



Alassani Fousseni Igodo

DEVELOPING AN ONLINE SHOP
WITH
OPEN SOURCE TECHNOLOGIES

Technology and Communication

2016

ACKNOWLEDGEMENT

I would like to express my gratitude to my thesis supervisor Mr. Ghodrat Moghadampour who has helped and guided me during my thesis. I would like to thank him for his endless assistance.

I would like to thank the Head of Degree Programme Mr. Seppo Mäkinen who has made me fulfilled my thesis.

I am grateful to my family for his support (Tuulia Hietikko, Daniel Etonam Amir Igodo, Mohamed Saddat Igodo, Laura, Ella-Noora, Mimosa, Hindou and Zalia) and also grateful to my friend Joseph Aduayi-Akue for his support.

My last gratitude goes to the Staff of Vaasa University of Applied Sciences who gave me huge knowledge during my studies.

Vaasa, 04th April 2016

Alassani Fousseni Igodo

VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES
Degree Program in Information Technology

ABSTRACT

Author	Alassani Fousseni Igodo
Title	Developing an online shop with open source technologies
Year	2016
Language	English
Pages	66 pages
Name of Supervisor	Ghodrat Moghadampour

The aim of the thesis was to develop an online shop website using open source technologies (PHP, HTML5, CSS3, JavaScript, MySQL and Apache Web Server) for electronics products in Togo where customers will be able to buy products online.

The application provides two separate views; public view and admin view. The public view is meant for users and customers where they can view and order products. The admin view is meant for application manager who can maintain products and customers.

The administrator will be able to see all the orders made by all the customers. The administrator will also be able to view, add, update and delete the product and customer information.

The user will be able to view, search and buy products. The application allows the users to register and login. The registered users will have their own accounts where they can see all their personal information as well as their orders. Registered customers will also be able to update their personal information, manage their orders and accounts.

Keywords

PHP, MySQL, HTML5, CSS3, JavaScript

CONTENTS

ABSTRACT

ACKNOWLEDGEMENT

1	INTRODUCTION	9
2	RELEVANT TOOLS AND TECHNOLOGIES.....	10
2.1	HTML 5.....	10
2.2	CSS3.....	11
2.3	PHP.....	12
2.4	JavaScript	13
2.5	MySQL.....	14
2.6	UML	14
2.7	XAMP	15
2.8	Mercury Mail Transport System	15
2.9	PhpMyAdmin	16
2.10	Notepad++	16
2.11	MySQL Workbench.....	16
2.12	PayPal	16
3	APPLICATION DESCRIPTION	18
3.1	Application Main Modules.....	18
3.1.1	Administration Modules	20
3.1.2	Front-End Modules	21
3.2	Applications Main Functions	22
3.3	MVC Module of Online Shopping Cart (MVC)	27
4	SYSTEM DESIGN	28

4.1	Module Function Design	28
4.2	Sale Module.....	28
4.3	Back-end management System Module design.....	29
5	DATABASE AND GUI DESIGN	30
5.1	Database	30
5.2	User Interface (GUI) Design	32
5.3	Administration Interface (GUI) Design	38
6	IMPLEMENTATION.....	44
6.1	Description	44
6.2	Implementation of Functions.....	44
6.2.1	Setting up Mercury Mail Transport.	44
6.2.2	Available products in the system	54
6.2.3	User Registration.....	56
6.2.4	Customer Login.....	58
7	TESTING	60
7.1	Customer Shopping Cart Testing	60
7.2	Customer Login Testing.....	60
7.3	Administration Login Testing	62
7.4	Administrator Login Successful page	63
8	conclusions.....	65
8.1	Future Work	65

LIST OF FIGURES AND TABLES

Figure 1: The structure of HTML5 webpage architecture /1/	10
Figure 2: The supported features of CSS3 in each browser /2/	11
Figure 3: A simple Description of a Web server using PHP /3/	12
Figure 4: MySQL database Concept /5/.....	14
Figure 5: PayPal transaction model /12/	17
Figure 6: Application main functions	19
Figure 7: User Registration Function.....	23
Figure 8: Customer Login Function.....	24
Figure 9: Administrator Registration Function.....	25
Figure 10: Administrator Login Function.....	26
Figure 11: MVC Diagram /13/.....	27
Figure 12: Sale Module.....	29
Figure 13: Back-end Management system.....	29
Figure 14: Database diagram and tables created for our Xampp (phpMyAdmin).....	30
Figure 15: Administrator database table	31
Figure 16: Products brand database table	32
Figure 17: Home Page.....	33
Figure 18: Customer Registration page	34
Figure 19: Customer login page.....	35
Figure 20: Customer Account Page	36
Figure 21: Shopping cart.....	37
Figure 22: Checkout page with PayPal	38
Figure 233: Admin Registration Page.....	39
Figure 24: Email received in the new administrator Email box	40
Figure 25: New administrator data in the table, not activated yet	40
Figure 267: New Administrator Account activated in the database	41
Figure 278: Administrator login page.....	42
Figure 28: Running Xampp, Mercury, Apache and MySQL.....	45

Figure 29: Admin Panel of Mercury Mail Server	46
Figure 30: Mercury Configuration Sub Menu	47
Figure 31: Mercury SMTP Server	48
Figure 32: Mercury POP Server Configuration	49
Figure 33: MercuryE SMTP Client.....	50
Figure 345: Setup Email address	51
Figure 354: Email address creation.....	51
Figure 36: Fake email sending	52
Figure 37: Fake email received.....	52
Figure 38: Test customer login result	61
Figure 39: Customer login test 2.....	62
Figure 40: Administrator login test 1	63
Figure 41: Administrator Login success page.....	63
Figure 42: Product Inserting test.....	64
Snippet 1: Customer Session	53
Snippet 2: Session closing	53
Snippet 3: Product attributes from database	55
Snippet 4: Code for product display	55
Snippet 5 : User Registration function	57
Snippet 6: Customer Login checking	58
Snippet 7: Code for accessing customer account.....	59

1 INTRODUCTION

Online shops also called Ecommerce is a form of electronic commerce which allows people to buy goods or services directly from a seller by using the Internet through a web browser. It can be digital products like eBooks etc.

