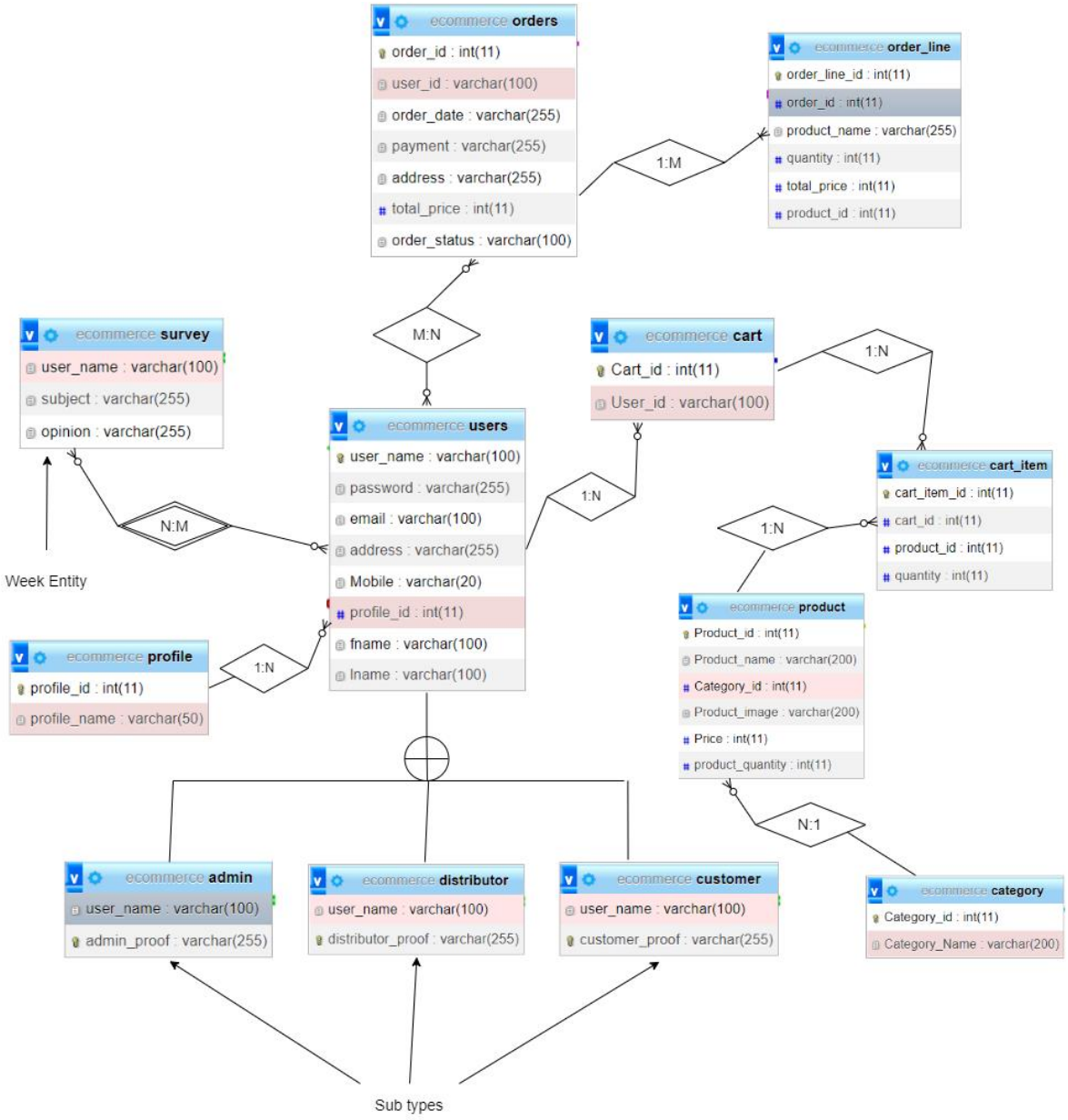
CS 5423: Database Project – Part II Report

NAME: Sai Srikanth Sarabu

CWID: A20343781

My Database design is as described as below ER diagram, which consists of 12 tables: users, profile, product, category, cart, cart_item, order, order_line, admin, customer, distributor and survey. Here each table has its significance, the user table stores all user's data like customer, admin, distributer.



