**ONLINE RETAIL STORE**

**SHOPPX24**

**Brought to you by**

**Group 27**

Aditya Jain (2020554)   |          Rupin Oberoi (2020571)

Shashank Shekhar Singh (2020119) |    Vaibhav Wali (2020257)

**What is SHOPPX24 ?**

Now you don’t need to rush to the market all the time to buy daily products. We have

brought to you   SHOPPX24 - One STOP for all your shopping needs !!!

In this digitalized world , where so many online shopping applications are there then  why choose our app? The answer is fastest delivery at your comfort with lowest prices.

Now you just need to download our application on your pc / mobile phone and enjoy our unlimited services. You can buy  grocery items , daily-cares, stationeries, appliances , electronic devices and a lot more in just one click.

Just sign up your account , add your items into your cart and get them delivered at your doorstep.

We have flexible payment options so that nothing stops you from ordering.

**Scope of the project:**

This project aims to create an online shopping platform where users can purchase day to day products like grocery , electronics , appliances and many other products at their comfort.

It has 4 login options:

1. For Customers

2. For Employees

3. For Supplier

4. For Delivery persons

Customers can view different products , add them to their cart or wishlist and can place orders. They can check their orders placed so far. They can add, items ,remove them or update the cart as they want.

Employees can manage(add or remove or modify) items in the inventory , and belong to various departments such as marketing, customer service etc.

Suppliers can add items to the inventory as they get the new stock of that item. They can select particular pin-codes from the customers for delivery.

Delivery person can see what orders need to be delivered by them on a particular day

**Stakeholders :**

* Customer
* Supplier
* Accounts Department
* IT Department
* Warehouse
* Customer Services
* Sales Executives
* Marketing dept
* Delivery Department
* Suppliers
* Delivery person

**Entity , Attributes and Relationships :**

* **inventory:**

**Schema**: Inventory( item\_id, company\_name, item\_name, qty, price, category, discount)

**Attributes**

* item id : INT , NOT NULL , PRIMARY KEY
* company\_name: VARCHAR(30) , DEFAULT NULL
* item\_name: VARCHAR(50), NOT NULL
* quantity : INT ,NOT NULL
* price : INT , NOT NULL
* category : VARCHAR(20), NOT NULL
* discount : INT , DEFALUT NULL
* visibility : INT, DEFAULT 1
* **employee:**

**Schema**: Employee( emp\_id, first\_name, last\_name, phone no , doj , house\_no, locality, city, state, pin\_code, dob, department, salary, ,gender , username, password)

**Attributes**

* emp\_id : INT, NOT NULL , PRIMARY KEY
* first\_name: VARCHAR(30), NOT NULL
* last\_name: VARCHAR(30) , DEFAULT NULL
* phone\_no : BIGINT, DEFAULT NULL ,UNIQUE KEY
* doj : DATE , NOT NULL
* house\_no.: VARCHAR(40), NOT NULL
* locality: VARCHAR(25), NOT NULL
* city: VARCHAR(20), NOT NULL
* state: VARCHAR(30), NOT NULL
* pincode: INT , DEFAULT NULL
* dob : DATE DEFAULT NULL
* department : VARCHAR(20) , NOT NULL
* salary : INT , NOT NULL
* gender : CHAR(1) , DEFAULT NULL
* username : VARCHAR(15), DEFAULT NULL , UNIQUE KEY
* password : VARCHAR(64), NOT NULL
* **customer**:

**Schema**: Customer(cust\_id, first\_name, last\_name, phone\_no , house\_no, locality, city, state, pin\_code, username, password)