In Togo there are many people who have their business and who want to have a website (Ecommerce) so that they can sell their products through internet. Hence we are going to explain step by step how the Ecommerce can be designed and implemented using PHP, HTML5 and MySQL technologies. The main tool of the project development is PHP language.

The project is divided into two (2) main categories: Administrators (shortly called Admins) and Consumers (shortly called Users).

The owner or the staffs can be the admins. They could add products, modify products, delete products simply they could edit products and put all the information needed on the website.

The consumer (also called User) is anyone who browses the website to find products and buy them. The user will be able to browse, find, add products to the cart and checkout using PayPal. He will be able to create an account and manage his own account. He will be able to edit and update his personal information, check the status of his orders.

This thesis contains five (5) chapters in order to explain the project. The first chapter introduces the whole project; the second explains the development environment, the technology and the tools used in the whole project. The structures and requirements are described in Chapter 3. Chapter 4 describes how the requirements are achieved. And the Chapter 5 is the overall conclusions of the project.

2 RELEVANT TOOLS AND TECHNOLOGIES

Relevant tools and technologies used for the project will be discussed in this section.

2.1 HTML 5

Hypertext Markup Language, commonly referred to as HTML is the basic or standard markup language used to create web pages. It is a form of HTML elements. The latest version of HTML is HTML5. Most of the modern web browsers nowadays support HTML5. It supports new features, new HTML elements, new attributes, full CSS3 support, video and audio, 2D/3D graphics that help users and especially web developers. With HTML5 it is easy to include and handle audio and video in the browser without thinking about plugins. The structure of HTML5 webpage architecture is represented in the figure below.

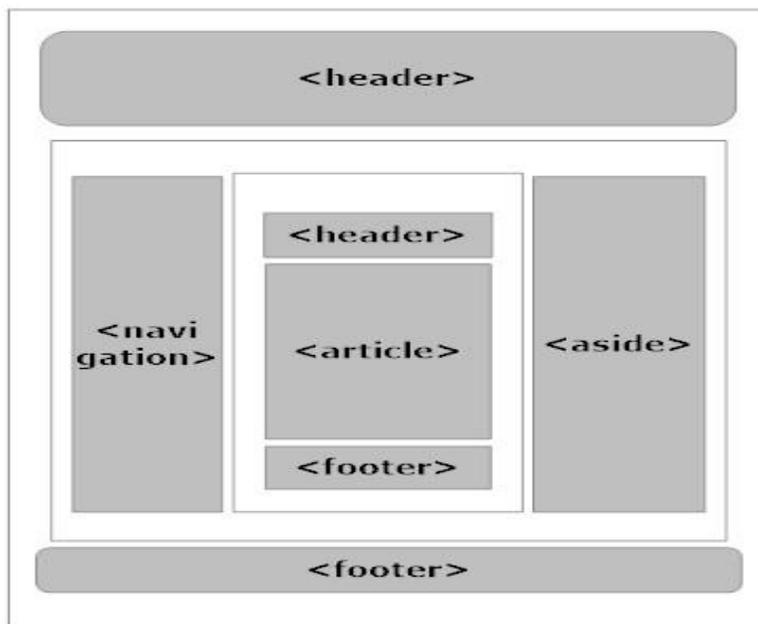


Figure 1: The structure of HTML5 webpage architecture ./1/

The DOCTYPE declaration is very simple in HTML 5 as well as the character encoding declaration:

```
<! DOCTYPE html>
```

```
<meta charset="UTF-8">
```

2.2 CSS3

CSS (Cascade Style Sheet) is a style sheet language for describing the framework and layout of web pages. It includes colors, layout and fonts. All the modern web browsers support it. Most of CSS3 modules are W3C Recommendations like HTML5.

At least partially supported									
	IE 11	Firefox 41	Chrome 46	Safari 8	Opera 31	Opera Mini 5.0-8.0	Android Browser 40	Chrome for Android 42	Firefox for Android 38
Canvas (basic support)	Yes	Yes	Yes	Yes	Yes	Partial	Yes	Yes	Yes
New semantic elements	Yes	Yes	Yes	Yes	Yes	Partial	Yes	Yes	Yes
dataset & data-* attributes	Yes	Yes	Yes	Yes	Yes	Partial	Yes	Yes	Yes
Form validation	Yes	Yes	Yes	Partial	Yes	Partial	Yes	Yes	Yes
disabled attribute of the fieldset element	Partial	Yes	Yes	Yes	Yes	Partial	Yes	Yes	Yes
Spellcheck attribute	Yes	Yes	Yes	Yes	Yes	Partial	Partial	Partial	Partial

Figure 2: The supported features of CSS3 in each browser /2/

CSS can be define inline in a webpage or can be define in a separate style sheet file and import to the webpages as follow:

```
<! DOCTYPE html>

<html lang = "en-US">

<head>

<meta charset = "UTF-8">
```

```

        <link rel = "stylesheet"    type = "text/css"    href =
"myStyle.css" />

</head>

```

2.3 PHP

PHP (Hypertext Preprocessor) is a server side language. It is designed for web development to implement dynamic web pages and can be embedded into HTML. PHP codes executes on server before sending the output to browser.

