**Project Members;**

**-Hasan Naseem, Faraz Ozair and Hozefa Haider**

**Introduction**

Database Management Systems is one of the Course that Computer Science Students at Habib University have to take in order to complete their course requirements. The course aims to target the concept of databases, their management and targets some concepts of Data Science. Part of the Course Requirements is create a project that aims to satisfy a real business scenario, incorporating all the principles taught in the Course.

My friends and I created a website and named it Dudepanda, a website where people can order bakery items and get it delivered to their place. Following is a detailed description of our project and the steps carried out to complete it.

**Business Scenario**

Dudepanda is a simple ecommerce marketplace of bakeries located in karachi. Customers can save themselves from the hassle of going to the bakery and buying items rather place an order through this website and get it delivered directly to their place like ordering a birthday cake for their loved ones.

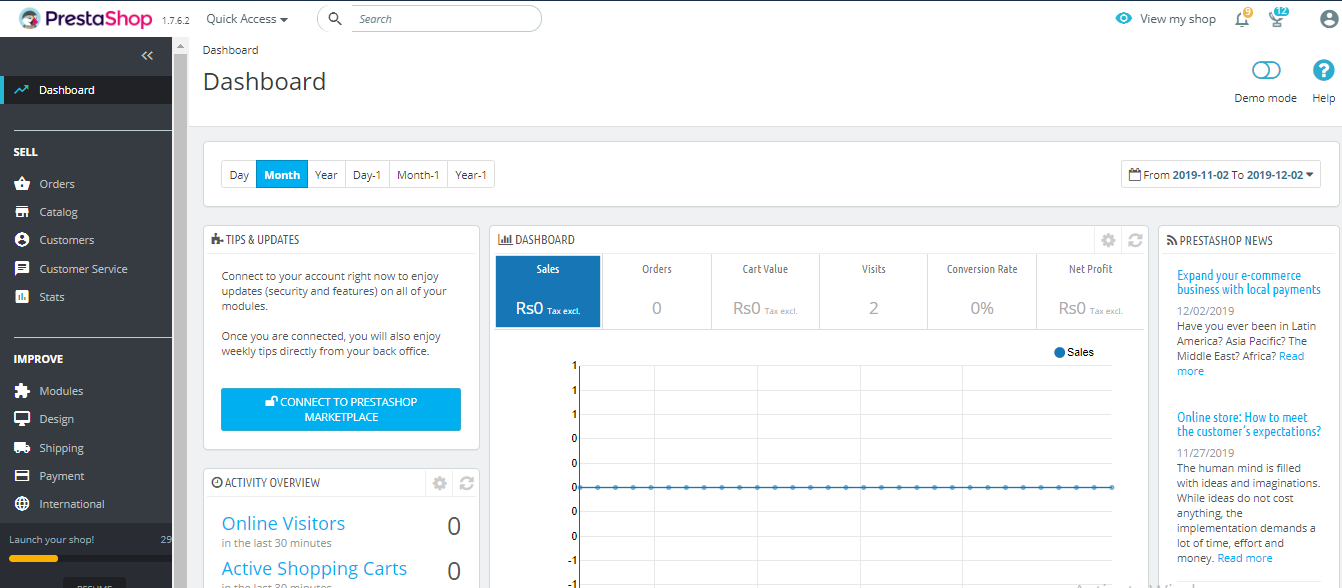
The website required a database which would essentially contain the data of All the Vendors, Customers, Food items, Orders etc. So the first job was to decide and create an entity relationship diagram (ERD) that would form the basis of our Database.

Since creating a website and then launching it meant for us to buy a domain, we just made an offline website using Prestashop.

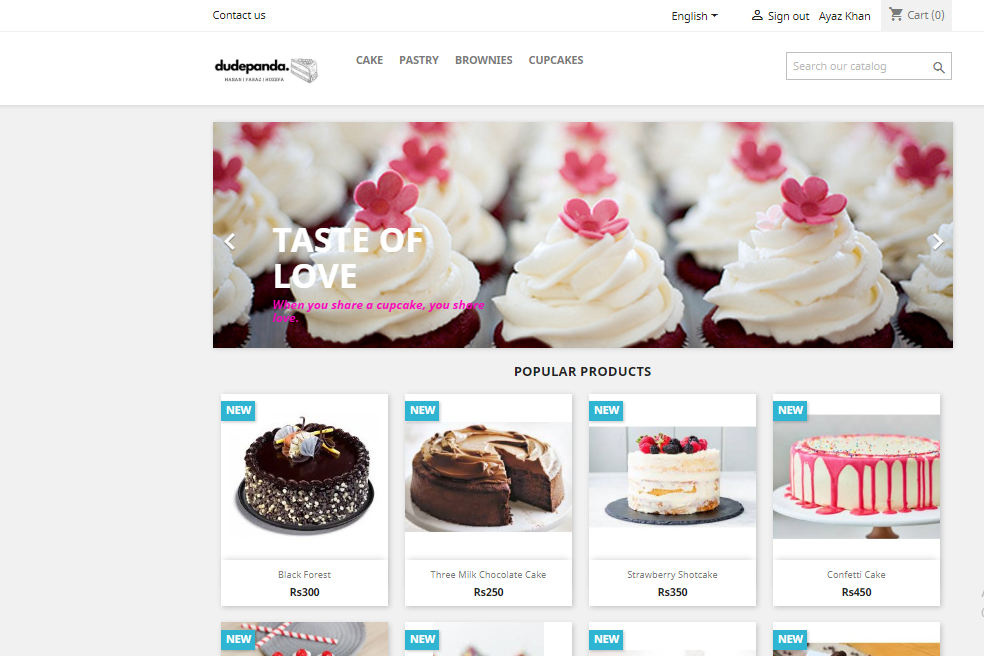
**MODULES & FUNCTIONALITY**

The website has three main interfaces or modules which is the user interface, admin interface and MySQL administration tool.