Entity Relationship DIAGRAM

The users table is related to multiple tables, it has profile id as foreign key, profile table consists of profile names, took separate profile table for future purposes to add more privileges based on the profile. The user table has sub types of admin, distributor, customer, each table has its own type of proof for the respective profile enter as proof, so sub types for each of the user is taken. Here admin has to show admin license as proof, distributor has to show their marketplace id, customer need to show government id. In this way each subtype varies.

The user table has survey as its weak entity, this survey does not have a primary key, it stores all the survey information in it, where it is completely dependent on users table.

Products table consists of product image, category_id, product_id, price, product_quantity, product_name, here category id is foreign key from category table which consists of category name. Products can be added by admin or distributor by logging in as admin or distributor profile. Cart table stores all the items when user clicks on add to cart, the cart stores user_id which is specific each user. The cart_item table stores all the products selected by user in the cart, each product quantity selected and total price for each product. Here the product id is the foreign key to products table.

The orders table stores all the orders placed and order_line stores items ordered, the order_line consists items related to order by the user. The orders table consists of date, total price, payment information, address to be shipped and order status. Order_line stores product_name, product_id, quantity, total price for the particular product ordered.

Normalization of tables: All the table shown above are in BCNF.

All the tables are in **1 NF**, since all the columns in table are atomic, has same data type, and can store only single value. And all the tables are in **2 NF**, 2nd Normal form is nothing but table should be in 1st normal form and no not have any functional dependency, i.e., non prime attribute should not depend on prime attribute, but all my table have only one attribute as their primary key, therefore no table can have functional dependency.

For the **3rd normal form**, each table should be in 2nd normal form and do not have transitive dependency, i.e., each attribute in the table should not depend non prime attribute.

Table **users**: It has username as its primary key which is unique, and remaining columns password, email, address, Mobile, profile_id, fname, lname, each of the non prime attributes are not depended on each other. They can have values that do not depend on any attribute. Therefore users is in 3 NF.

Table **admin**: admin consists of 2 column username and admin_proof, the username is primary key, and the admin_proof is non prime attribute, here we have only one non prime attribute which do not depend on any non prime attribute. Similarly tables **distributor, customer, profile, category, and cart** which have only one non prime attribute, therefore those tables are in 3NF. The survey table has no primary key, it has partial key since it is weak entity and all the attributes no not depend on each other.

Table **product:** It has product id as its primary key which is unique, and remaining columns product_name, category_id, product_image, size, product_quantity each of the non prime attributes are not depended on each other. They can have values that do not depend on any attribute. Therefore product is in 3 NF.

Table **orders**: It has order_id as its primary key which is unique, and remaining columns user_id, order_date, payment, total_size, address, order_status, each of the non prime attributes are not depended on each other. They can have values that do not depend on any attribute. Therefore orders is in 3 NF. Similarly for **cart_item**.

Table **order_line** : This table may look like having product id and product name, here product_name is the name in which user has ordered, if the admin or distributor, updates the name of the product, then the order_line gets updated therefore the product_name is independent of the the product_id, in this way the order_line is in 3NF.

BCNF: For a table to be in BCNF it need to be in 3NF and prime attribute should not depend on the non prime attribute, it is like non prime attribute should not derive prime attribute.

Table **users** username is unique and not depend any other attributes present in the table. The Tables **cart, admin, customer, distributor, profile and category**, has only 2 attributes, which do not depend the non prime attribute, each primary key is unique. Therefore are in BCNF.

The **orders** table have order_id as primary key which is autoincremented and not depend on any of the non prime attributes, similarly for **order_line, cart_item, product**, all are in BCNF.