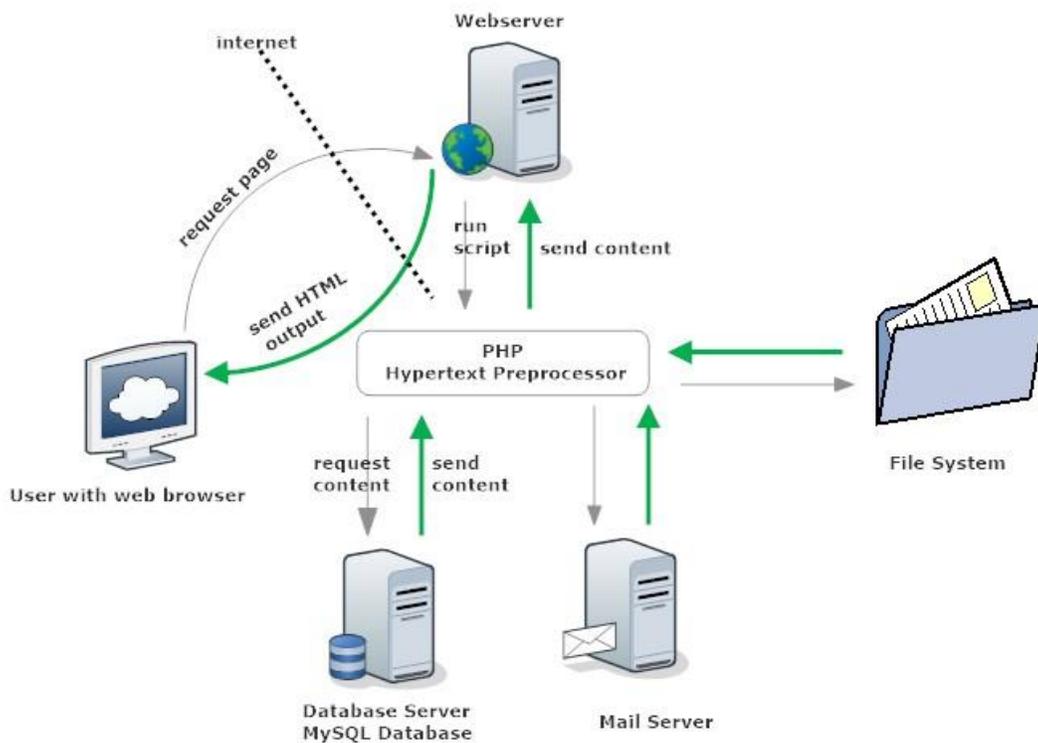


Figure 3: A simple Description of a Web server using PHP /3/

When a user requests a page using a web browser, the request is sent to the web server where a PHP script is run. The PHP request content from MySQL server where all the data are saved. The database server send the requested content to the web server, which sends the output as HTML file to the browser of the user. PHP is used often for dynamic pages in web development.

The default syntax start with ” <?php “and end with “?>”.

```
<?php  
  
Echo "This is a php syntax form"  
  
?>
```

2.4 JavaScript

JavaScript is a powerful and dynamic language for programming on the web. It can be embedded into HTML pages or it can be independently. It provides client-side interactivity directly via the web browser. It can also be object based functional language./4/

The difference between JavaScript and PHP is that JavaScript codes execute directly on the client browser and PHP codes execute on the server before sending the result to the browser.

Javascript basic syntax is as follow:

```
<script language="javascript" type="text/javascript">  
  
    document.write("Hello World!")  
  
</script>
```

2.5 MySQL

MySQL is an open source used for database management. It is an open source Relational Database System (RDBMS) that uses Structured Query Language(SQL). It is most noted for its fast processing, proven reliability, ease and flexibility of use. It is much use in most of the Content Management Systems(CMS).

MySQL is sponsored by the Swedish company MySQL AB, which is owned by Oracle Corp. It is written in C and C++ and is compatible with all major operating systems.



Figure 4: MySQL database Concept /5/

2.6 UML

The Unified Modeling Language (UML) is mainly used to offer a way to visualize systems architecture in a diagram. It is used for constructing and documenting of a system too. /6/

We implement our online shop using a local server, we used XAMPP as web server.

2.7 XAMPP

XAMPP is an integrated development environment, which includes Apache HTTP Server, MySQL Database and PHP, Mercury, PERL or Python on a Windows-based computer.

Apache is just an open source web server (free). MySQL is an open source database.

Mercury is a Mail Transport System. It will be used in our project to send and receive mail as we will be working on an environment (XAMPP) which offers local web server.

XAMPP comes with other different modules too including OpenSSL and PHPMyAdmin. /7/

In our project we will use PHP, MySQL, and Mercury on Windows 8 operating System.

2.8 Mercury Mail Transport System

Mercury Mail Transport System (Mercury MTS) is a great way to send mail from a local server. Mercury Mail Transport System (Mercury MTS) is a standards-compliant donation ware (was freeware prior to January 2007) mail server developed by David Harris, who also develops the Pegasus Mail client.

Mercury is a fully independent mail server and can provide email services to all standards-compliant email clients, such as Eudora or Microsoft Outlook. Both versions of Mercury are highly modular, allowing support for different sets of Internet protocols to be installed as required. /8/

Mercury is extremely standards-compliant, supporting all major Internet mail-related protocols including SMTP (for both sending and receiving mail), POP3 and IMAP. The Win32 version also supports a dialup connection. Both versions have many features, with especially powerful support for managed mailing lists.

2.9 PhpMyAdmin

PhpMyAdmin is a free and open source MySQL administration tool written in PHP and was first released in 1998 under the GNU General Public License. It has cross-platform support for the major operating systems and supports administration of multiple servers.

It supports most of MySQL features and has an intuitive web interface. It also has supports for creating PDF graphics of database layout, importing data from CSV and SQL formats as well as exporting data to various formats, such as SQL, XML, PDF, CSV, among others. /9/

2.10 Notepad++

Notepad++ is free open source code editor for Microsoft operating system and it supports many programming languages. It is a useful tool for developers. /10/

2.11 MySQL Workbench

MySQL Workbench is a unified visual tool for database architects, developers. It provides data modeling, SQL development and comprehensive administration tools for server configuration. /11/

2.12 PayPal

PayPal Holdings. Inc is an American company operating a worldwide online payments system. PayPal serves money transfer as electronic alternatives to traditional paper methods like checks and money orders.

PayPal is the fastest, safer way to send money, make online payments, receive money or set up a merchant account. PayPal is one of the world's largest internet payment companies.

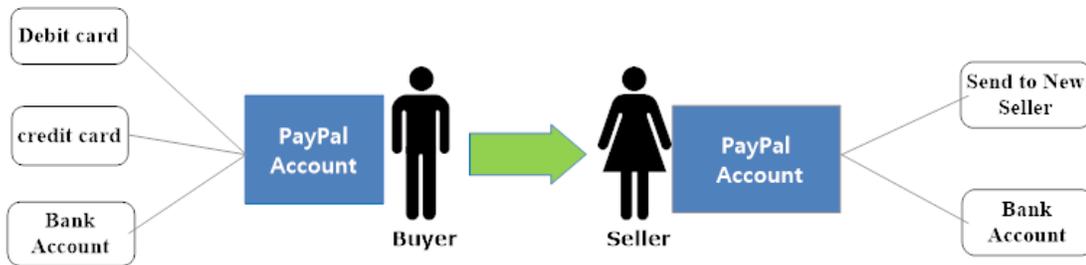


Figure 5: PayPal transaction model /12/

3 APPLICATION DESCRIPTION

The system analysis is based on the requirements of the whole project.