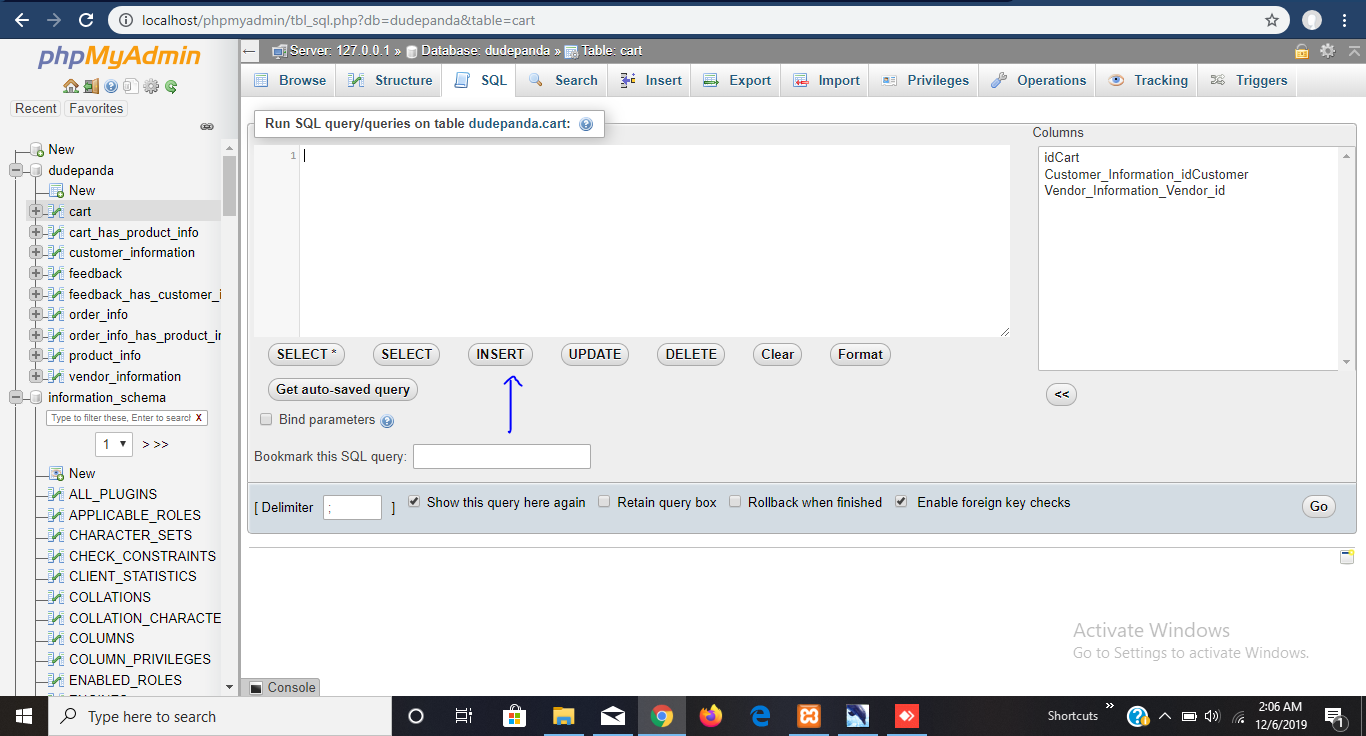
1. **admin interface**



1. **and user interface/Front end**

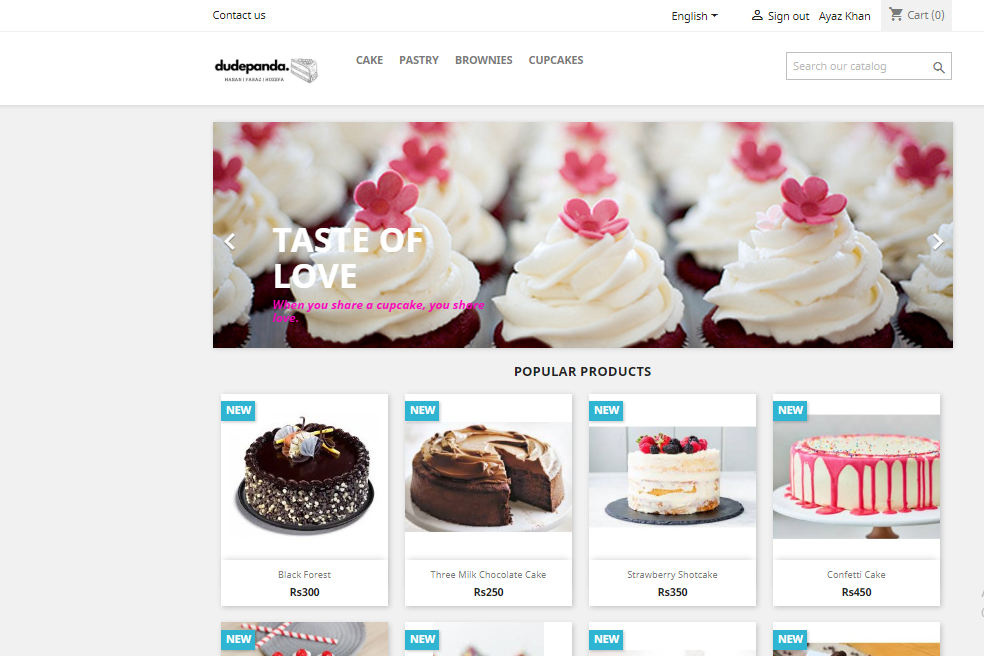


1. **MySQL administration tool**;



