**ONLINE RETAIL MANAGEMENT SYSTEM**

**by**

**Nikesh Reddy S 13BCE1136**

**Shashank D 13BCE1036**

**Gourav K 13BCE1068**

*A project report submitted to the*

**SCHOOL OF COMPUTING SCIENCES ENGINEERING**



**VIT UNIVERSITY, CHENNAI**

**Vandalur – Kelambakkam Road**

**Chennai – 600 127.**

**MAY 2016**

**ACKNOWLEDGEMENT**

**Course Code: CSE325**

This project report entitled “**ONLINE RETAIL MANAGEMENT SYSTEM**” is a bonafide work of **SHASHANK. D (13BCE1036), NIKESH. S (13BCE1136) and KOUTIKA GOURAV (13BCE1068)** carried out the Project work under the supervision and guidance of **Prof. ALOK CHAUHAN**. We wish to express our sincere thanks and deep sense of gratitude to our project guide, **Prof. ALOK CHAUHAN**, Professor, School of Computing Sciences and Engineering, for his consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work.

**TABLE OF CONTENTS**

1. Abstract
2. Introduction
3. SQL
4. ER Diagram
5. Modules
6. Outcome
7. Use-case Diagram
8. Modular Description
9. Sequence diagrams

**ABSTRACT**

Online Retail Management Software is a software application to be developed for the use in any retail unit/shop. The software aims at computerizing all the activities related to the retail unit. It should be a comprehensive one to cover all the aspect of a retail shop.

The application has 13 modules consisting of several sub modules each:

**Module 1:** Supplier/Creditor Detail

**Module 2:** Customer/Debtor Detail

**Module 3:** Item Details

**Module 4:** Purchase

**Module 5:** Purchase Return

**Module 6:** Stock Maintenance

**Module 7:** Order Processing

**Module 8:** Sales

**Module 9:** Sales Return

**Module 10:** Accounts

**Module 11:** Query

**Module 12:** Reports

**Module 13:** Feedback

**INTRODUCTION**

Online Retail Management System is a form of electronic shopping store where the buyer is directly online to the seller’s computer usually via the internet. There is no intermediary service. The sale and purchase transaction is completed electronically and interactively in real-time. The development of this new system contains the following activities, which try to develop on-line application by keeping the entire process in the view of database integration approach. User gets its email id and password to access their account.

**Administrator** of Online Retail Management System has multiple features such as Add,Delete, Update shopping Items.

**Customer** of Online Retail Management System has multiple features such as Sign up,Login, Shop online, Get an invoice.

**Features of Online Retail Management System:**

1. Secure registration and profile management facilities for Customers.
2. Browsing through the e-Mall to see the items that are there in each category of products like Apparel, Kitchen accessories,

Bath accessories, Food items etc.

1. Creating a Shopping cart so that customer can Shop N number of items and checkout finally with the entire shopping cart

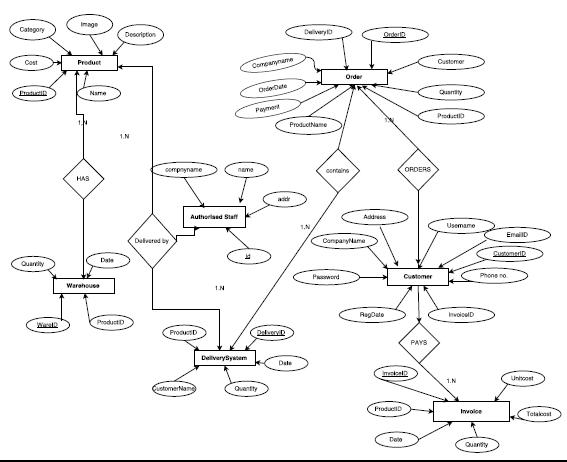
1. Customers should be able to mail the Shop about the items they would like to see in the Shop
2. Secured mechanism for checking out from the Shop ( Credit card verification mechanism ). Updates to customers about the Recent Items in the Shop.
3. Uploading Most Purchased Items in each category of products in the Shop like Apparel, Kitchen accessories, Bath accessories,

Food items etc.

**SQL**

SQL stands for Structured Query Language. SQL is used to communicate with a database. According to ANSI (American National Standards Institute), it is the standard language for relational database management systems. SQL statements are used to perform tasks such as update data on a database, or retrieve data from a database. Some common relational database management systems that use SQL are: Oracle, Sybase, Microsoft SQL Server, Access, Ingres, etc. Although most database systems use SQL, most of them also have their own additional proprietary extensions that are usually only used on their system. However, the standard SQL commands such as "Select", "Insert", "Update", "Delete", "Create", and "Drop" can be used to accomplish almost everything that one needs to do with a database. This tutorial will provide you with the instruction on the basics of each of these commands as well as allow you to put them to practice using the SQL Interpreter.

**ER DIAGRAM**



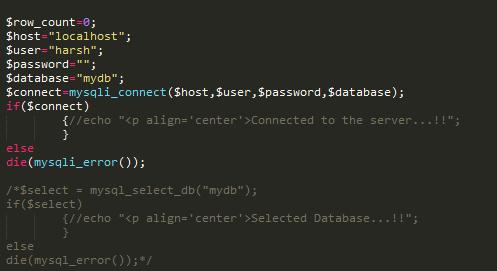
**MODULES**

Our project consists of several modules namely,

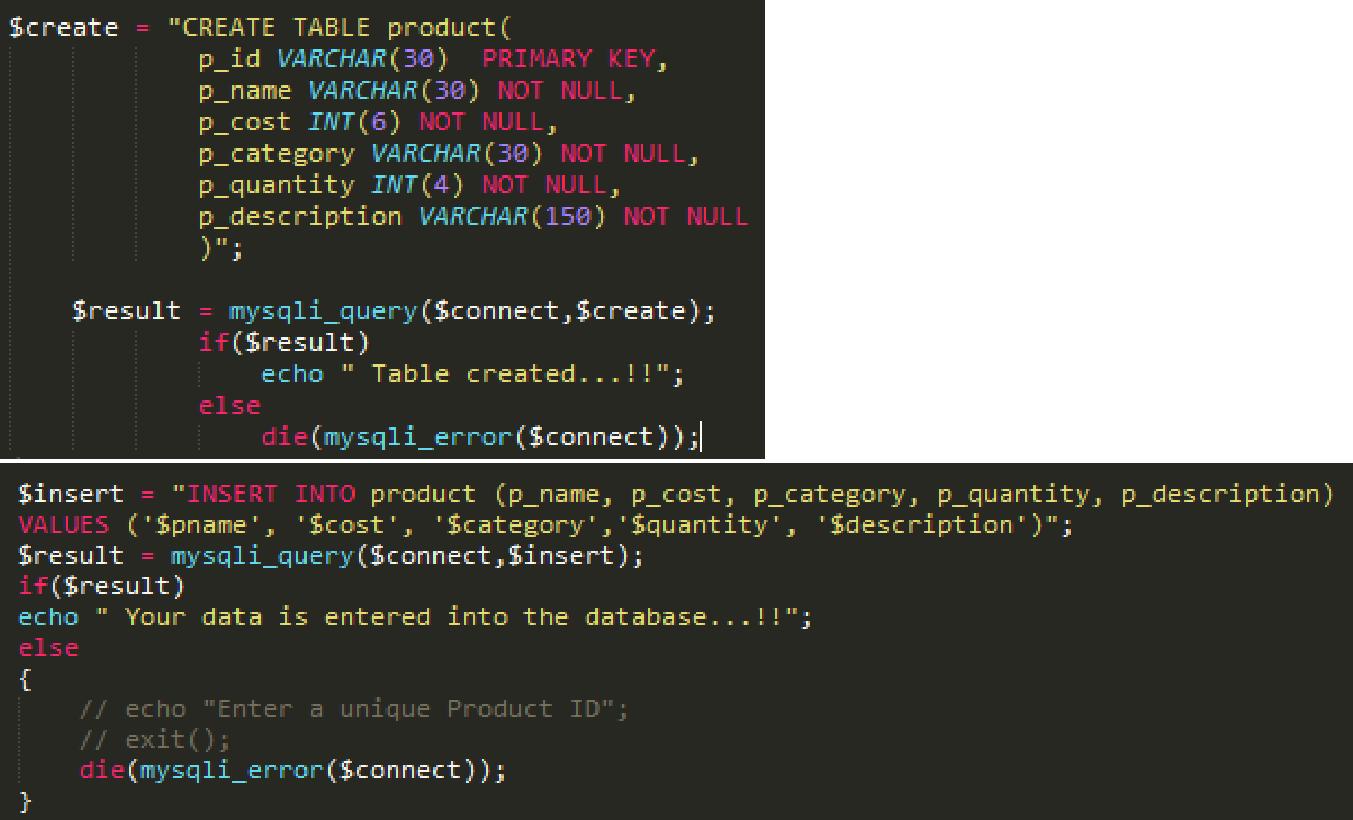
* Administrator
  + Add a product
  + Modify a product
  + Delete a product
  + Search a product
* Customers
  + Login
  + Sign up
  + Shop products
  + Generate an invoice

Now we are going to see the various queries used in building these modules individually.

