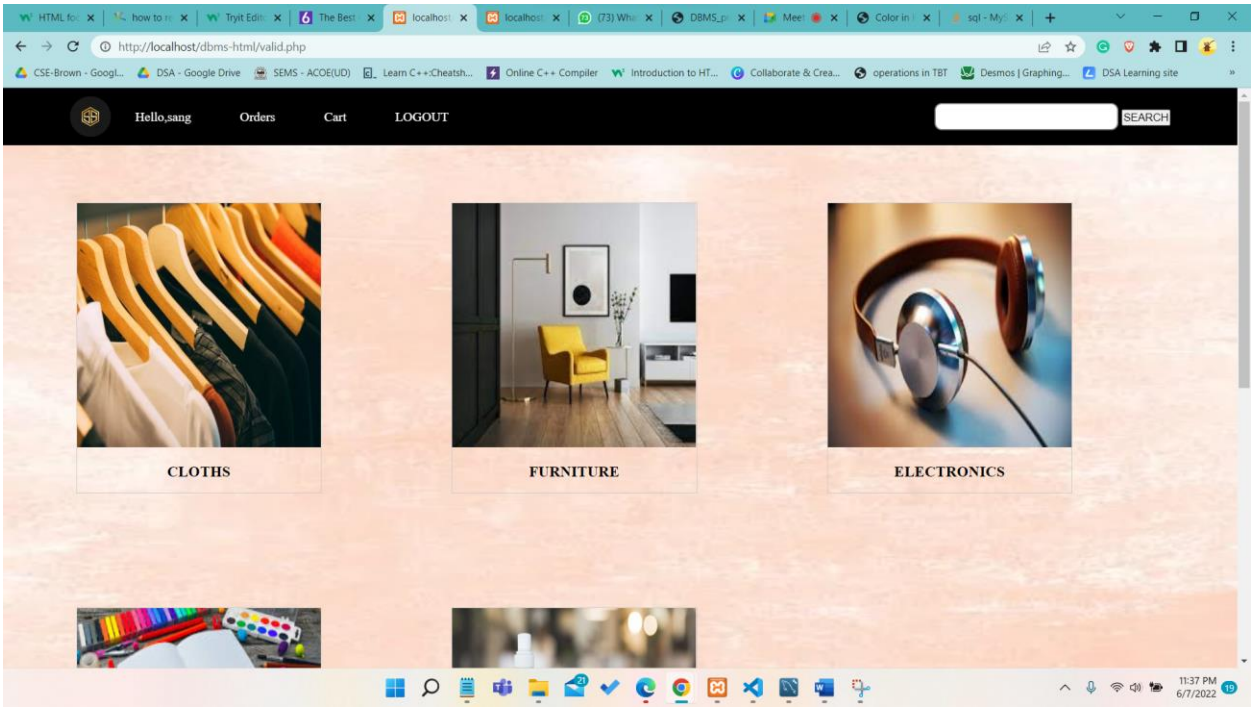
```
BEGIN
INSERT INTO highsold(vendorproductid,vendorid) values(new.vendorproductid, (SELECT vendorid From vendorproduct
where vendorproductid=new.vendorproductid));
END
```