3.1 Application Main Modules

In order to implement the project, it is necessary to have some functions that will make the whole workload easier.

This section is divided into different parts; basic requirements and must have requirements (Customers).

Basic requirements allow any user to:

- Search: search products by key word
- Display: display products by name, word, category and brand
- Add to shopping cart: add products to shopping cart
- Contact: contact info desk for information
- Register: register and be able to order

Must have requirements allow any registered user (customer) to:

- Login: login with registered email and password
- View: view own personal information, orders and payments
- Edit: edit personal information
- Change own password

The diagram below helps to have the overview of the people involved and what they will be able to do. It is an important phase in a system analysis, and the developer can show the requirements of all the people involved.

In the project there are two (2) main groups: User group and Administrator group.

The user group is divided into 2 groups; simple user and customer.

A simple user will be able to browse products, search products and add them to the shopping cart and a customer will be able to do everything that a simple user can do and will also be able to login, manage own account (edit personal information, update information, change profile picture and password) and also be able to track the status of own orders.

An administrator will be able to search, view, edit, delete products. An administrator will be able to view all the products, all the customers, all the orders and check all the payments. An administrator will be able to update the customer information.

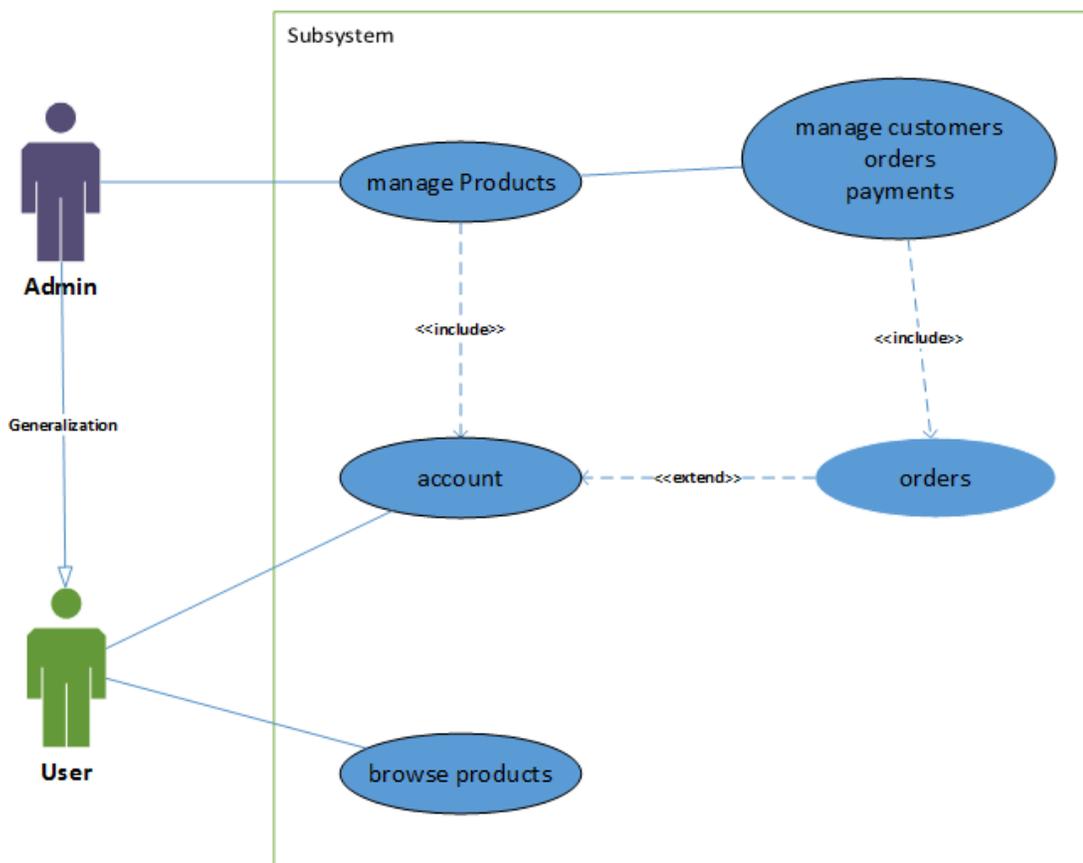


Figure 6: Application main functions

3.1.1 Administration Modules

This system management module is the administrator module and also the registered members' module. It allows the administrator to add products, product information etc. and also the registered members to manage their own account.

Administrator modules are back end modules. There are modules that gives rights to the administrator to manage the online shop.

- **Product Management Module**

This is the function for the management of the products information, such as product name, description, price, products images etc.

- **Members Management Module**

The implementation of this function allows the administrators to manage all the registered members, view all the members and so on...

- **Administrator Account Module**

This is for the management of basic information of the system administrator, add new administrator, modify password.

- **Admin Login Module**

With this function the administrator can login to the back-end management system and use all the functions or modules.

- **Payment View Module**

This provides to the administrator to view all the payments made by the customers.

- **Admin Logout Module**

The administrator will be logged out with this function.

3.1.2 Front-End Modules

The modules below are mainly for front-end. They provide any user to view, browse products and buy products.

- **Registered User (Customer)**

This function provides to the registered members to update their own information: email. Address, phone number, profile picture, to change password etc...

- **Order Module**

The registered members or customers will be able to track their orders status with this function.

- **Customer Logout Module**

The logged customers will be able to logout safely from their account.

- **Customer Checkout Module**

All logged customers will be able to checkout, to pay the products they already added to their cart with PayPal.

- **Shopping Cart**

Any user will be able with this function to add product to the cart, and also modify the quantity by viewing the cart.

In order to checkout a user needs to add at least one product to the shopping cart and also create an account.

The shopping cart is an important element in any online shopping system.