The tools and languages used in this project are:

XAMPP (X (cross platform), Apache, MySQL, PHP, Perl): Tool for the execution, Apache server, MySql database, php programming language.

And I imported bootstrap html, javascript and css packages for styling purposes.

My high level implementation is front end is in html, javascript and css backend is completed implemented in php using MySql database. Generic user can view the website home page and surf the ecommerce home for products, will be able to choose product based on the category and even can search for the products. If the user likes any of the product and wishes to buy the product, the user can sign if the account already exists, the user sign up to create account, once created user can sign in, add the product to cart, select the quantity required and provide payment information and address for shipping and place the order. The user can check the order status clicking on the orders in the home. The user can even submit survey report as feedback to improve the website.

Admin page for admins and distributors, in the page the admin can monitor products availability and adding new products and creation of admin and distributors. And admin can

add category if any category arrives to the website and look into the userslist, orders placed by the customers, and survey feedback back from the users.

TRIGGERS used in the project:

beforeCartItemDelete : This trigger is used when the item is deleted from the cart, the product_available attribute for the specific product can increase the quantity selected, as the user no longer wishes to buy, so it can be available for other customers.

beforeCartItemInsert : This trigger is used when the item is inserted into the cart, the product_available attribute for the specific can decrease the quantity selected as the user opting to buy. Resultant available to be reduced for other customers.

beforeCartItemUpdate : This update trigger is used to update the product_available based on the updated quantity by the user.

beforeOrderlineInsert: This trigger is used to reduce product_available as soon as order is placed.

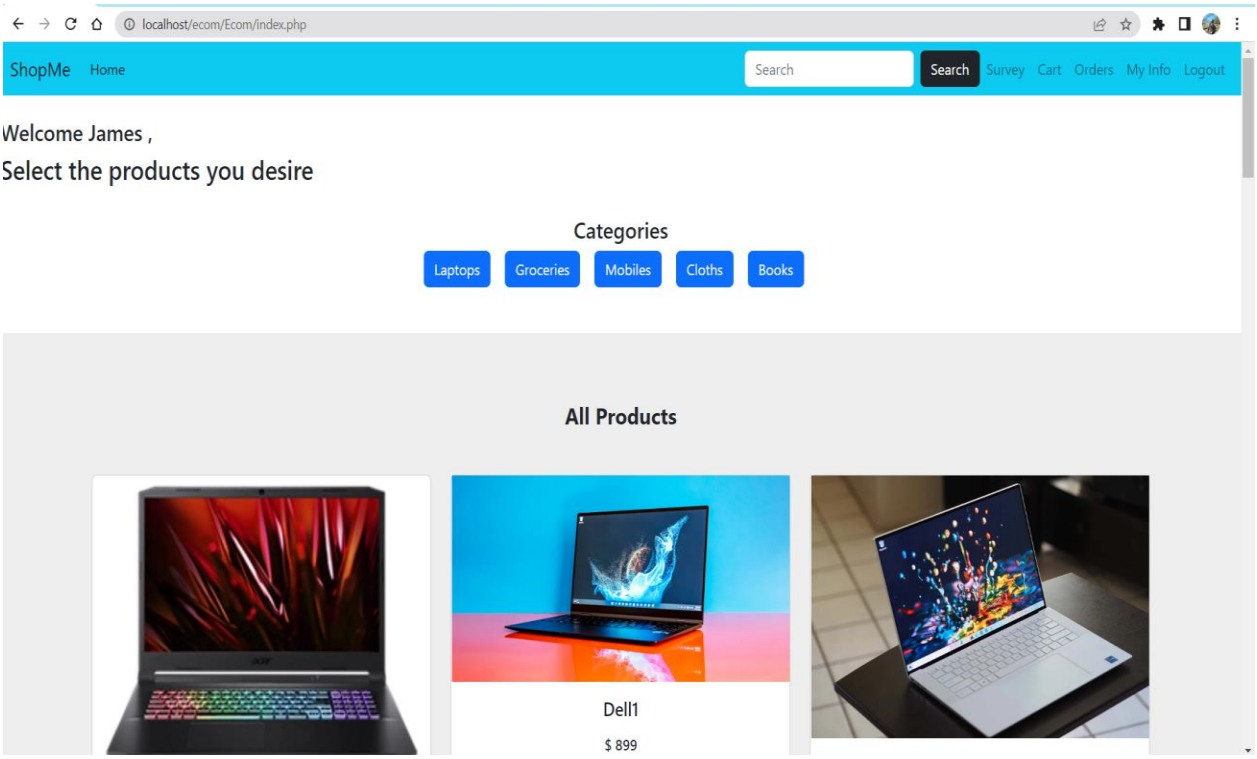
PROCEDURE used in this project:

I used procedure when deleting product by the admin in the admin page, the procedure takes IN parameter as product_id and runs delete query on the product for the particular product_id in the procedure. Procedure name: **Product_delete**

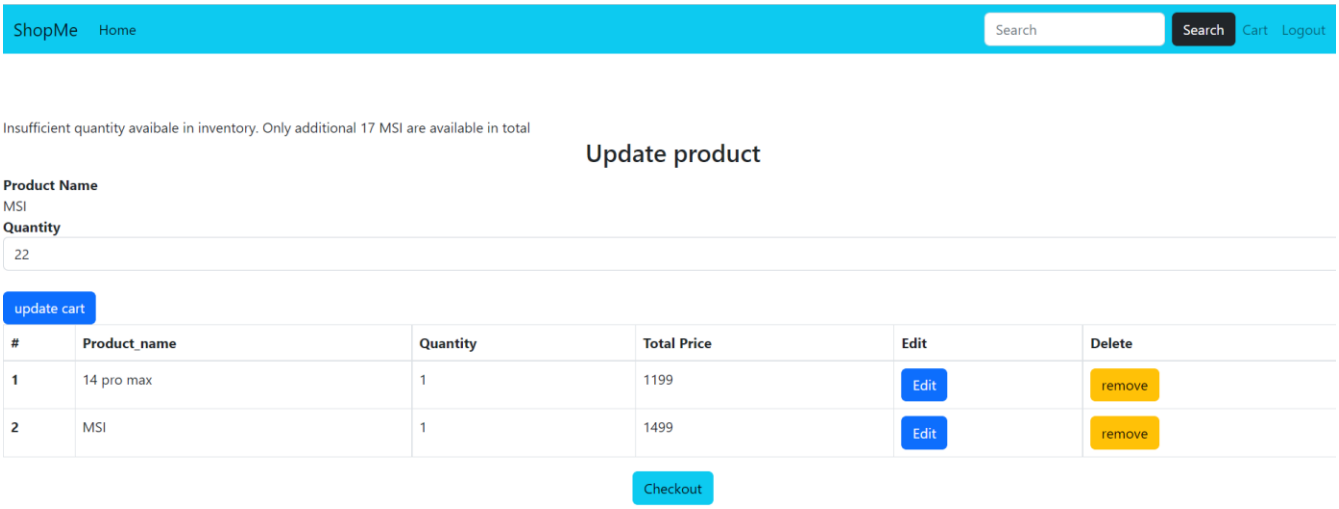
VIEW used in this project:

I used view in the admin page, for the admin to view users list available in the database, I used JOIN to tables users and profile and created a view named **'userslist'.**