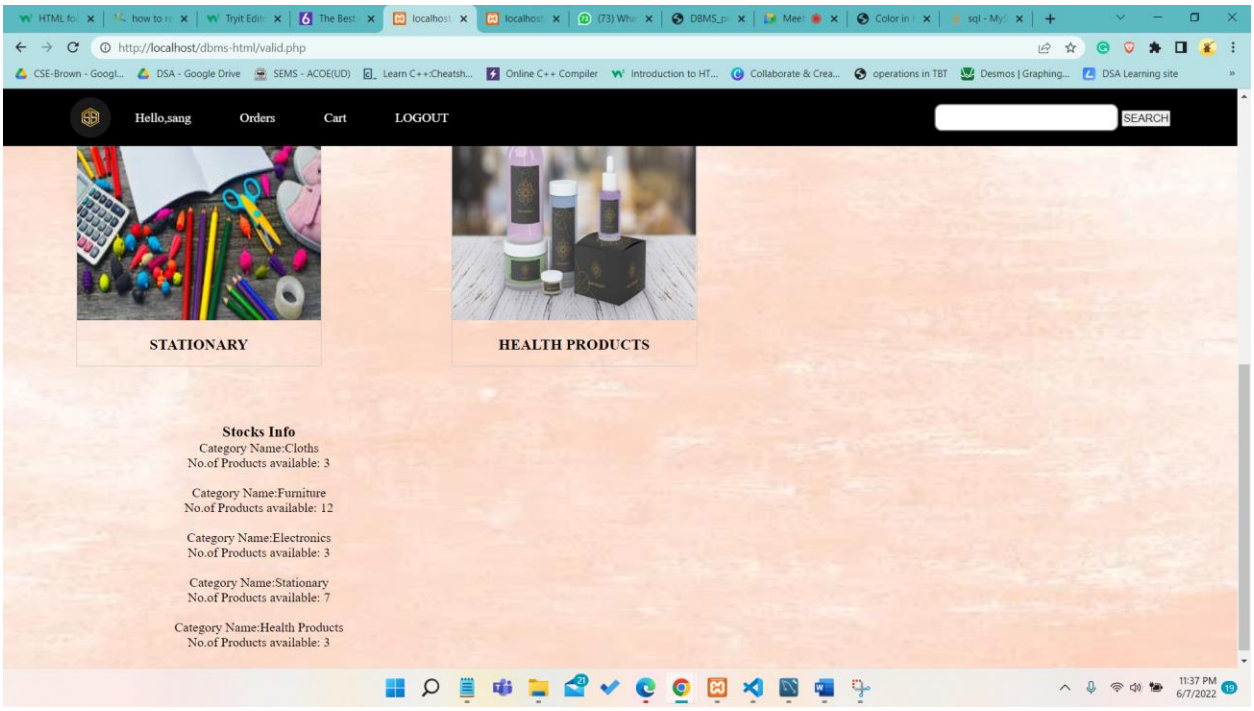
Trigger	Event	Table	Statement	Timing	Created	sql_mode	D
vendorproduct_AFTER_INSERT	INSERT	orderedproducts	BEGIN UPDATE vendorproduct SET quantity=qu...	AFTER	2022-06-06 22:13:39.28	NO_ZERO_IN_DATE,NO_ZERO_DATE,NO_ENGI...	ro
highsold_AFTER_INSERT	INSERT	orderedproducts	BEGIN INSERT INTO highsold(vendorproductid,...	AFTER	2022-06-07 22:43:17.87	NO_ZERO_IN_DATE,NO_ZERO_DATE,NO_ENGI...	ro
earning	INSERT	orderedproducts	BEGIN update			ATE,NO_ENGI...	ro

```
BEGIN
update earnings set earning=earning+10 where earnings.vendorproductid=new.vendorproductid;
END
```




http://localhost/dbms-html/validven.php
Hello,BESTEZY
Earnings
LOGOUT
SEARCH
ADD PRODUCTS
ADDED PRODUCTS
PRODUCT NAME:gelusil
PRICE:1500
QUANTITY:30
PRODUCT NAME:pencil
PRICE:4
QUANTITY:100
PRODUCT NAME:notebook
PRICE:25
QUANTITY:100
PRODUCT NAME:chair


http://localhost/dbms-html/valid.php
Hello,sang
Orders
Cart
LOGOUT
SEARCH
CLOTHS
FURNITURE
ELECTRONICS


http://localhost/dbms-html/valid.php
Hello,sang
Orders
Cart
LOGOUT
SEARCH
STATIONARY
HEALTH PRODUCTS
Stocks Info
Category Name:Cloths
No.of Products available: 3
Category Name:Furniture
No.of Products available: 12
Category Name:Electronics
No.of Products available: 3
Category Name:Stationary
No.of Products available: 7
Category Name:Health Products
No.of Products available: 3
11:37 PM
6/7/2022