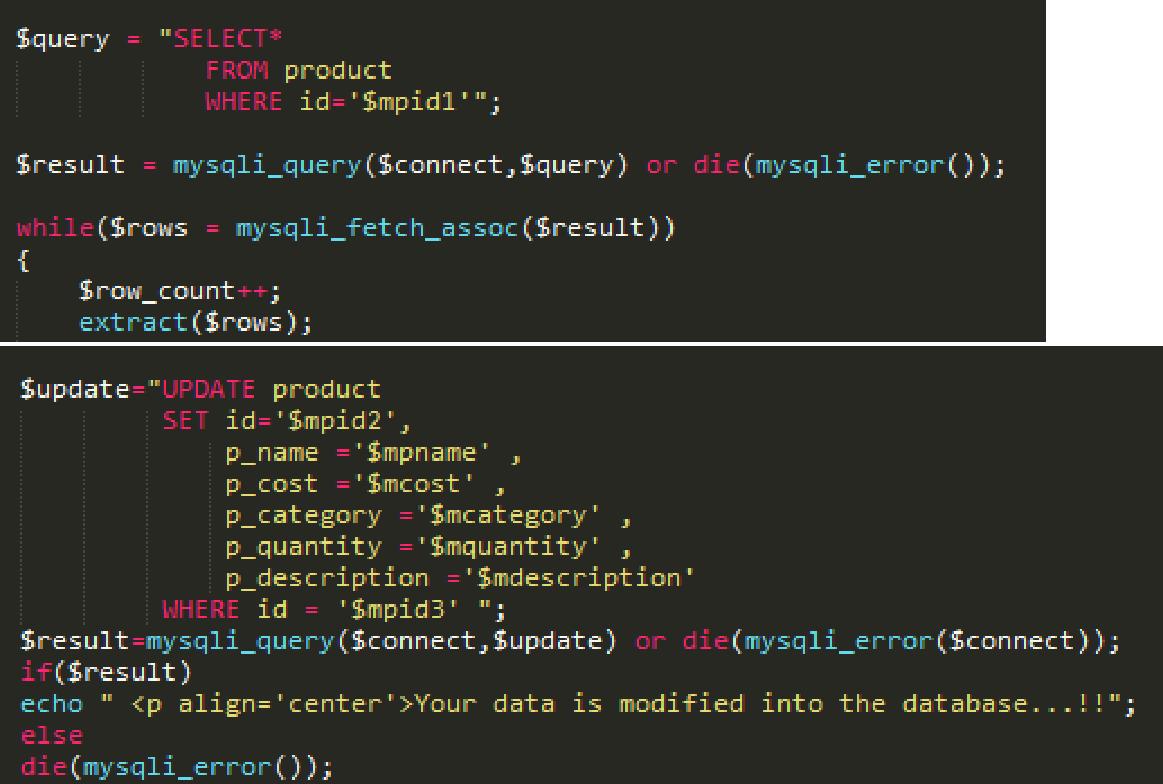
**ESTABLISH CONNECTION WITH THE DATABASE:**



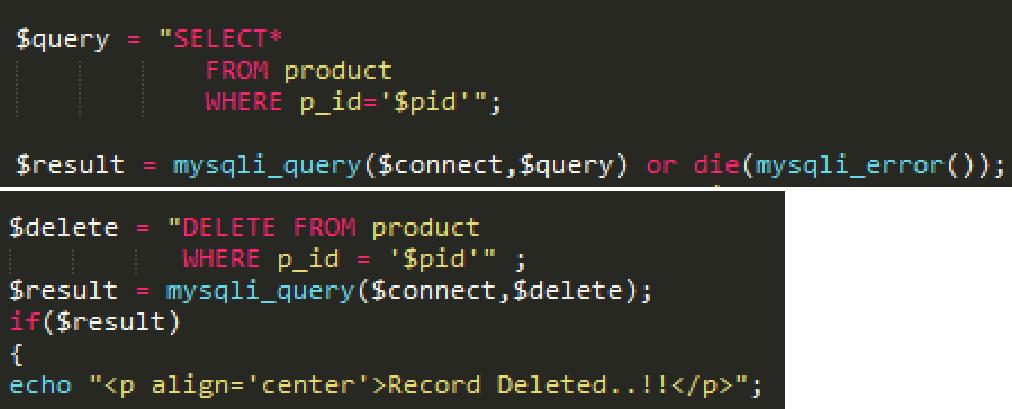
**ADMINISTRATOR: 1) Add a product:**



**2) Modify a product**



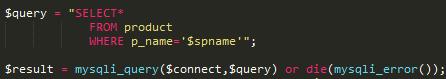
**3) Delete a product**



**4) Search a product**

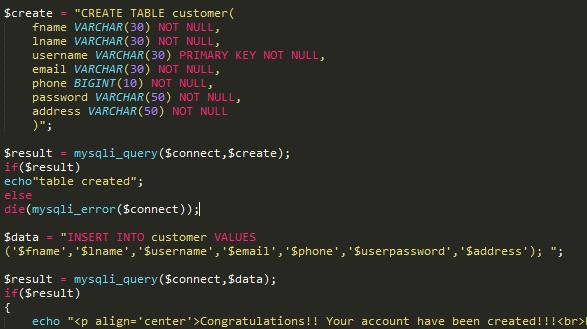
Searching is done based on productID,product Name,productCategory etc.

For example, if searching is done based on product name then the query would be:

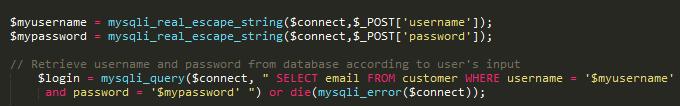


**CUSTOMERS**

**1) Sign up**



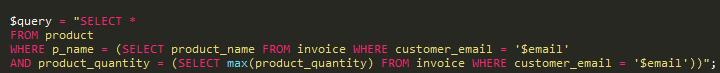
1. **Login**
2. **Shop products**



Displaying products category wise



**Also “Recommendations” have been provided based on the past purchases of a particular customer**:

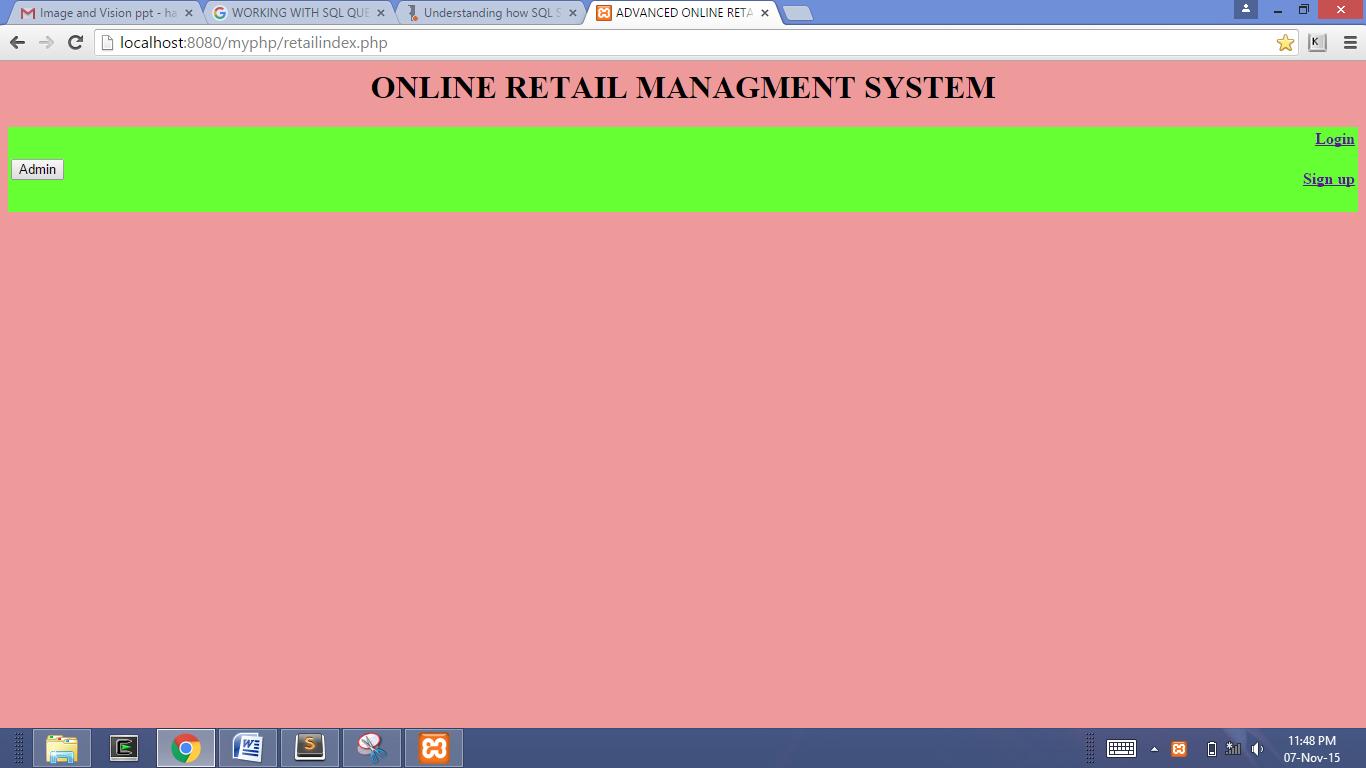


**4) Generate an invoice**

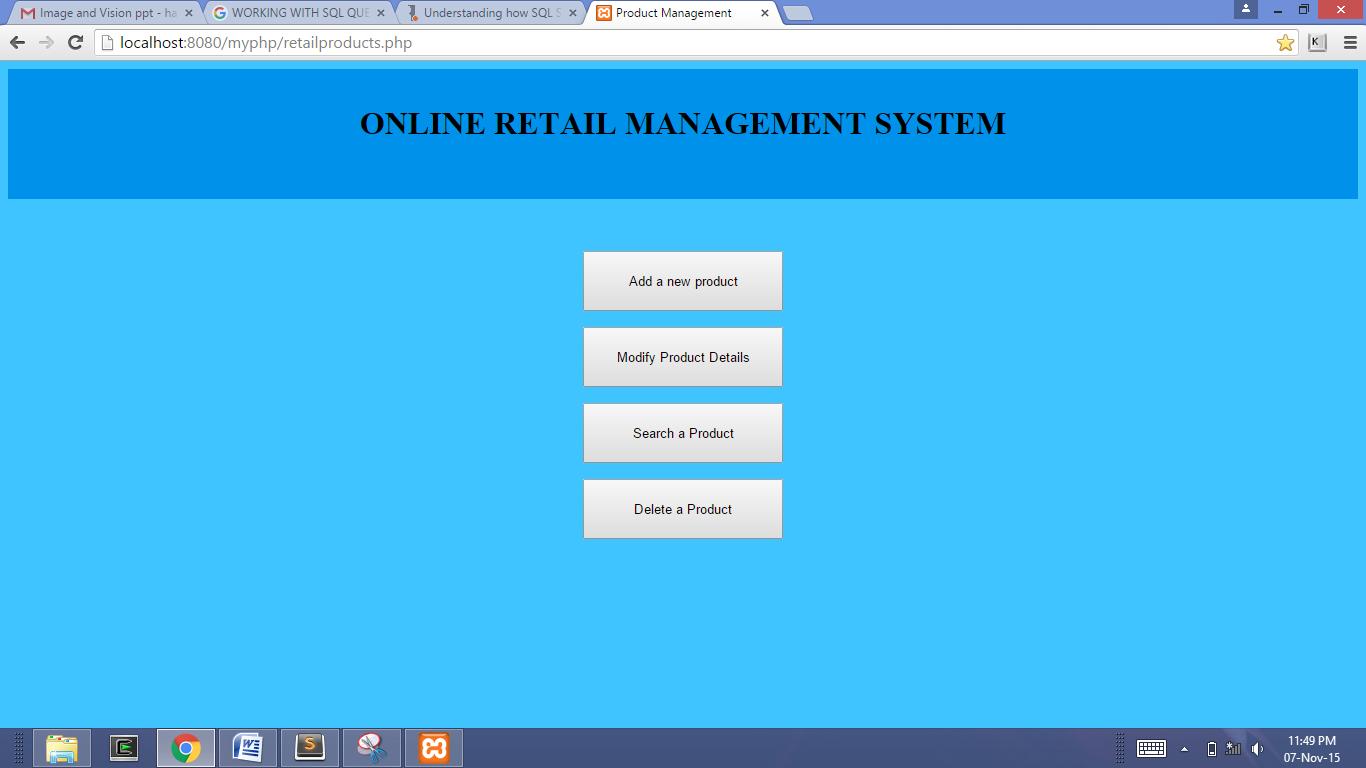


**OUTCOME**

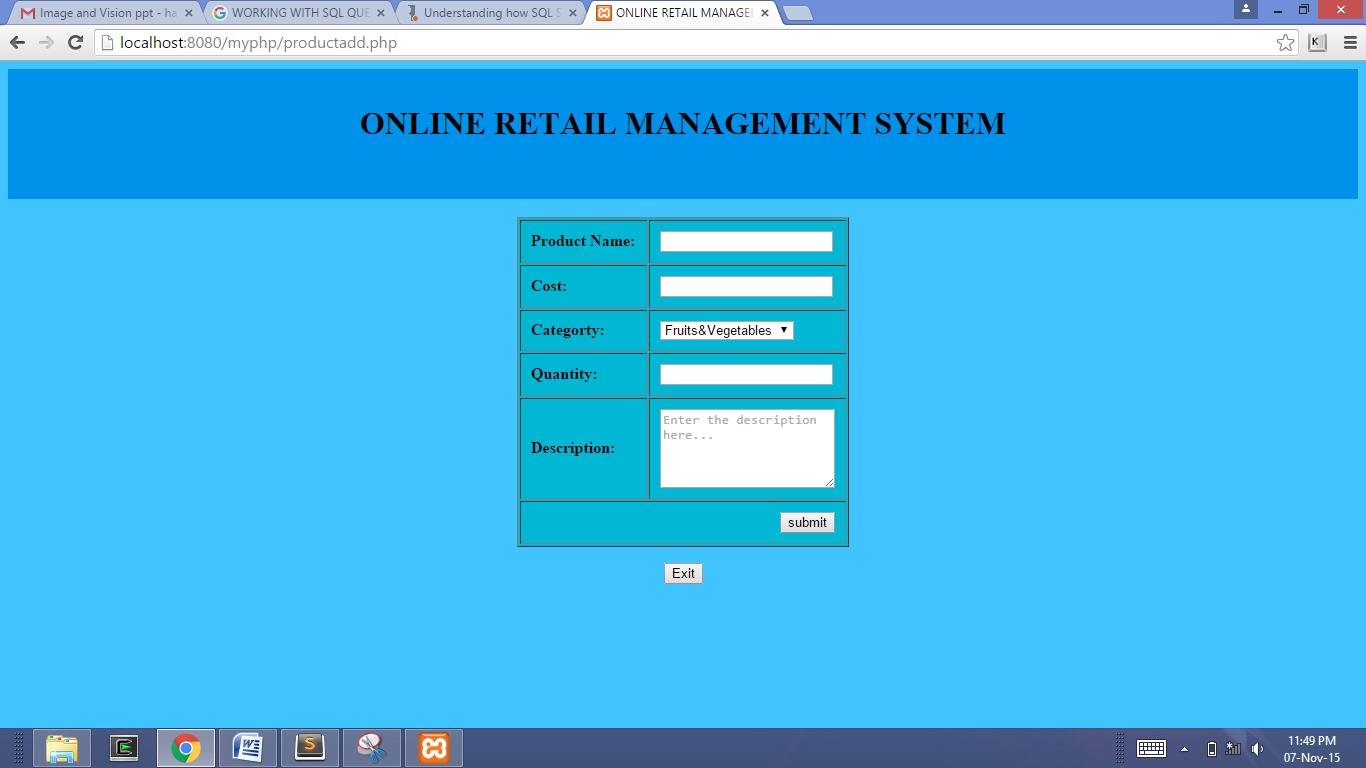
**Index Page:**



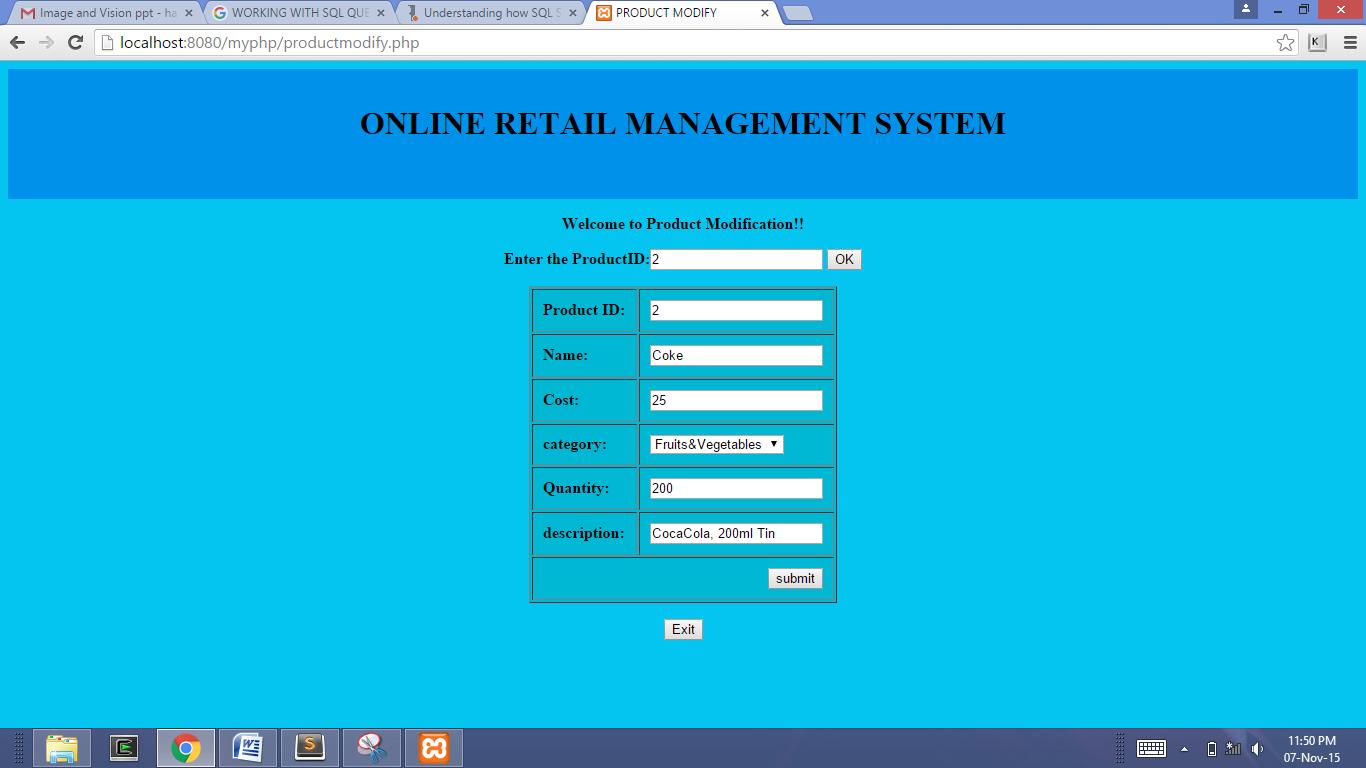
**Admin Portal:**



**Add a product:**



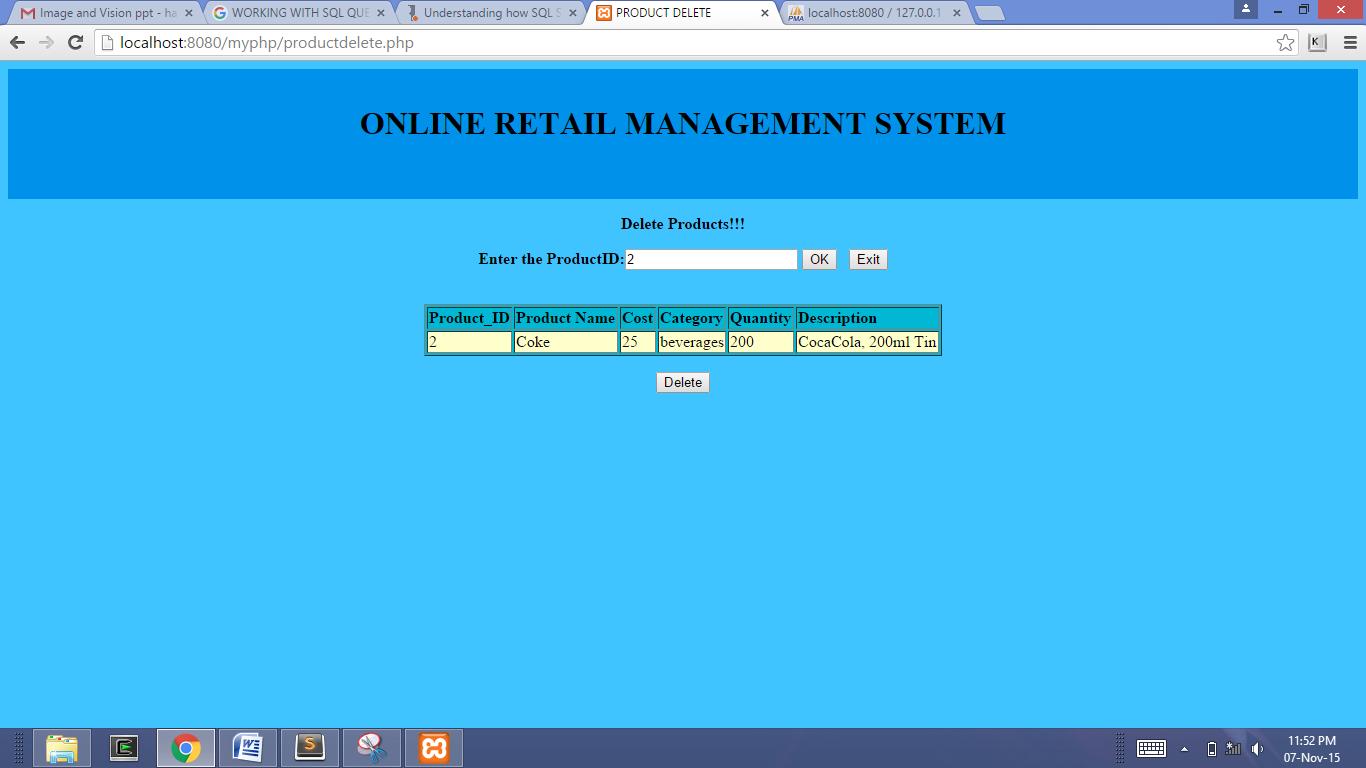
**Modify a product:**



**Search a product:**

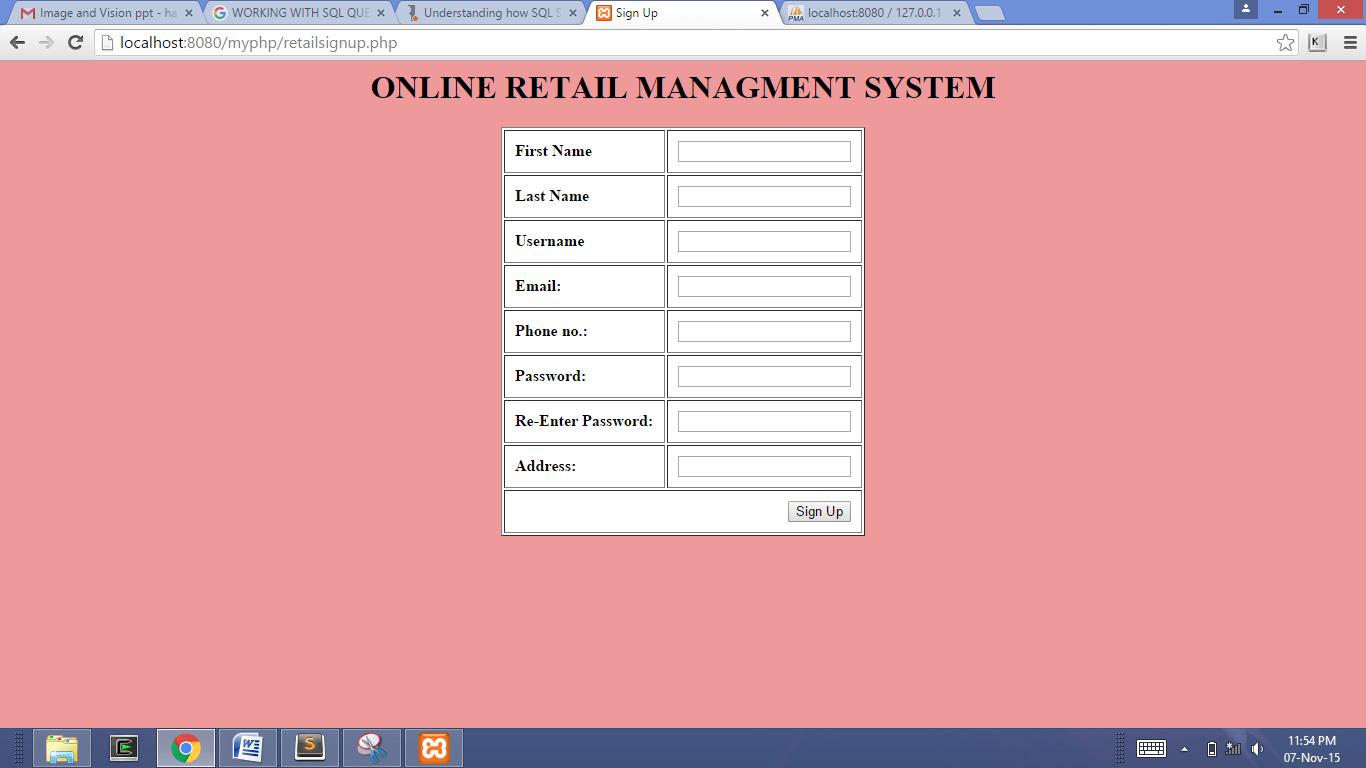


**Delete a product:**

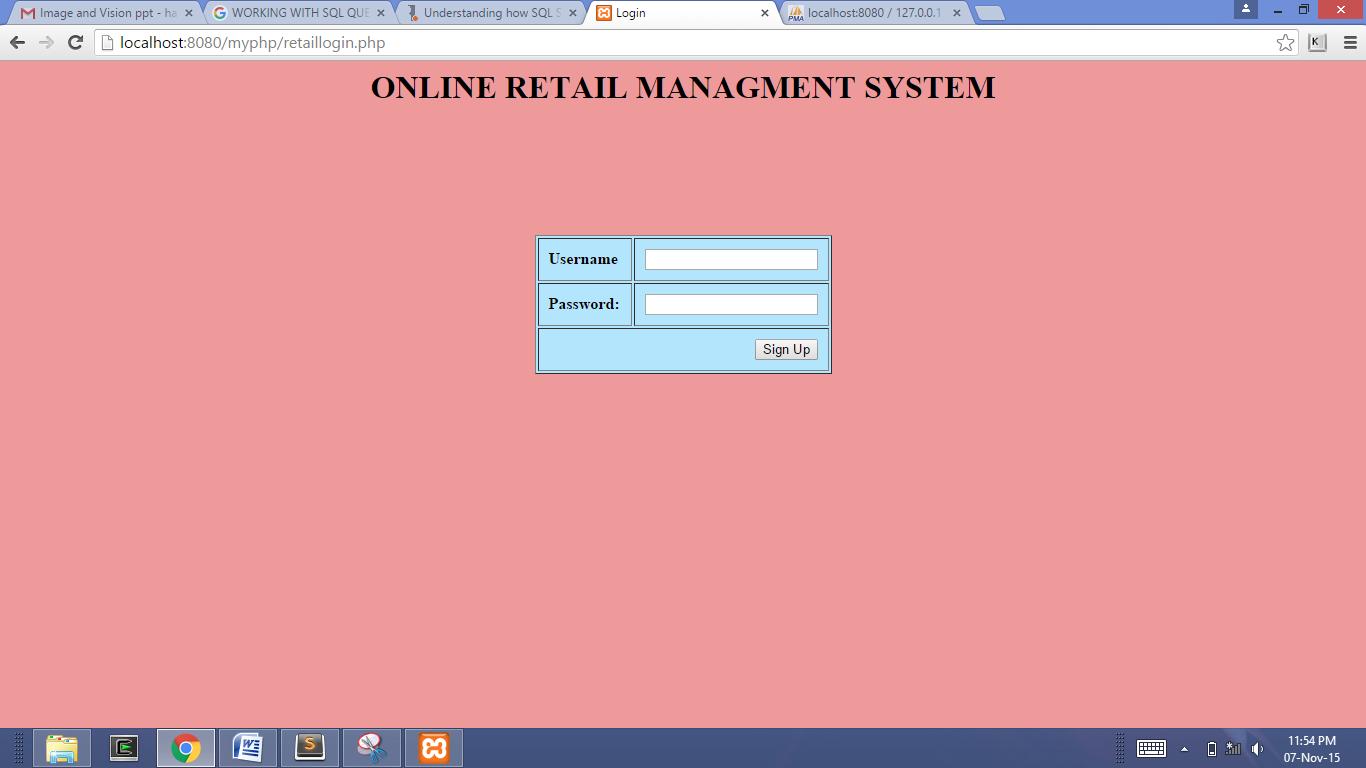


**CUSTOMERS:**

 **Sign up**



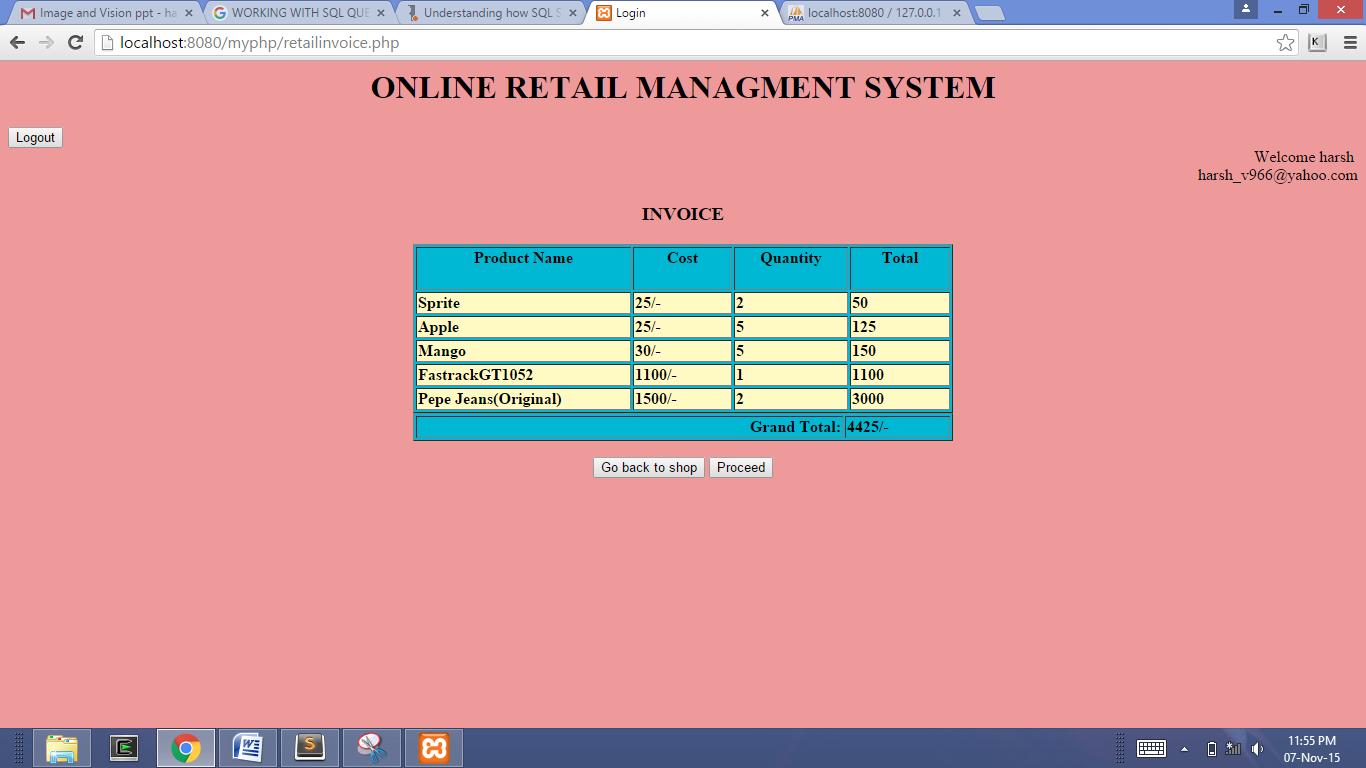
 **Login**

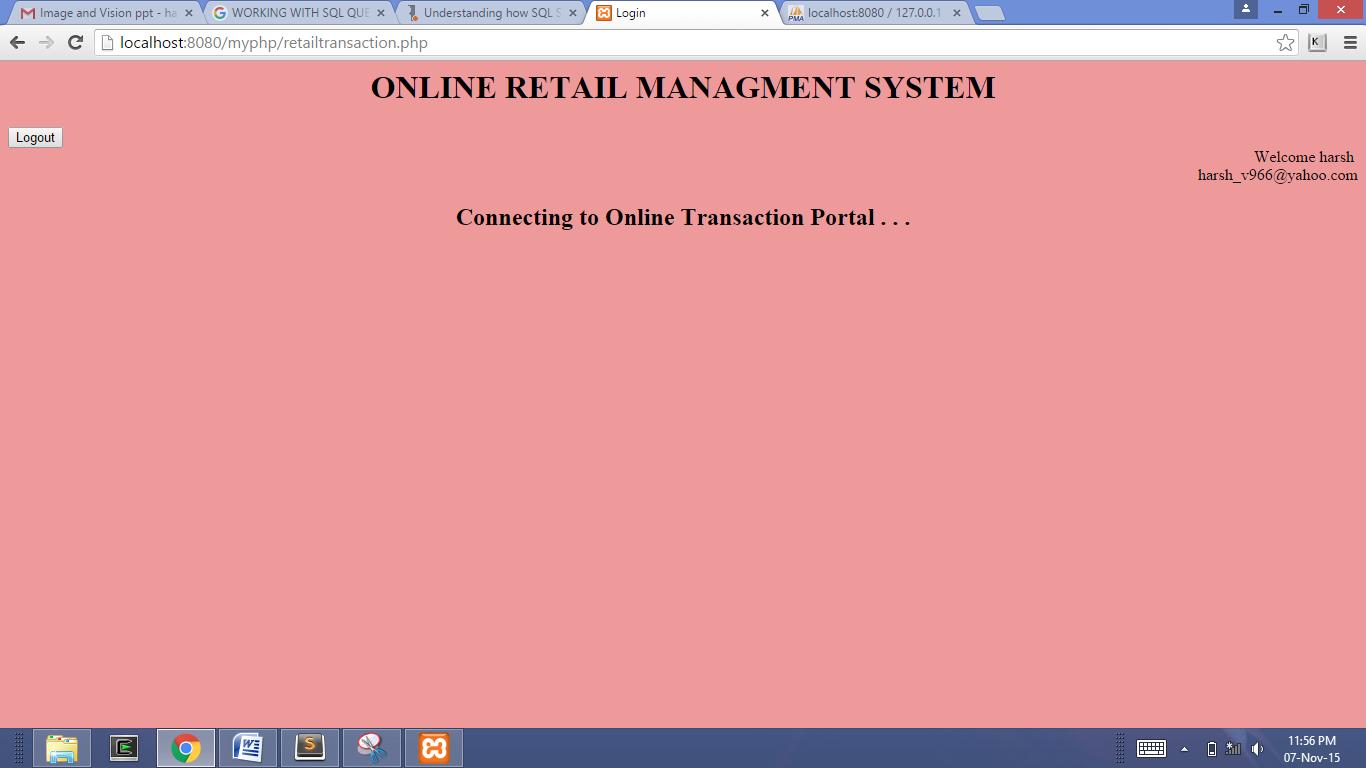


 **Product shop**

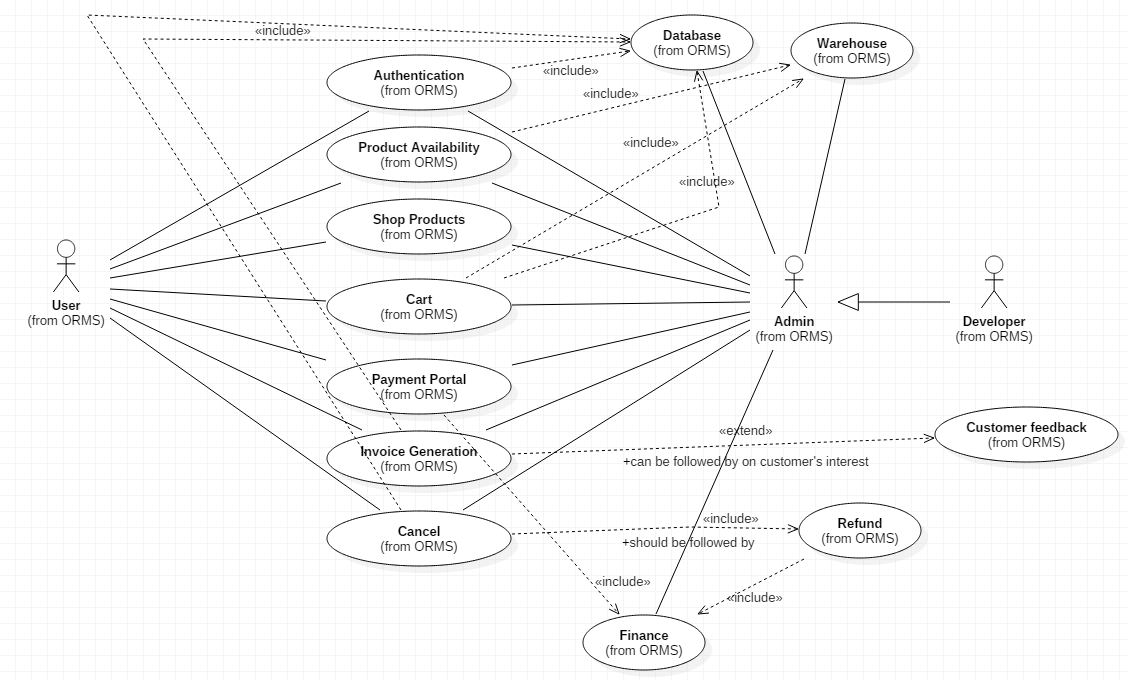


 **Invoice**





**USE CASE DIAGRAM**



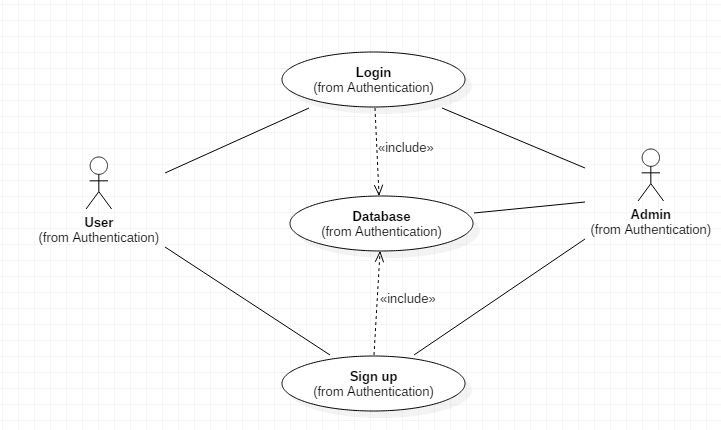
|  |  |  |
| --- | --- | --- |
| **Use-case** | **Pre-condition** | **Post-condition** |
| Authentication | In order to purchase products from ORMS, users need to have an online account. So, user should either have credentials to login into a secure ORMS account or create one. | Users would be authenticated with respect to their credentials and will be logged into their accounts providing access to purchase products. |
| Product Availability | Search of products by product name or type in order to be purchased can be reviewed by the users. | Users would find their desired products if available in stock in the warehouse and would be notified if the product is available but out of stock and would be denied of showcase if no such product of the user’s desire is available. |