**Attributes**

* cust\_id : INT, NOT NULL , PRIMARY KEY
* first\_name: VARCHAR(30), NOT NULL
* last\_name: VARCHAR(30) , DEFAULT NULL
* phone\_no : BIGINT, DEFAULT NULL ,UNIQUE KEY
* house\_no.: VARCHAR(40) , NOT NULL
* locality: VARCHAR(25) , NOT NULL
* city: VARCHAR(20), NOT NULL
* state: VARCHAR(30), NOT NULL
* pin\_code: INT , DEFAULT NULL
* username : VARCHAR(15), DEFAULT NULL
* password : VARCHAR(64), NOT NULL
* **cart** :

**Schema**: cart(cart\_id,qty,cost)

**Attributes**

* cart\_id: INT , NOT NULL ,PRIMARY KEY
* qty: INT, DEFAULT NULL
* cost: INT, DEFAULT NULL
* **cart\_item:**

**Schema**: cart\_item( item\_id, cart\_id, qty, cost )

**Attributes**

* item\_id: INT , NOT NULL, PRIMARY KEY, FOREIGN KEY
* cart\_id: INT , NOT NULL , PRIMARY KEY, FOREIGN KEY
* qty: INT, NOT NULL
* cost: INT, NOT NULL
* **cust\_cart\_rel**:

**Schema**: cust\_cart\_rel( cust\_id, cart\_id )

**Attributes**

* cust\_id: INT , NOT NULL, PRIMARY KEY, FOREIGN KEY
* cart\_id: INT , NOT NULL , PRIMARY KEY, FOREIGN KEY
* **orders**:

**Schema**: orders(order\_id, order\_date, qty, cost, order\_status, payment\_id)

**Attributes**

* order\_id : INT, NOT NULL , PRIMARY KEY
* order\_date: DATE, NOT NULL
* qty: INT , NOT NULL
* cost: INT, NOT NULL
* order\_status: VARCHAR(15) , NOT NULL
* payment\_id: INT, NOT NULL , FOREIGN KEY
* **order\_item**:

**Schema**: order\_item (item\_id, order\_id, qty, cost)

**Attributes**

* item\_id : INT,NOT NULL , PRIMARY KEY , FOREIGN KEY
* order\_id : INT, NOT NULL , PRIMARY KEY , FOREIGN KEY
* qty : INT , NOT NULL
* cost : INT ,NOT NULL
* **cust\_order\_rel**:

**Schema**: cust\_order\_rel( cust\_id, order\_id )

**Attributes**

* cust\_id: INT , NOT NULL, PRIMARY KEY, FOREIGN KEY
* order\_id: INT , NOT NULL , PRIMARY KEY, FOREIGN KEY
* **payments**:

**Schema** : payments (payment\_id , payment\_method)

**Attributes**

* payment\_id : INT , NOT NULL , PRIMARY KEY
* payment\_method : VARCHAR(15) , NOT NULL

**supplier:**

**Schema:** supplier(co\_name, co\_id, phone\_no, email, address\_l1, address\_l2,

pincode, password)

**Attributes:**

* co\_name: varchar(100), not null
* co\_id: int, primary key
* phone\_no: char(10), not null
* email: varchar(100)
* address\_l1: varchar(30), not null
* address\_l2: varchar(100), not null
* pincode: int
* password: varchar(30), not null
* supply\_areas: (multivalued and hence separate table is created)

* **Delivery\_person:**

**Schema:** (dlvr\_id, first\_name, last\_name, phone\_no, city, state, country, salary,

username, password)

**Attributes:**

* dlvr\_id: int, primary key
* first\_name: varchar(30) not null
* last\_name: varchar(30)
* phone\_no: bigint
* city: varchar(30)
* state: varchar(30)
* country: varchar(30)
* salary: int
* username: varchar(40), not null, unique
* password: varchar(15), not null
* works\_in\_pincodes (multivalued, hence a separate table is created)
* **Weak Entity**:
* cart\_item
* order\_item

It is because cart\_item entity depends on customer\_cart entity. If there is no customer cart then no items can be added to it.

Similarly, order\_item entity depends on customer\_order. If there is no customer order then no order items can be viewed.