- **Browsing Module**

This function is needed to view products and browse. Any user can search; browse all the products on the website. With this function, a user will be able to get all the information about any single product name, description, price, and brand.

- **User Registration module**

This provides the registration function for the user. With this function user will be able to create his own account, and then become a registered member.

- **Customer Login Module**

With this function the customer or registered member will be able to login into his own account with his email and his password.

- **Search Module**

Any user will be able to search products on the website with this function.

3.2 Applications Main Functions

This is a graphical representation of the system, it explains how data is processed and transferred into the system. It shows the relationship of each part.

Before the general user even view or search products, the administrator needs to add the products. Through the product management module, the administrator will save the product information into the database and then display them on the website. And it is at that stage that the general user will be able to view and search products.

When buying also the registered members (customers) will provide their information including login information, purchase information and order information.

Below, are shown the functions needed for the project using classes.

▪ User Registration Function

For a user to purchase on the web store, he needs to register. He first will fill the form on the registration page. If all the data fields respect the requirements of the fields, the user will submit the form. After submitting, a PHP file will check the data and if any field is incorrect, the application will send an error message and then the user will still be on the registration page with the error fields marked, if the data are correct, the PHP file will send the data to the database and insert them to the corresponding table “Customers” and then redirect the user who becomes a customer automatically to the customer page. The sequence diagram below show the steps explained.

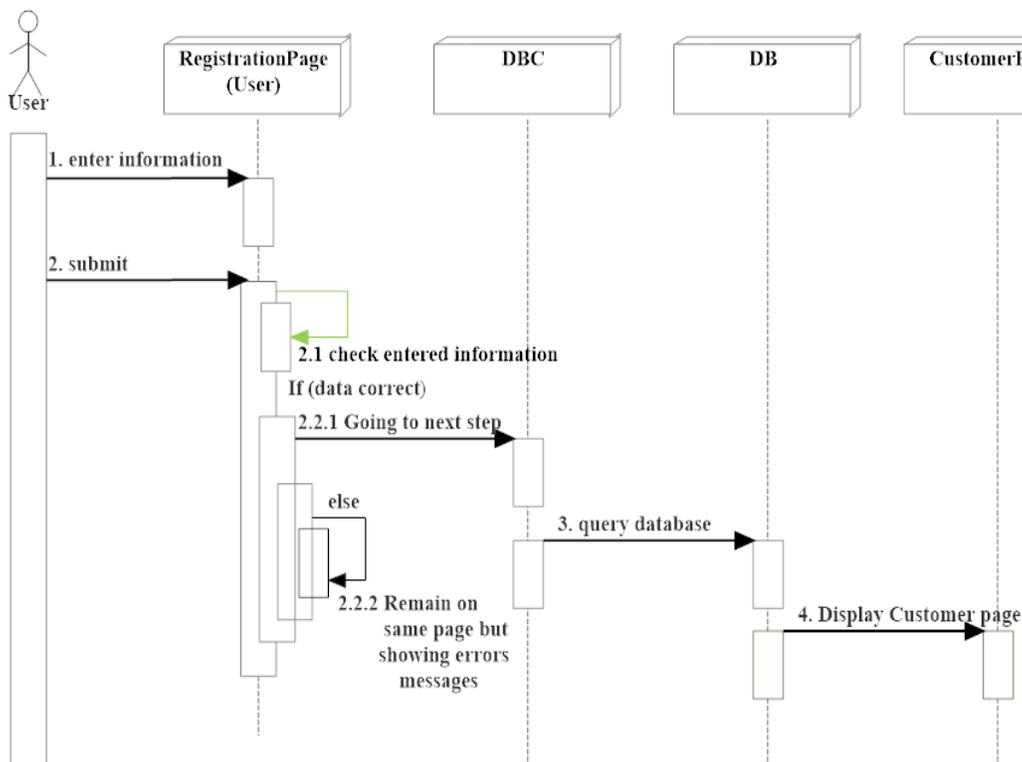


Figure 7: User Registration Function

▪ Customer Login Function

In the step, the customer will use his confidential data (his email and password given when registering) to login. After filling the form and submitting it, the PHP file responsible for the customer login, will check if all the data provided by the customer are the same in the database (table “Customers”). If there are not the same, the error messages will be shown on the corresponding field to the customer. If the data are the same with the data in the database, the customer will be redirect to the customer page. If the customer, without login, has a product in the cart and would like to checkout, he will be redirected to the checkout page after login.

The diagram below shows the sequence of customer login application.