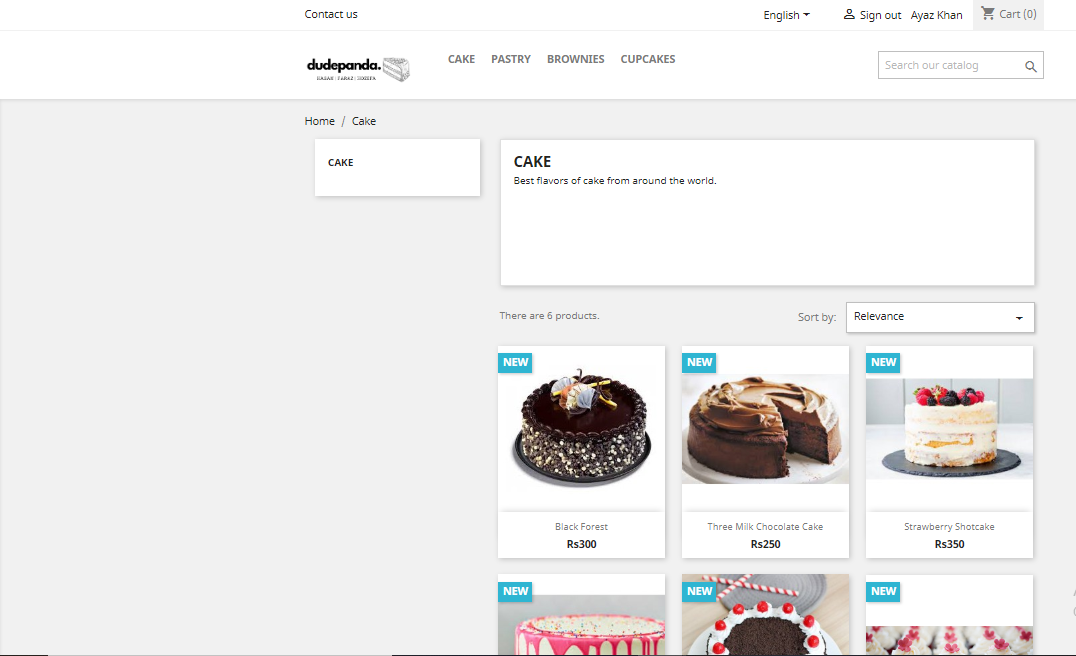
Now I’ll walk you through some of the functionalities of the website.

1. **HOW TO ORDER AN ITEM**
2. Following is the front end;