**Relationships**

|  |  |  |  |
| --- | --- | --- | --- |
| ENTITY 1 | ENTITY 2 | RELATION | TYPE |
| customer | cart | belongs\_to | One to one |
| cart | cart\_item | cart\_contains | One to many |
| cart\_item | inventory | part\_of\_cart | Many to one |
| inventory | order\_item | part\_of\_order | One to many |
| order\_item | orders | order\_contains | One to one |
| orders | customer | ordered | Many to one |
| employee | inventory | manages | Many to many |
| supplier | Inventory | supplies\_item | Many to many |

**Ternary relationship**

Entity 1: Inventory

Entity 2: Customer

Entity 3: orders

Relation: ordered

**Indexing**

* Relation: Employee

Attribute: Salary

* Relation: Inventory

Attributes: price, discount ,item\_name

**Triggers**

* make\_new\_carts – On addition of a new customer to the customer table, the corresponding cart is created and a record is also added to the cust\_cart\_rel table.
* cart\_upd\_cost - If an employee changesthe price of an item in inventory then this trigger will change its price in the cart of the customers who have selected this item.

cart\_upd\_cost - If an employee changes the visibility of an itemin the inventory i.e. makes it invisible for the customersthen the item gets removed from the customer’s cart.

**Views :**

* create view cust\_view\_inv as select item\_id, company\_name, item\_name ,price, category, discount from inventory where visibility = 1 and qty > 0 ;

--- It allows the customer to view only those items from the inventory that are made visible by the employee and whose qty > 0.

* Create view emp\_view\_emp as select emp\_id , first\_name , last\_name , phone\_no ,email\_id , city , department from employee

--- It does not allow the employees to view other employees salary and address.

* Create view cust\_view\_contact as select first\_name , last\_name , phone\_no , email\_id from employee where department=’customer\_service’;

--- It allows customers to see only those employees whose department is customer service.

* create view cust\_view\_about\_us as select first\_name , last\_name , email\_id from employee where department=’management’

--- It allows the customer to view the details of the employees in management department when he clicks on about us option.

**Grants:**

* Employees can edit (add ,delete, update) items in the inventory.
* Customers can view or select inventory and edit their cart.

**GUI Design :**

The project includes a simple GUI for easy and attractive user interaction with the application. The design is made simple so that the user does not have to tackle a lot to try all the features of our application.

We have used “ tkinter ” library of python for designing the graphical interface. It provides a complete toolkit for designing an application.

When a user(whether a customer or an employee) runs the application , the first page he gets to see is the login page where he can enter his username and password or if the user is using the application for the first time then first he needs to register with our app.

New registration page asks for the personal details of the user like his name, phone number, email, address and finally the username and password he wants to set for the app.

Then after entering username and password, the user will click login as customer or login as employee according to the category he belongs.

For the customer , he gets to see the inventory as he logins the app. There is a

button to show user cart and also to show previous orders . For adding an item to

the cart , user has to enter the item id and its quantity in the respective fields and

then click on “ add to cart ” button . It is to note that the user should not enter

invalid input like wrong item id or item quantity that is not available otherwise the

program will give error.

It has an option to search a particular item in the inventory by its name. The user

can also sort the inventory items according to their names, prices and

discount ,both in ascending and descending order according to his ease.

For the employee, same GUI as for customer . In addition there is an option to update the quantity of any item in accordance with the stock , visibility of an item,

price and discount on any item.

For the suppliers they have an option to add new products, update product quantity .

For delivery persons, they can see the order details of the order to be delivered like delivery date, address .

**Contribution**

Aditya Jain - E-R diagram , Documentation , MYSQL, Python

Rupin Oberoi - E-R diagram , Tables creation , MYSQL ,SQL queries,GUI

Shashank Shekhar Singh - Python code , MYSQL , E-R diagram ,SQL queries, GUI

Vaibhav Wali - Python code , MYSQL , github setup, Queries