| Shop Products | Required products can be placed as an order either in bulk or as a single product. | Order would be placed or added to cart based on the user’s confirmation or action performed. |
| Cart | ORMS provides its users an opportunity to place a single order for multiple products using the cart option for which the user needs to add products to cart and place order. | The products added to the cart will be placed on user’s confirmation and may also be deleted from the cart on user’s actions. |
| Payment Portal | The user should have placed an order of few items and would like to make the payment online. | On validation of the account details of the user, payment transfer from the user’s account to the credit of the official ORMS account. |
| Invoice Generation | The user should have ordered few products and the payment should have been done either online or at the time of delivery. | The user would possess an online and also an offline invoice stating the transaction details, order details and the customer details for what the user have purchased. |
| Cancel | The user must have an order placed either with a transaction done to purchase products from the site and possess an invoice with him/her or processed a request of products as cash on delivery. | The transaction/order is cancelled and the refund would be processed if the payment is made or only the order is cancelled if payment is not made. |
| Database |  |  |
| Warehouse |  |  |
| Finance |  |  |

**MODULAR DESCRIPTION**

|  |  |  |
| --- | --- | --- |
| **Use-case** | **Actors involved** | **Description** |
| Authentication | User  Admin | In order to purchase products from ORMS, users need to have an online account. So, user should either have credentials to login into a secure ORMS account or create one. Each and every user’s details will be stored in the database maintained by the ORMS and data will be retrieved and added when needed. Hence database is an include for the use-case. |
| Product Availability | User  Admin | Search of products by product name or type in order to be purchased can be reviewed by the users. Product availability will be checked with the warehouse/stock of ORMS. Hence, warehouse is an include. |
| Shop Products | User  Admin | Required products can be placed as an order either in bulk or as a single product. |
| Cart | User  Admin | ORMS provides its users an opportunity to place a single order for multiple products using the cart option for which the user needs to add products to cart and place order. Each and every time a product is added to a cart, it means that the demand of the product is increased by a certain value. It will be checked with the warehouse and if the number of products available are less than a certain threshold, necessary measures will be taken. User’s cart will be updated in the database. Hence database and warehouse are both includes for the use-case cart. |
| Payment Portal | User  Admin | The user should have placed an order of few items and would like to make the payment online. Debit from the user’s account and credit to the ORMS account will be handled by the finance of ORMS. Thus, finance of ORMS is an include. |