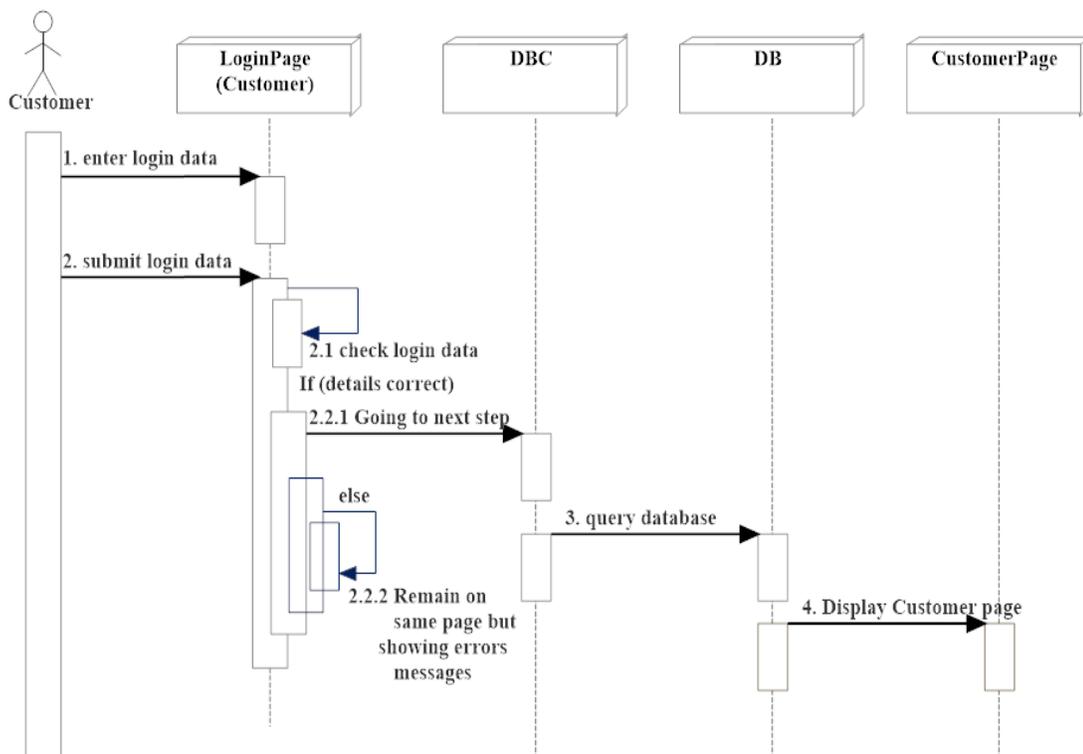


Figure 8: Customer Login Function

▪ Administrator Registration Function

The steps are the same as the customer registration but for the administrator, the data are saved or stored in the database (table “administrators”). When all the data are correct, a confirmation email is sent into his email with an activation link.

After clicking on the activation link, the administration will be able to login with his confidential information (his email and password).

The diagram below shows the sequence of administrator registration application.

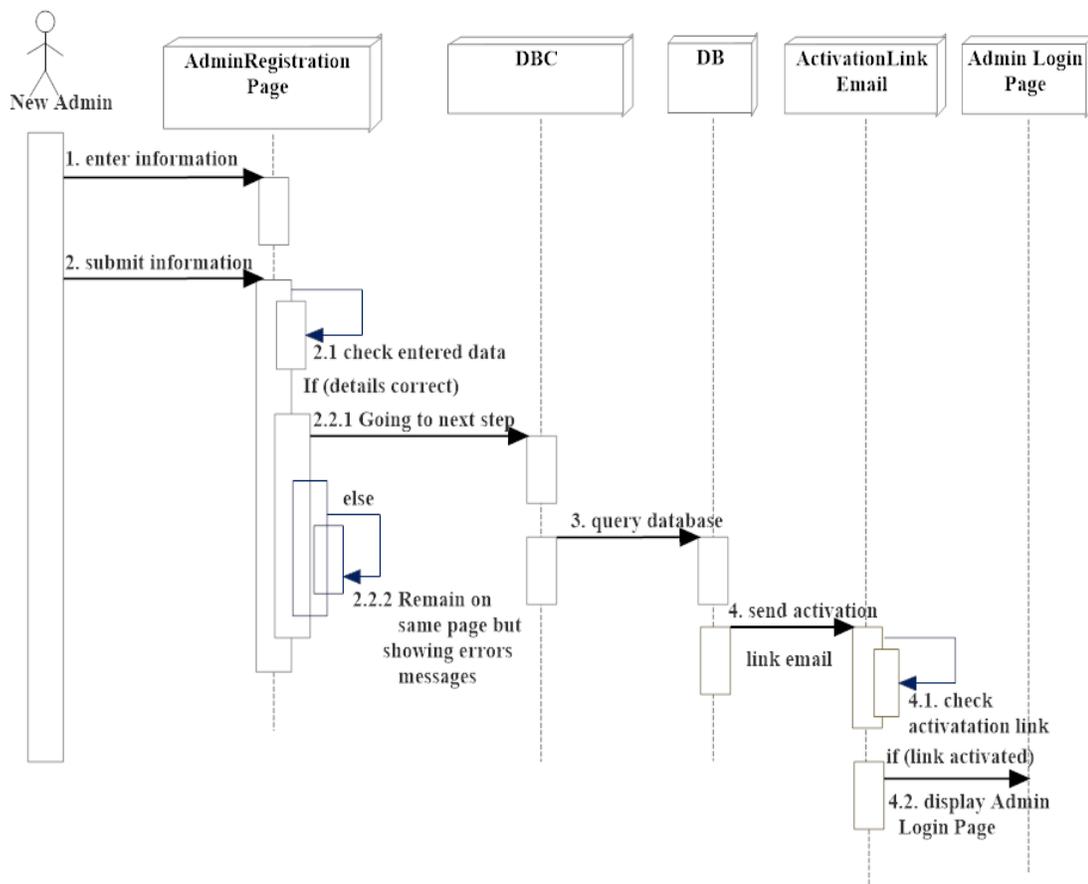


Figure 9: Administrator Registration Function

- **Administrator Login Function**

The phase is the same as the customer login function phase with the corresponding tables in the database.

The diagram below shows the sequence of administrator login application.