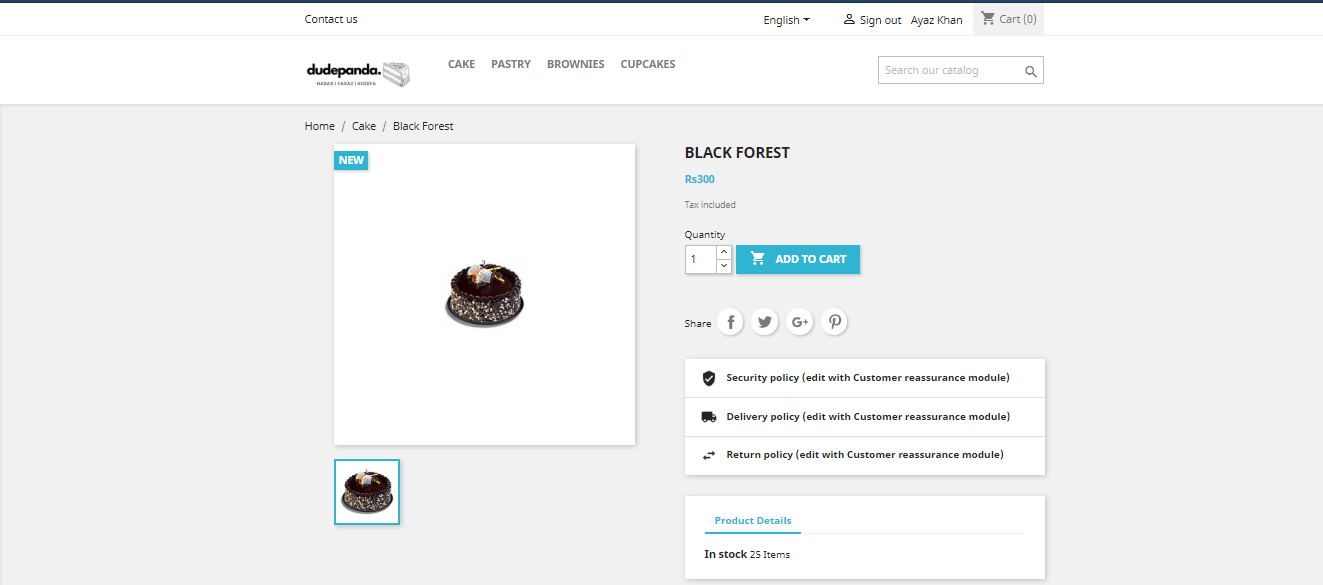
So let's order a cake

1. Click on *cake* on the top



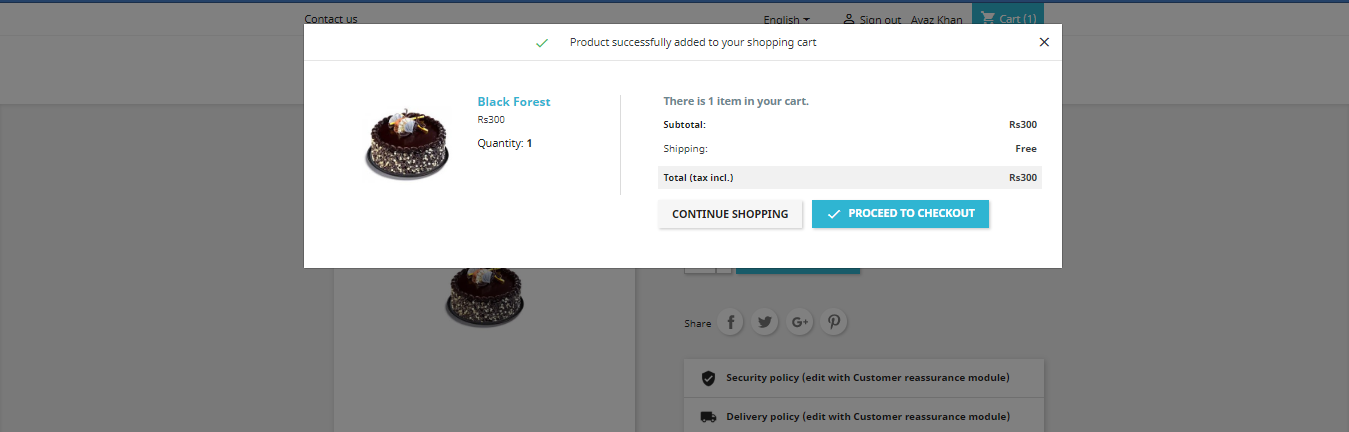
The following screen will appear

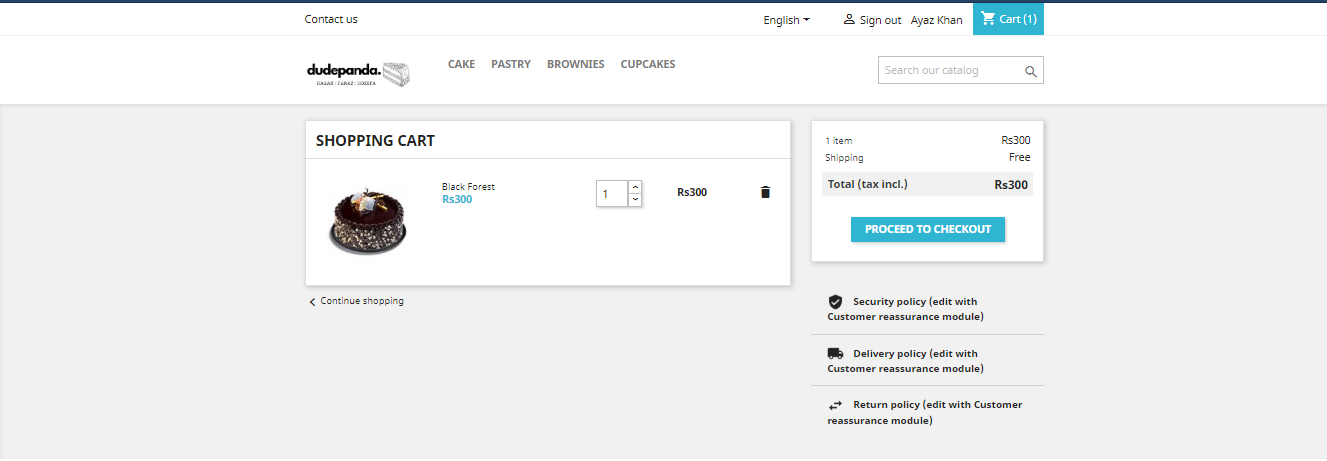
1. Next choose a cake that you want. I’ll be choosing *Black Forest* cuz why not
2. The following screen will then appear