| Invoice Generation | User  Admin | The user should have ordered few products and the payment should have been done either online or at the time of delivery. Once the invoice is generated, the transaction is done and hence in order to enhance the facilities and also for awareness, ORMS requests for customer feedback which is of course optional which explains why Customer feedback is an extend. |
| Cancel | User  Admin | The user must have an order placed either with a transaction done to purchase products from the site and possess an invoice with him/her or processed a request of products as cash on delivery. On cancellation, the order details are deleted from the database and also a refund as debit from the ORMS account to the credit of user’s account from the finance will be initiated if payment is done. Hence, database and finance are includes. |
| Refund | Admin | Refund will be initiated from the ORMS account to the user’s account if an order is cancelled and the user has made the payment. |
| Database | Admin | The use case has nothing to do with the user but maintains the records of the users and admin and also the product details simply all data. |
| Warehouse | Admin | Represents the warehouse/stock of the ORMS products available. |
| Finance | Admin | Represents and handles the finance operations of ORMS. |

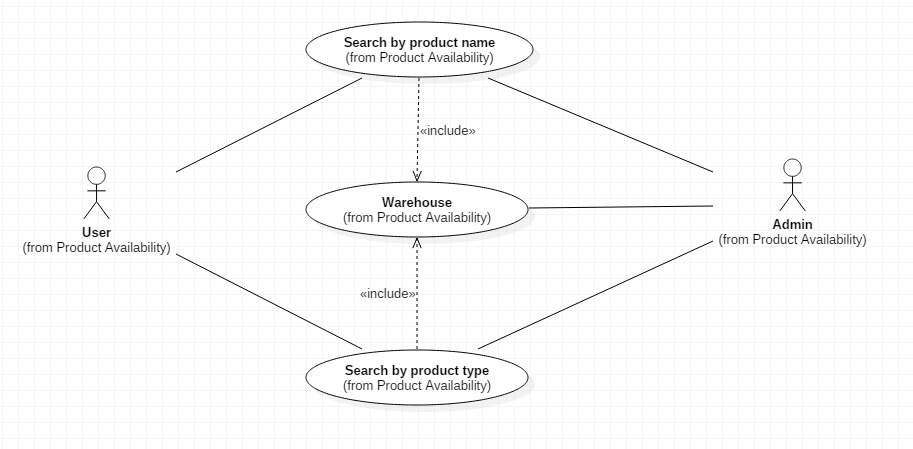
**SUB USE-CASE DIAGRAMS**

**AUTHENTICATION:**

****

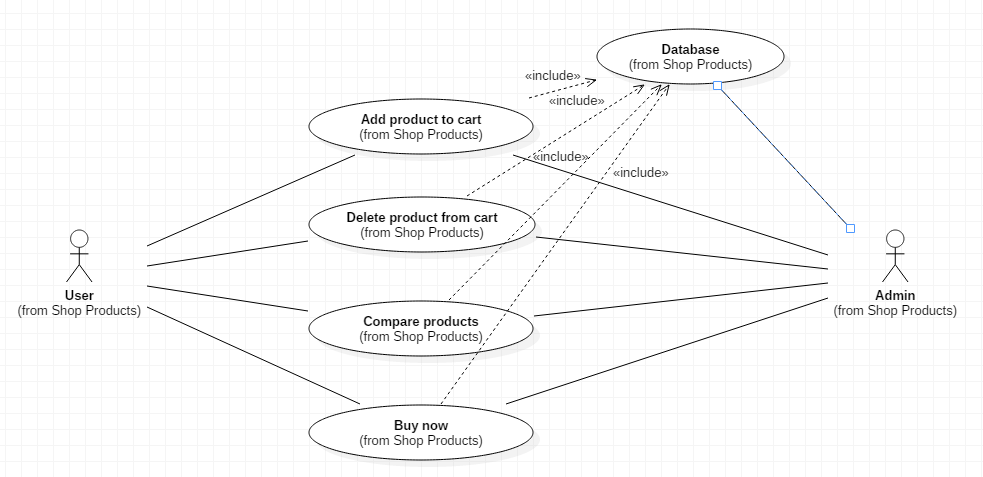
|  |  |  |  |
| --- | --- | --- | --- |
| **Use-case** | **Pre-condition** | **Post-condition** | **Actors involved** |
| Login | Need to have necessary credentials to login into an account. | Users will be logged into their account and will be granted access to purchase products from the site. | User  Admin |
| Sign-Up | - | Users create their account in ORMS and gets their details stored in the database. | User  Admin |
| Database |  |  | Admin |