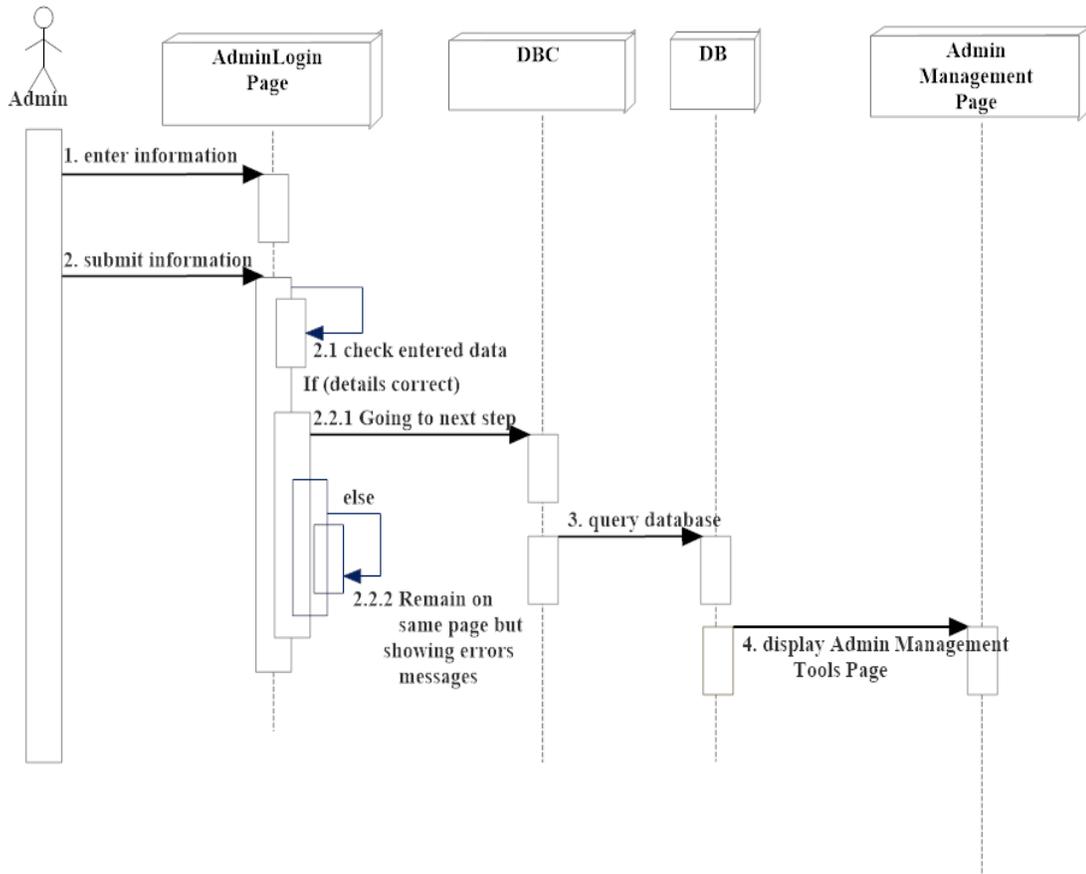


Figure 10: Administrator Login Function

3.3 MVC Module of Online Shopping Cart (MVC)

This phase is the Model module, View module and Controller module. The diagram below illustrates all the steps and all the modules connections with the database as well.

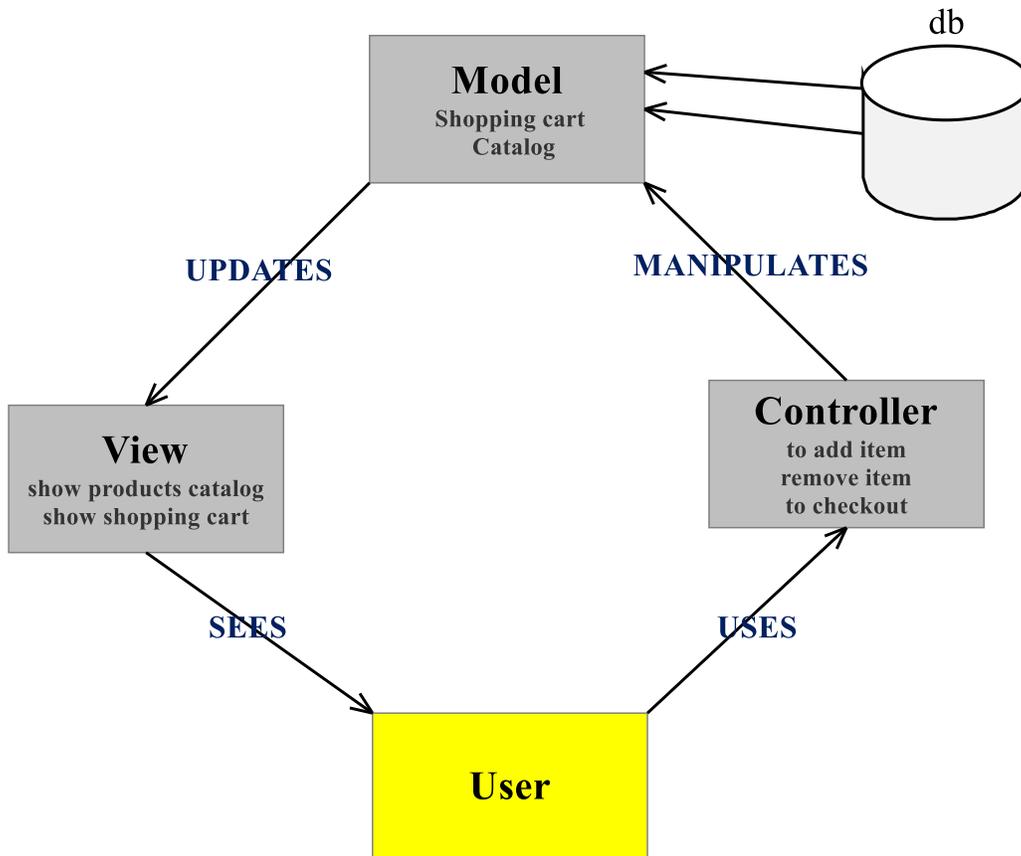


Figure 11: MVC Diagram /13/

From the figure we can see all the components in different modules. All the modules enable the user to browse products, delete products and display the contents of the shopping cart and also to checkout.

4 SYSTEM DESIGN

This part is about to transform the logical model into a physical model. It means that all the functions or modules listed in the previous part will be designed with structure and database design.

The whole online system can be divided into two (2) main sections: Management Module section and the Sale section module.

4.1 Module Function Design

The whole system is divided into two main sections, and each main module has many sub-modules.

The Sale module includes shopping cart, customers' login and logout, and other sub-modules.

The system management module includes administrator login and logout, product management, order management, and other sub-modules.

4.2 Sale Module

Below is the figure to show the sale module function diagram.

The sale module includes different modules: shopping cart, login and logout, customer center, product advertising, special products.

In The shopping cart module, any customer can add, delete, update products in the cart.