Select the quantity and then press add to cart

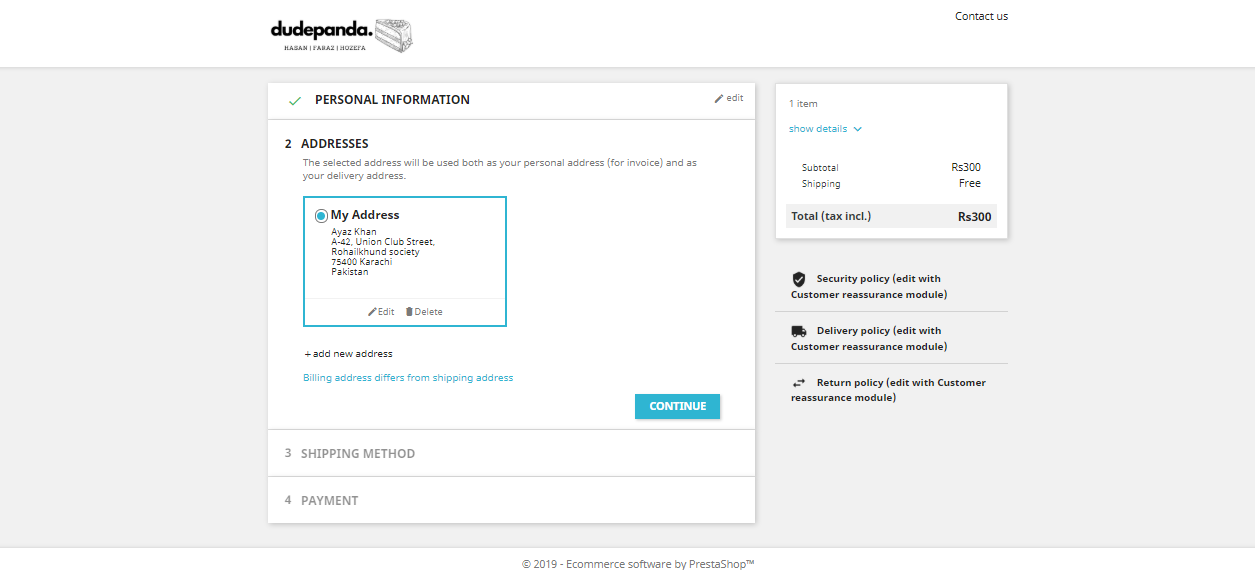
1. On doing so, this window will pop up



1. If that’s all what i want to order, click *Proceed to Checkout*
2. 

This window shows the order amount details (shipping cost if any)

