**CS6750 S18: Group 1**

**Project idea**

* Business Model
  + Online Retail Management System
* Importance
  + Simplify and optimize tracking, management of and maintenance of online business
  + Allowing online sellers to have a statistical view of customers’ consumption habits

**User requirements**

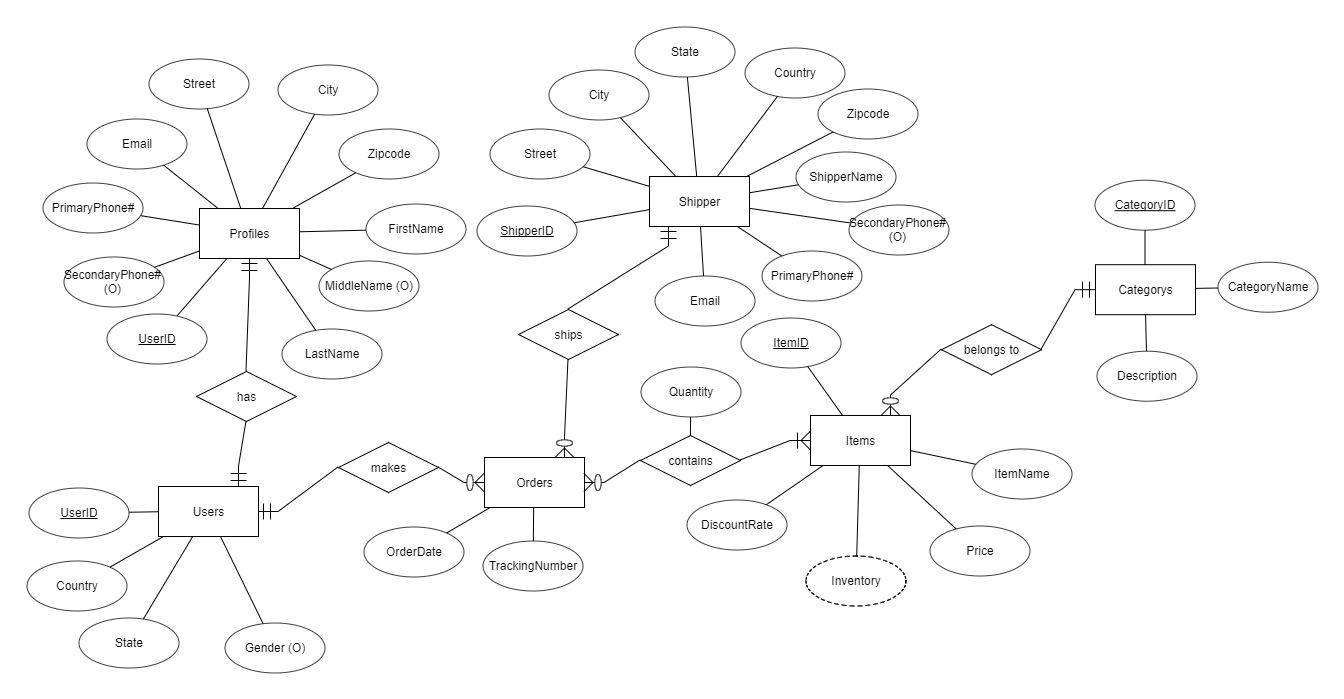
* Be able to update item inventory at the click of a button.
* Be able to see which items and categories are selling the most.
* Be able to make orders for users.
* Be able to see user order history.
* Be able to view records of all database tables.

**Database tables and data sources**

* Tables
  + Users
  + Profiles
  + Categories
  + Shippers
  + Items
  + Orders
  + OrderDetails
* Sources
  + We generated the data ourselves.

(see next page for more)

**ERD**



**SQL queries**

* Trigger for adding order details/updating inventory upon ordering an item:

BEGIN

IF (NEW.quantity > (Select Inventory from Items where New.ItemID = Items.ItemID)) THEN

SIGNAL SQLSTATE '02000' SET MESSAGE\_TEXT = 'Warning: No inventory left!';

ELSE

UPDATE Items i set i.inventory = i.inventory - NEW.quantity where i.ItemID = NEW.ItemID;

END IF;

END

* Update inventory:

UPDATE Items SET Inventory = <inventory input> WHERE ItemID = <itemID input>

* Add Item

INSERT INTO Items (ItemName, CategoryID, Price, DiscountRate, Inventory) VALUES('$inputItemName','$item\_category','$inputItemPrice', '$inputItemDiscount','$inputItemInventory')

* Add Order

INSERT INTO Orders (UserID, ShipperID, OrderDate, TrackingNumber) VALUES('$user\_name','$shipper\_name','$OrderDate', '$tracking\_number')

* Add OrderDetail

INSERT INTO OrderDetails VALUES('$OrderID','$item\_name', '$inputOrderQuantity')

* Delete Order (if trigger sends warning message)

DELETE FROM Orders WHERE OrderID = <OrderID>

* View User Order history:

Select Username, ItemName, Price, Quantity, OrderDate FROM (SELECT Username, ItemName, Price, Quantity, OrderDate FROM Users u LEFT JOIN Orders o ON u.UserID = o.UserID LEFT JOIN OrderDetails od ON o.OrderID = od.OrderID LEFT JOIN Items i ON od.ItemID = i.ItemID) t WHERE Username = <username input>;

* View most popular categories:

Select c.CategoryName, SUM(b.Quantity) from Items a

left join OrderDetails b

on a.ItemId = b.ItemID

left join Categories c

on a.CategoryID = c.CategoryID

Group by c.CategoryName

Order by SUM(b.Quantity) DESC;

* View most popular items:

Select b.CategoryName, a.ItemName, SUM(c.Quantity) from Items a inner join Categories b

on a.CategoryID = b.CategoryID

inner join OrderDetails c

on a.ItemID = c.ItemID

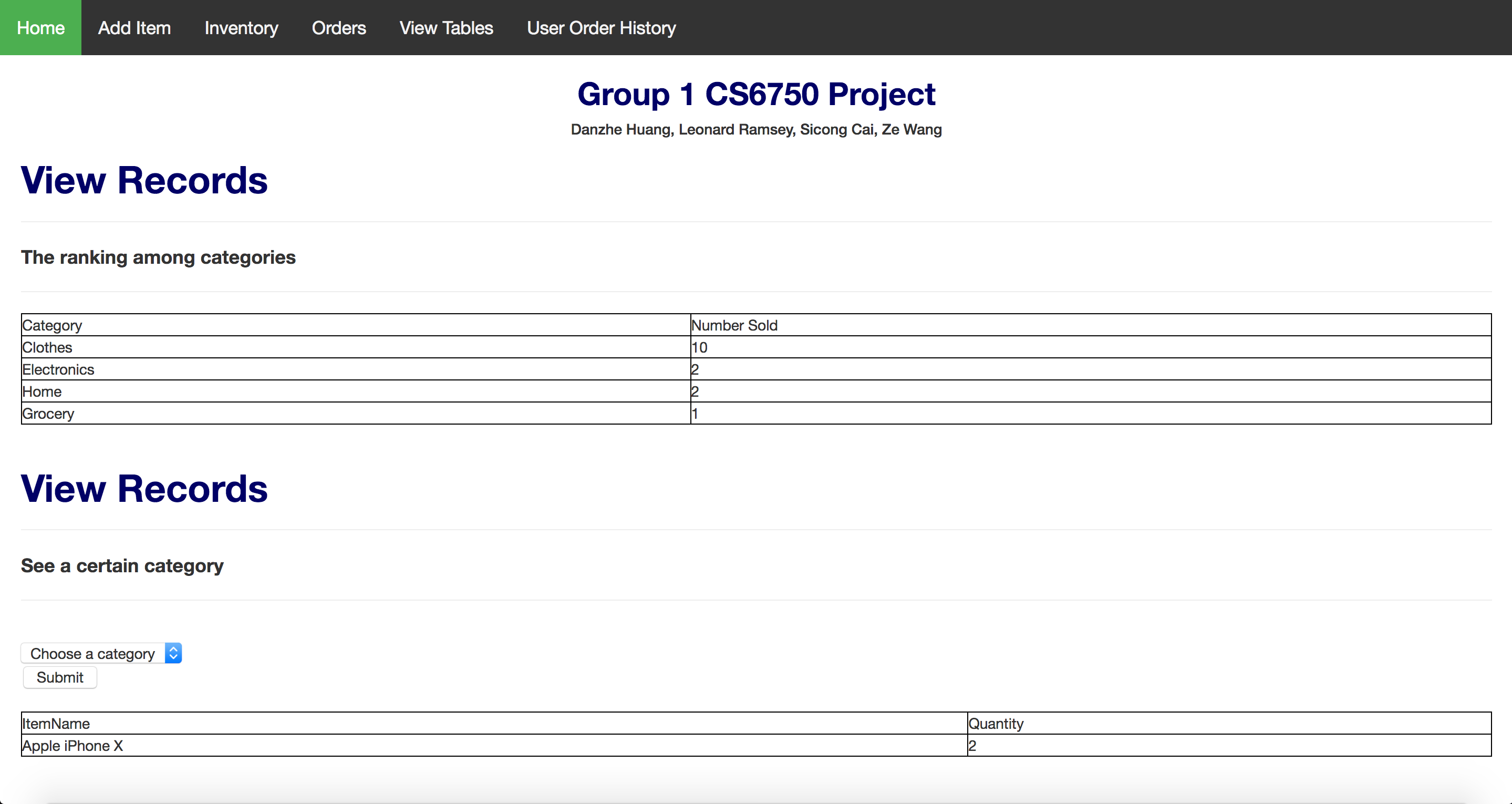
where b.CategoryName = <category input>

GROUP BY a.ItemName

Order by c.Quantity DESC

**Sample results**

The website is fast, simple, and has several simple/complex database functions. A screenshot of the homepage is below, and for more, see the two links under “Application interface”.



Tools for your interface implementation,

HTML/CSS/PHP

Application interface.

Source code: <https://github.com/leonardramsey/cs6750_g1>

Application: <http://plato.cs.virginia.edu/~lr3hj/index_g1.php>