**PRODUCT AVAILABILITY:**

****

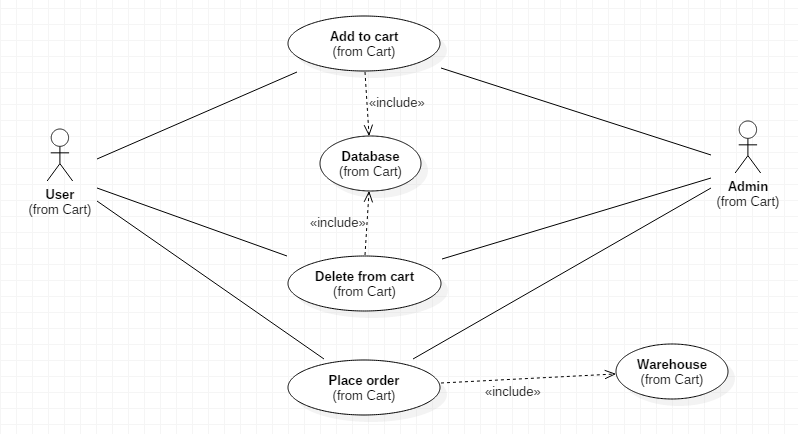
|  |  |  |  |
| --- | --- | --- | --- |
| **Use-case** | **Pre-condition** | **Post-condition** | **Actors involved** |
| Search by Product Name | Need to have name of the product to search. | Users will be displayed their desired products if available in the warehouse. | User  Admin |
| Search by Product Type | Need to have type of the product to search. | Users will be displayed their desired products if available in the warehouse. | User  Admin |
| Warehouse |  |  | Admin |

**SHOP PRODUCTS:**

****

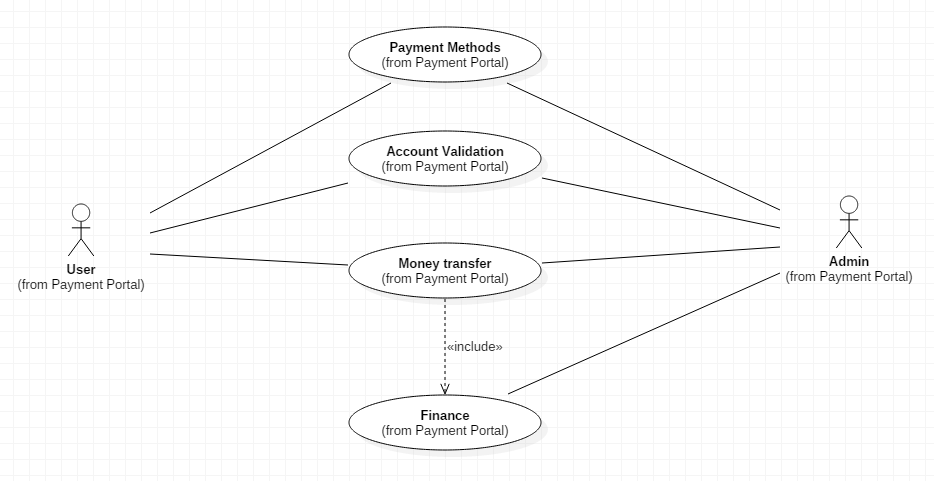
|  |  |  |  |
| --- | --- | --- | --- |
| **Use-case** | **Pre-condition** | **Post-condition** | **Actors involved** |
| Add product to cart. | Need to have the product to add to the cart. | Users will find their desired products added to the cart. | User  Admin |
| Delete product from cart. | Need to have the product in cart to delete. | Users will find their undesired products deleted from the cart. | User  Admin |
| Compare products | Need to have products required to compare. | The details of the products are fetched from the database and are displayed as a comparison. | User  Admin |
| Buy now | Users who want to buy single product need to select their product and click the buy now button. | Users will be redirected to the payment portal to buy their desired product. | User  Admin |
| Database |  |  | Admin |

**CART:**

****

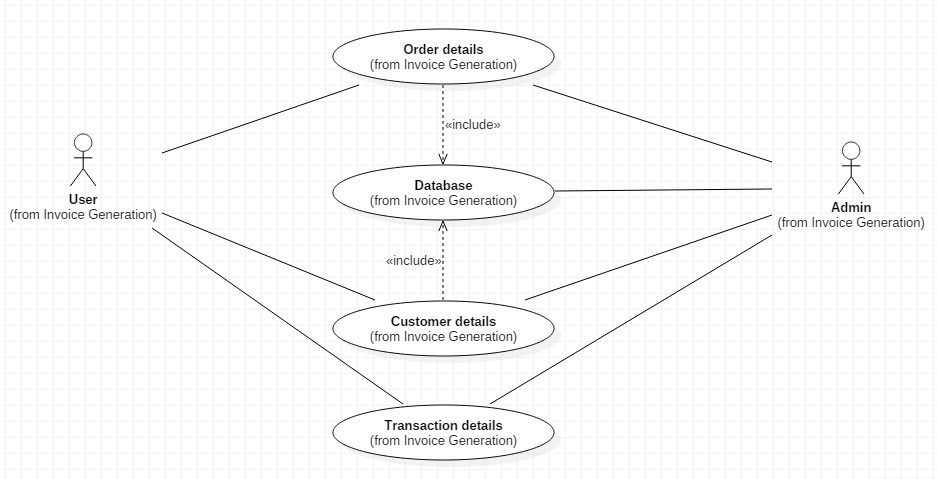
|  |  |  |  |
| --- | --- | --- | --- |
| **Use-case** | **Pre-condition** | **Post-condition** | **Actors involved** |
| Add to cart. | Need to have the product to add to the cart. | Users will find their desired products added to the cart. | User  Admin |
| Delete from cart. | Need to have the product in cart to delete. | Users will find their undesired products deleted from the cart. | User  Admin |
| Place order. | Need to have products in cart to place order. | The order will be placed and users will be redirected to the payment portal. | User  Admin |
| Database |  |  | Admin |

**PAYMENT PORTAL:**

****

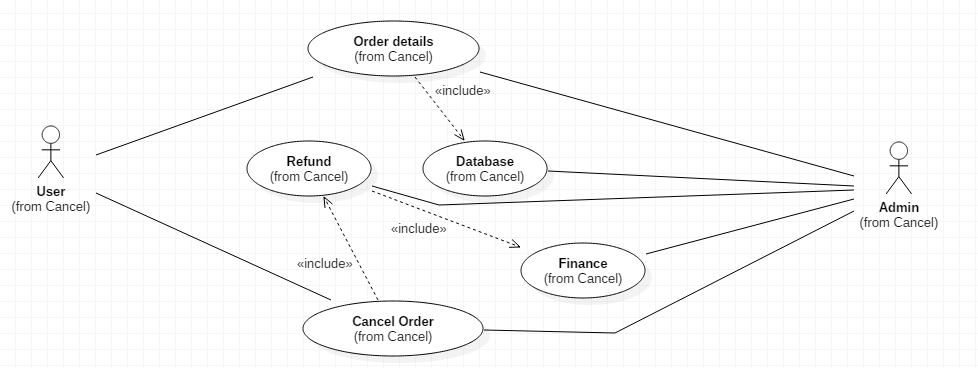
|  |  |  |  |
| --- | --- | --- | --- |
| Payment methods | User can be prepared to pay so that they can select one of the payment methods or can select cash on delivery. | Users will be redirected to respective payment gateways based on their choice. | User  Admin |
| Account validation | In order to pay online, users should have a bank account and the respective credentials of the account. | Users’ accounts will be validated for smooth money transfer. | User  Admin |
| Money transfer | Bank account of the user must be validated. | Money is transferred as a debit from the user’s account and credit to the ORMS account. | User  Admin |
| Finance |  |  | Admin |