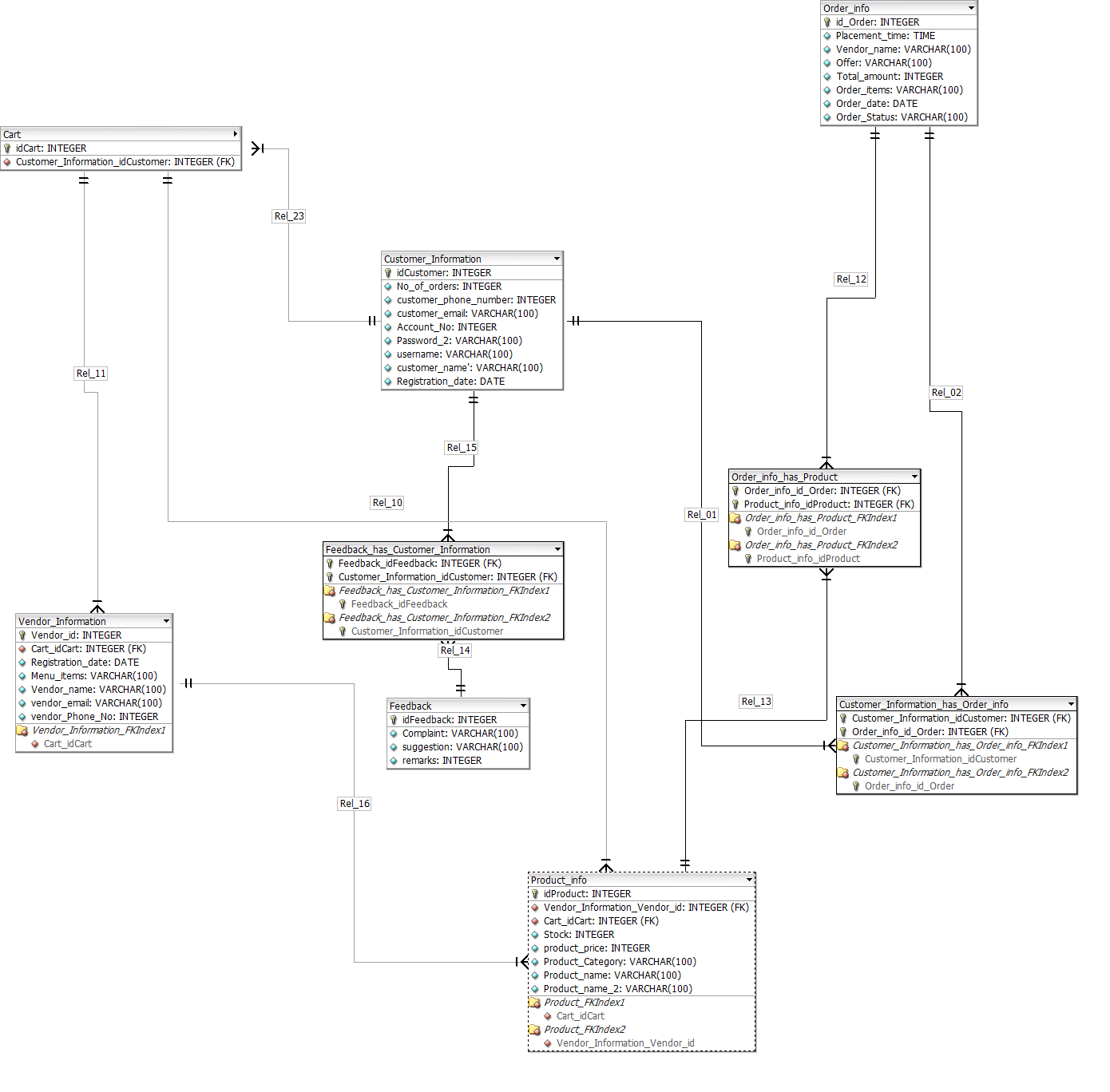
click *Proceed to Checkout*

1. 

Edit address and then press continue

**ERD;**

Following is a picture of our ERD;



The entities in this ERD are **Vendor Information, Customer Information, Product\_Info, Cart,Order\_Info, Feedback**

Following is the description of the entities and the relationships between them:

-**Vendor Information**

This table will contain the list of Vendors registered with the Company. Each Vendor has certain attributes such as Vendor name, phone no#, Registration date etc. Each Vendor is assigned a Vendor ID by the Company

- **Customer Information**

Customer info will contain the details of the users that have signed up on the website whether or not they use the application or not. This table contains the users account details (account# , password, username) and other details such as phone number, email address and of course the name of the Customer. Each Customer is assigned a Customer ID by the company at the time when the Customer creates an account

-**Product Information**

Product Information will display the Menu of the Different Vendors. Each Product is assigned a Product ID by the Company

**-Cart**

Each Customer will have its own Cart (and hence a seperate ID assigned by the Company) which will hold information about the order.

-**Order Information**

This table will contain the information about the details of a particular order placed by a customer, including its placement time. Each order will have a distinct ID assigned by the Company.

-**Feedback**

Customers will be reviewed about their experiences regarding their usage and satisfaction. This table will contain all that data

**Relationships:**

-**Vendor Information**

* 1:n relationship with Product\_info table since the vendor would have the information of the product he is giving to the company. Therefore, the vendor will have information for many products at any given time and many products could have the same vendor.
* M:n relationship with Cart since 1 vendor information could be in many different carts and 1 cart could have many different vendors in it.

-**Customer Information**

* M:n relationship with Order\_info table since the customer should be aware of what he is ordering and what is the price and quantity of what he is ordering. Hence, a customer could have many orders and vice versa.
* M:n relationship with Feedback table since the customer may have many feedbacks and same feedback may also be given by many customers.
* 1:n relationship with cart since 1 customer could have many carts in a day if he orders multiple times and many carts could be of the same customer as well.

-**Product Information**

* M:n relationship with Sale\_info table, single product could have many sales and similarly 1 sale could be of many products.
* M:n relationship with Order\_info table, 1 product could have many orders and similarly 1 orders could be of many different products.
* M:n relationship with Cart since many products could be in a single cart and many carts could have one product as well.

- **Order Information**

* M:n relationship with Customer Information since one order could be placed by many customers and one customer could place many orders as well.
* M:n relationship with Sale\_info table since one order could have many sales and one sale could be of many orders.
* M:n relationship with Product\_info since one order could have many products and many products could be in one order.
* M:n relationship with Cart since 1 cart could have many orders in it and many carts could contain the same order.

-**Feedback**

* M:n relationship with Customer information since 1 customer could have many feedbacks and 1 feedback could be given by many customers as well.

-**Cart**

* M:n relationship with Product\_info table since many products could be in a single cart and many carts could have one product as well.
* M:n relationship with Vendor\_info since 1 vendor information could be in many different carts and 1 cart could have many different vendors in it.
* M:n relationship with Order\_info since 1 cart could have many orders in it and many carts could contain the same order.

**Environment For Database:**

The environment used for Database is Prestashop, which is an open source E-commerce solution, which can be used to run stores in the cloud or via self-hosting. It is currently used by 250,000 shops and is available in 65 languages. Prestashop is fairly easy to use and provides a powerfully responsive store interface for shoppers, offers a comprehensive set of features for free.