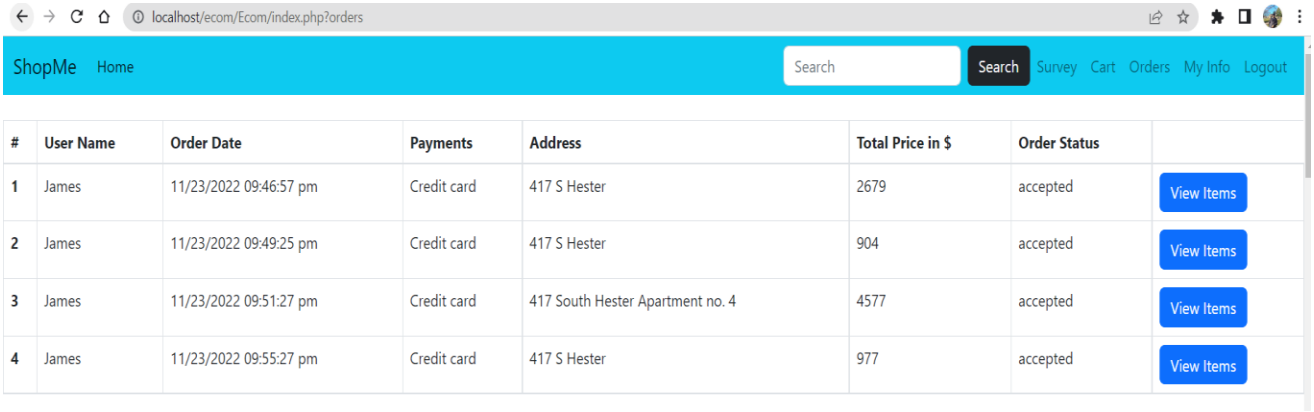
SCREENSHOTS



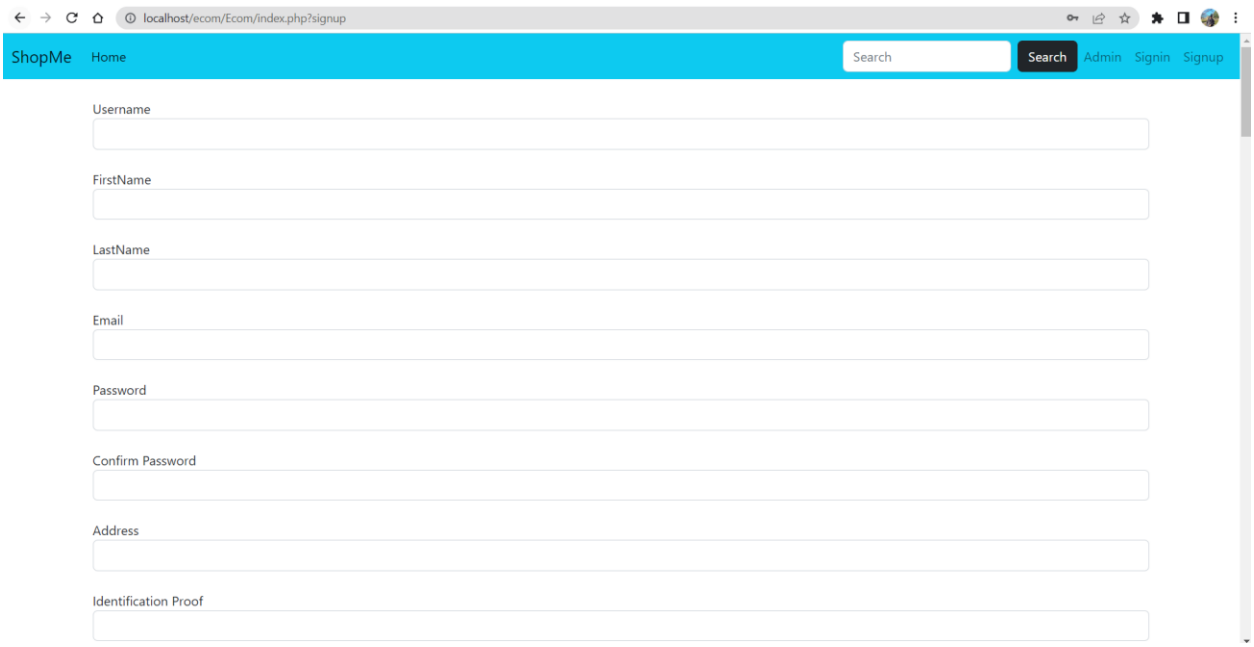
Screenshot shown above is the home for the website. User can surf and select products to buy based on the category and can search for products by searching.