**INVOICE GENERATION:**

****

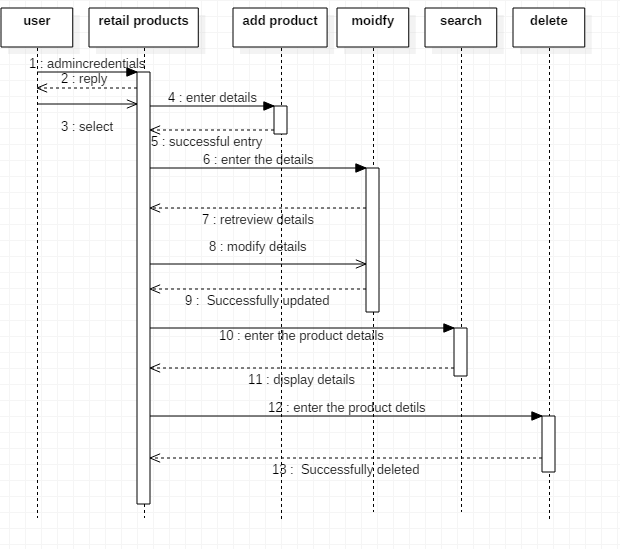
|  |  |  |  |
| --- | --- | --- | --- |
| Order details | User must have placed an order | Users order details will be fetched from the database and be displayed to the user. | Admin |
| Customer details | User should have an account | Users’ accounts’ details will be fetched from the database and be displayed to the user. | Admin |
| Transaction details | Users must have paid the payment of their orders. | Users’ transaction details will be fetched from the database and be displayed to the user. | User  Admin |
| Database |  |  | Admin |

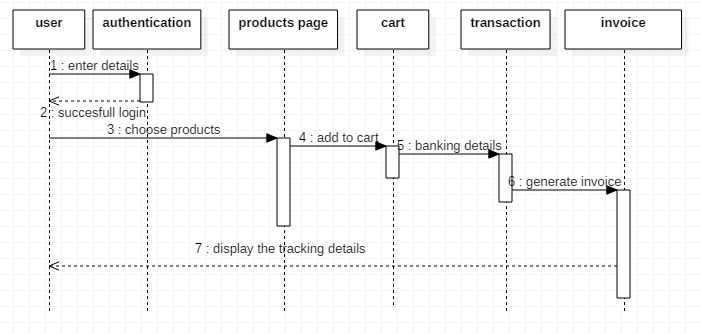
**CANCEL:**

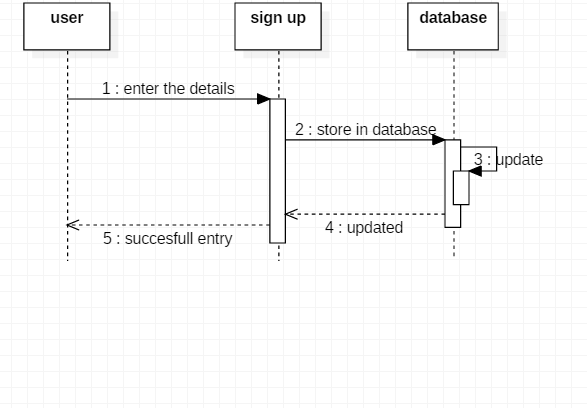
****

|  |  |  |  |
| --- | --- | --- | --- |
| Order details | User must have placed an order | Users order details will be fetched from the database and be displayed to the user. | Admin |
| Cancel order | User should have an order placed either paid/unpaid | User’s order requested would be cancelled and refund would be initiated if amount had been paid. | User  Admin |
| Refund |  | Refund will be initiated from the ORMS account to the user’s account if an order is cancelled and the user has made the payment | Admin |
| Database |  |  | Admin |
| Finance |  |  | Admin |

**SEQUENCE DIAGRAMS**

****

****

****

**TESTING**

**FUNCTIONAL TESTS:**

Functionalities:

* Login
* Form submission – Sign Up.

**LOGIN:**

Let there be two users:

**1) 2)**

Username: Matt Username: Nick

Password: 123 Password: xyz

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Case Type** | **Test Case Name** | **Test Cases** | **Expected Results** |
| 1 | Validation Testing | Test if user is able to login successfully. | **1)**  Username: Matt  Password: 123  **2)**  Username: Nick  Password:xyz | User must successfully login to the web page |
| 2 | Validation Testing | Test if unregistered users is not able to login to the site | **1)**  Username: matt  Password: 123ac  **2)**  Username: nick  Password:xyzbc | Proper error must be displayed and prompt to enter login again |
| 3 | Validation Testing | Test with valid username and empty password such that login must get failed | **1)**  Username: matt  Password:  **2)**  Username: Nick  Password: | Proper error must be displayed and prompt to enter login again |
| 4 | Validation Testing | Check if the login function handles case sensitivity | **1)**  Username:MATT  Password: 123  **2)**  Username: Nick  Password:XYZ | Login must fail saying incorrect username/ password |
| 5 | Validation Testing | Check if login handles multiple logins from same user. | **1)**  Username:MATT  Password: 123  **2)**  Username:MATT  Password: 123 | Login page should reply saying user already logged in |
| 6 | Defect Testing | Missing Information | **1)**  Username: Matt  Password:  **2)**  Username:  Password:xyz | Proper message should be displayed saying all fields are mandatory to be filled. |
| 7 | Defect Testing | Extra Information | **1)**  Username: Matt and nick  Password: 123  **2)**  Username: Nick  Password:xyz or 1=1 | Message saying irrelevant data should be displayed. |
| 8 | Defect Testing | Wrong type of input | **1)**  Username: @#$%  Password: 123  **2)**  Username: Nick  Password:^&() | Message should be printed saying improper input format. |
| 9 | Defect Testing | Verify account lock | **1)**  Username: matt1  Password: 1231  Username: matt2  Password: 1232  Username: matt3  Password: 1233  **2)**  Username: Nick1  Password:4561  Username: Nick2  Password:4562  Username: Nick3  Password:4563 | Account should be locked and access should be granted only after gettting certain assurance from the user |
| 10 | Defect Testing | SQL injection | **1)**  Username: Matt  Password: 123 or true  **2)**  Username: Nick  Password:xyz or 1=1 | Proper error should be displayed. |

**FORM SUBMISSION:**

**Let the form be:** (‘\*’ means required fields.)

Name: \_\_\_\_\_\_\_\_\_\*

ID: \_\_\_\_\_\_\_\_\_\*

Email: \_\_\_\_\_\_\_\_\_\*

Phone: \_\_\_\_\_\_\_\_\_

**One valid user:**

Name: Nick

ID:ab12

Email: nick@yahoo.com

Phone:9291234567

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Case Type** | **Test Case Name** | **Test Cases** | **Expected Results** |
| 1 | Validation Testing | Check the behavior of form by not filling up any data into the form. | 1)Name: NULL  ID:NULL  Email: NULL  Phone:NULL  2)  Name:  ID:  Email:  Phone: | Display message saying fill mandatory fields. |
| 2 | Validation Testing | Check the by not filling up the required fields. | 1)  Name: Nick  ID:ab12  Email:  Phone:9291234567  2)  Name:  ID:ab12  Email:  Phone: | Display message saying fill all required fields. |
| 3 | Validation Testing | Check the behavior of form by adding random data in the text field. | 1)  Name: ewfqer%$  ID:%\_)$(  Email: nickf$%^.com  Phone:fwggs234567  2)  Name: Nick  ID:%%%  Email: nick@yahoo.com  Phone:929\*\*567 | Display invalid content |
| 4 | Validation Testing | Check if you can type invalid mobile number w.r.t country code. | 1) Name: Nick  ID:ab12  Email: nick@yahoo.com  Phone: +22 9291234567  2)  Name: Nick  ID:ab12  Email: nick@yahoo.com  Phone:+1a 1234567 | Display invalid phone number. |
| 5 | Validation Testing | Check if the “Create account” button if pressed leads to new page. | 1)  Name: Nick  ID:ab12  Email: nick@yahoo.com  Phone:9291234567  [Button pressed if form filled properly]  2)  Name: Nick  ID:  Email:  Phone:9291234567  [Button pressed if form not filled properly] | Navigate to a new page if submission successful. |
| 6 | Defect Testing | Check the upper limit of the text boxes | 1)  Name: Nickkvhhch jh gwegewg  ID:ab12  Email: nick@yahoo.com  Phone:9291234567  2)  Name: Nick  ID:ab12gwergwer fvefvrferfe  Email: nick@yahoo.com  Phone:9291234567 | Display Max textbox limit exceeded. |
| 7 | Defect Testing | Verify if the default language of the sign up form is English (united states). | 1) Name: निक  ID:ab12  Email: nick@yahoo.com  Phone:9291234567  2) Name: நிக்  ID:ab12  Email: nick@yahoo.com  Phone:9291234567 | Show invalid language response. Change the default language to English (United states) |
| 8 | Defect Testing | Check the Email text field that has random string instead of real email. | 1)  Name: Nick  ID:ab12  Email: sdthert  Phone:9291234567  2)  Name: Nick  ID:ab12  Email: kfadjkf@dfbw  Phone:9291234567 | Invalid email.  Abort form submission. |
| 9 | Defect Testing | Check if you can switch between yahoo or ymail as email domain name. | 1)  Name: Nick  ID:ab12  Email: nick@ymail.com  Phone:9291234567  2)  Name: Nick  ID:ab12  Email: nick@yahoo.com  Phone:9291234567 | Access should be granted if domain names are from the same email service provider. |
| 10 | Defect Testing | Check if you can show and hide the ID by clicking on “show ID” button inside ID text box. | 1)  Name: Nick  ID:ab12  Email: nick@yahoo.com  Phone:9291234567  2)  Name: Nick  ID:\*\*\*\*  Email: nick@yahoo.com  Phone:9291234567 | ID should be hidden and seen as per the user request. |