**Setup for Prestashop:**

* **What you need to get started:**

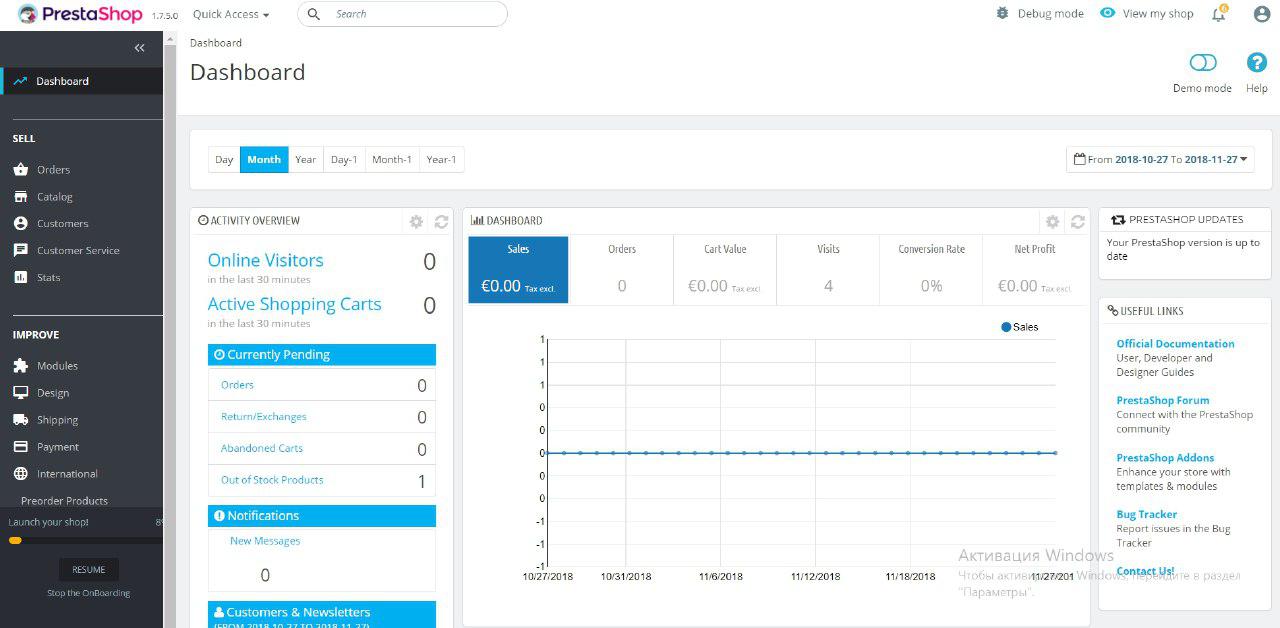
Here is a list of things which are required before you proceed to download Prestashop.

* System requirements:
  + PHP 5.4 or later.
    - Useful settings (in the php.ini file):
      * allow\_url\_fopen set to On,
      * register\_globals set to Off,
      * upload\_max\_filesize set to "16M" (or more).
    - Must-have PHP extensions (in the php.ini file): PDO\_MySQL, cURL, SimpleXML, mcrypt, GD, OpenSSL, DOM, SOAP, Zip, fileinfo.
    - Useful server tools: cron/crontab, Memcached.
  + MySQL 5.0 or later.
  + Better if:
    - Unix/Linux hosting.
    - Apache Web Server 2.0 or later or nginx Web Server.
      * Apache module settings:
        + mod\_rewrite enabled,
        + mod\_security disabled,
        + mod\_auth\_basic disabled.
    - At least 128 Mb of RAM dedicated to PHP. The more the better.
* Access codes to your FTP server, your MySQL database
  + These should be provided by your web host if you are not doing a local installation.
* Any text editor.
* Any FTP client.
* Any modern Web browser (if using Internet Explorer: at least IE9).

(taken from <http://doc.prestashop.com/display/PS17/What+you+need+to+get+started>)

* **Prestashop Setup:**
  + Download zip file for Prestashop version 1.7.6.2, which is available on its website, i.e. <https://www.prestashop.com/en/download>.
  + Use Xampp, which is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in PHP and Perl programming languages. After installing Xampp, we will start Apache and MySQL module. Now, our local host is available for use.
  + Now, when you type <http://localhost/phpmyadmin> on your web browser, it will open your local host database. Create a new database with name, “prestashop” from the Database tab.
  + Now, when you type <http://localhost/prestashop> on your web browser, it will start the installation setup for Prestashop.
  + Follow the steps and input details as required.
  + During the steps, when it asks about the database\_host, database\_name, database\_user and database\_password, type the details of the database you created on phpmyadmin.
  + As soon as you complete the steps, it will give you the generated links for both, prestashop customer hand and prestashop admin panel.
  + You can edit the shop according to your product through the admin panel, such as logo, products, themes, carousel, pictures etc.
  + You can also manage the orders, shipping, sales and over all performance of the store through the admin panel.

**- Prestashop Admin Panel:**



**SQL Queries of ERD:**

CREATE TABLE Customer\_Information (

idCustomer INTEGER NOT NULL AUTO\_INCREMENT,

customer\_phone\_number INTEGER UNSIGNED NULL ,

customer\_email VARCHAR(100) NULL ,

Account\_No INTEGER UNSIGNED NULL ,

Password\_2 VARCHAR(100) NULL ,

username VARCHAR(100) NULL ,

customer\_name VARCHAR(100) NULL ,

Registration\_date DATE NULL ,

PRIMARY KEY(idCustomer));

CREATE TABLE Feedback (

idFeedback INTEGER UNSIGNED NOT NULL AUTO\_INCREMENT,

Complaint VARCHAR(100) NULL ,

suggestion VARCHAR(100) NULL ,

remarks INTEGER UNSIGNED NULL ,

PRIMARY KEY(idFeedback));

CREATE TABLE Vendor\_Information (

Vendor\_id INTEGER UNSIGNED NOT NULL AUTO\_INCREMENT,

Registration\_date DATE NULL ,

Menu\_items VARCHAR(100) NULL ,

Vendor\_name VARCHAR(100) NULL ,

vendor\_email VARCHAR(100) NULL ,

vendor\_Phone\_No INTEGER UNSIGNED NULL ,

Cart\_id INTEGER UNSIGNED NULL ,

PRIMARY KEY(Vendor\_id));