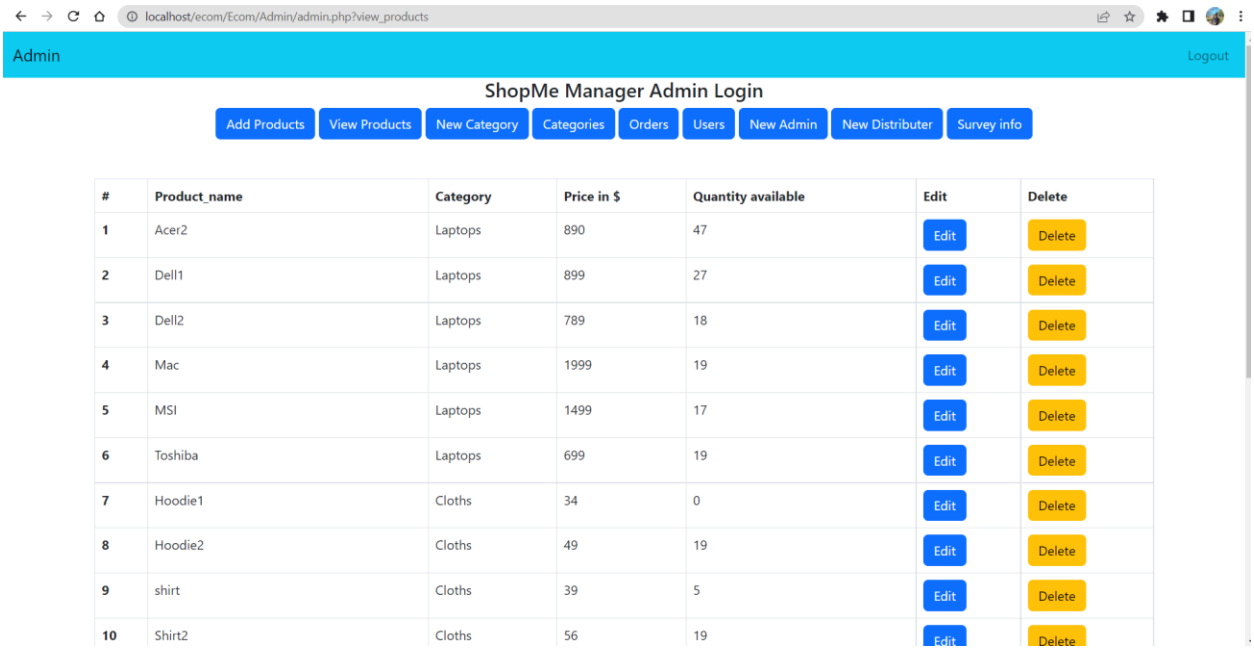
Screenshot shown above is the cart, the user can view cart and update the quantity by clicking edit button. User will be prompted based on the availability of the product as shown above



Screenshot shown above is the orders page, All the orders placed by the used are shown when users user clicks view items, he can see products bought for that particular order.



The above screenshot is sign up page for the customer, the customer can signup by entering unique username, which do not exist already.



#	Product_name	Category	Price in $	Quantity available	Edit	Delete
1	Acer2	Laptops	890	47	Edit	Delete
2	Dell1	Laptops	899	27	Edit	Delete
3	Dell2	Laptops	789	18	Edit	Delete
4	Mac	Laptops	1999	19	Edit	Delete
5	MSI	Laptops	1499	17	Edit	Delete
6	Toshiba	Laptops	699	19	Edit	Delete
7	Hoodie1	Cloths	34	0	Edit	Delete
8	Hoodie2	Cloths	49	19	Edit	Delete
9	shirt	Cloths	39	5	Edit	Delete
10	Shirt2	Cloths	56	19	Edit	Delete

The above screen shot is from admin login, the admin look into all the products and edit the product information, and delete if the admin wishes to.