CREATE TABLE Order\_info (

id\_Order INTEGER UNSIGNED NOT NULL AUTO\_INCREMENT,

Customer\_Information\_idCustomer INTEGER NOT NULL ,

Placement\_time TIME NULL ,

Vendor\_name VARCHAR(100) NULL ,

Offer VARCHAR(100) NULL ,

Total\_amount INTEGER UNSIGNED NULL ,

Order\_items VARCHAR(100) NULL ,

Order\_date DATE NULL ,

Order\_Status VARCHAR(100) NULL ,

PRIMARY KEY(id\_Order) ,

INDEX Order\_info\_FKIndex1(Customer\_Information\_idCustomer),

FOREIGN KEY(Customer\_Information\_idCustomer)

REFERENCES Customer\_Information(idCustomer)

ON DELETE NO ACTION

ON UPDATE NO ACTION);

CREATE TABLE Product\_info (

Product\_id INTEGER UNSIGNED NOT NULL ,

Vendor\_Information\_Vendor\_id INTEGER UNSIGNED NOT NULL ,

Stock INTEGER UNSIGNED NULL ,

product\_price INTEGER UNSIGNED NULL ,

Product\_Category VARCHAR(100) NULL ,

Product\_name VARCHAR(100) NULL ,

Product\_name\_2 VARCHAR(100) NULL ,

PRIMARY KEY(Product\_id) ,

INDEX Product\_FKIndex2(Vendor\_Information\_Vendor\_id),

FOREIGN KEY(Vendor\_Information\_Vendor\_id)

REFERENCES Vendor\_Information(Vendor\_id)

ON DELETE NO ACTION

ON UPDATE NO ACTION);

CREATE TABLE Order\_info\_has\_Product\_info (

Order\_info\_id\_Order INTEGER UNSIGNED NOT NULL ,

Product\_info\_Product\_id INTEGER UNSIGNED NOT NULL ,

PRIMARY KEY(Order\_info\_id\_Order, Product\_info\_Product\_id) ,

INDEX Order\_info\_has\_Product\_info\_FKIndex1(Order\_info\_id\_Order) ,

INDEX Order\_info\_has\_Product\_info\_FKIndex2(Product\_info\_Product\_id),

FOREIGN KEY(Order\_info\_id\_Order)

REFERENCES Order\_info(id\_Order)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

FOREIGN KEY(Product\_info\_Product\_id)

REFERENCES Product\_info(Product\_id)

ON DELETE NO ACTION

ON UPDATE NO ACTION);

CREATE TABLE Cart (

idCart INTEGER UNSIGNED NOT NULL AUTO\_INCREMENT,

Customer\_Information\_idCustomer INTEGER NOT NULL ,

Vendor\_Information\_Vendor\_id INTEGER UNSIGNED NOT NULL ,

PRIMARY KEY(idCart) ,

INDEX Cart\_FKIndex2(Vendor\_Information\_Vendor\_id) ,

INDEX Cart\_FKIndex3(Customer\_Information\_idCustomer),

FOREIGN KEY(Vendor\_Information\_Vendor\_id)

REFERENCES Vendor\_Information(Vendor\_id)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

FOREIGN KEY(Customer\_Information\_idCustomer)

REFERENCES Customer\_Information(idCustomer)

ON DELETE NO ACTION

ON UPDATE NO ACTION);

CREATE TABLE Cart\_has\_Product\_info (

Cart\_idCart INTEGER UNSIGNED NOT NULL ,

Product\_info\_Product\_id INTEGER UNSIGNED NOT NULL ,

Quantity INTEGER UNSIGNED NULL ,

PRIMARY KEY(Cart\_idCart, Product\_info\_Product\_id) ,

INDEX Cart\_has\_Product\_info\_FKIndex1(Cart\_idCart) ,

INDEX Cart\_has\_Product\_info\_FKIndex2(Product\_info\_Product\_id),

FOREIGN KEY(Cart\_idCart)

REFERENCES Cart(idCart)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

FOREIGN KEY(Product\_info\_Product\_id)

REFERENCES Product\_info(Product\_id)

ON DELETE NO ACTION

ON UPDATE NO ACTION);

CREATE TABLE Feedback\_has\_Customer\_Information (

Feedback\_idFeedback INTEGER UNSIGNED NOT NULL ,

Customer\_Information\_idCustomer INTEGER NOT NULL ,

PRIMARY KEY(Feedback\_idFeedback, Customer\_Information\_idCustomer) ,

INDEX Feedback\_has\_Customer\_Information\_FKIndex1(Feedback\_idFeedback) ,

INDEX Feedback\_has\_Customer\_Information\_FKIndex2(Customer\_Information\_idCustomer),

FOREIGN KEY(Feedback\_idFeedback)

REFERENCES Feedback(idFeedback)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

FOREIGN KEY(Customer\_Information\_idCustomer)

REFERENCES Customer\_Information(idCustomer)

ON DELETE NO ACTION

ON UPDATE NO ACTION);

**SQL QUERIES:**

* **To insert data:**

Just press the insert button, the query will appear itself and just plug in the values