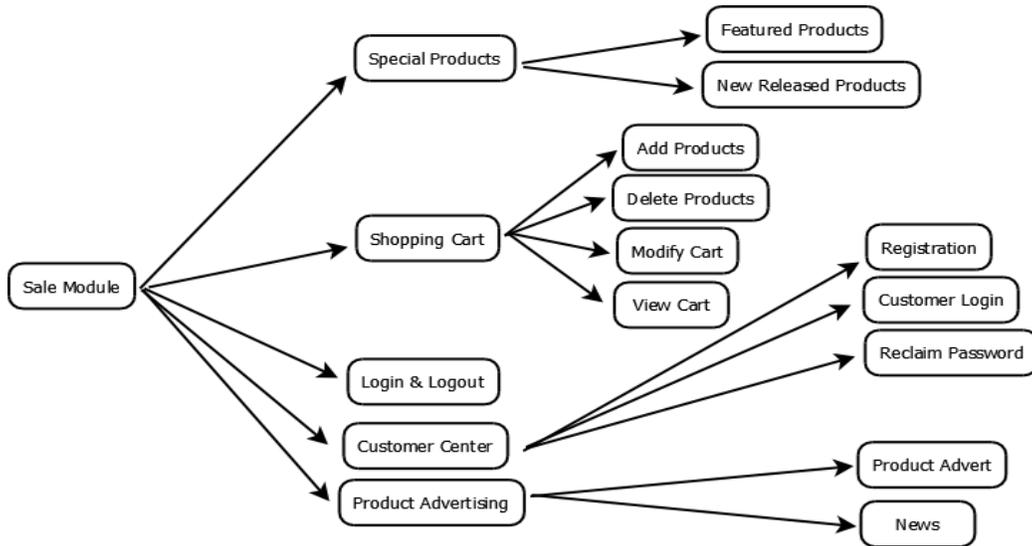


Figure 12: Sale Module

4.3 Back-end management System Module design

This includes many modules: products, brand, category, orders, customer management modules.

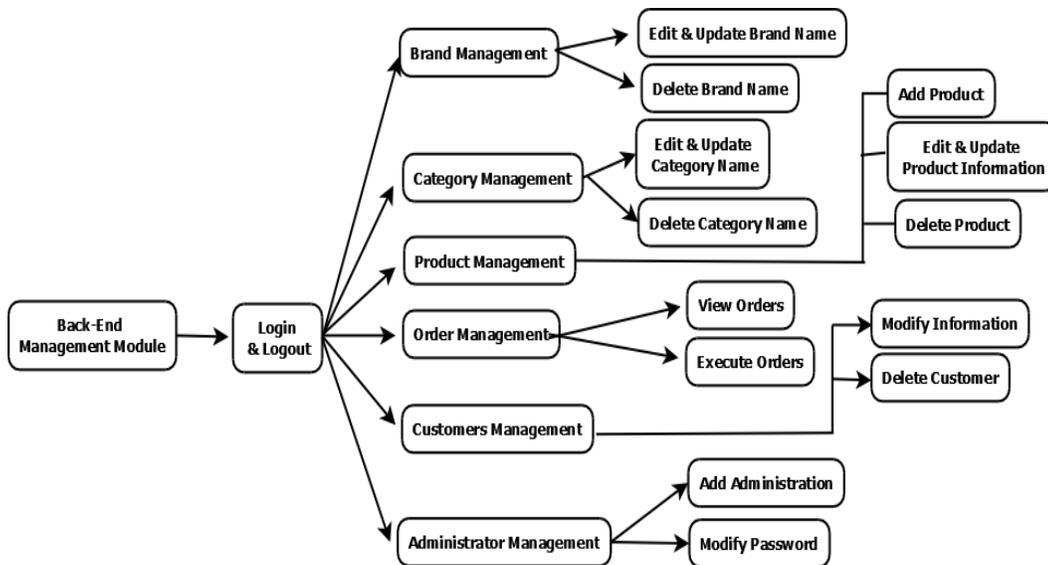


Figure 13: Back-end Management system

5 DATABASE AND GUI DESIGN

This part contains design for database and application GUI.

5.1 Database

MySQL database is used to store application data for this project. MySQL Database is used to create database, tables for the project. MySQL is a relational database management system. It is free and open source. As this project involved a lot of data, it is good to use database so that the whole data management be easy. All the data are stored in a specific table, and each table has specific number columns and rows.

The database name for our project is “mythesis”. It has 7 tables named as admins, brands, cart, categories, customers, orders and products shown on the figure below.

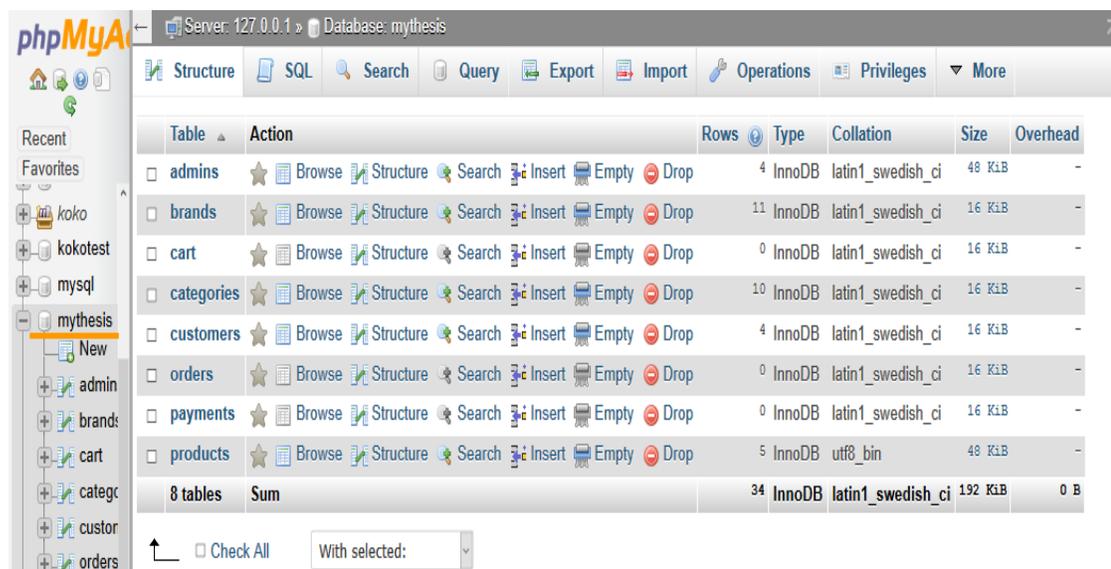


Table	Action	Rows	Type	Collation	Size	Overhead
admins	Browse Structure Search Insert Empty Drop	4	InnoDB	latin1_swedish_ci	48 KiB	-
brands	Browse Structure Search Insert Empty Drop	11	InnoDB	latin1_swedish_ci	16 KiB	-
cart	Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	16 KiB	-
categories	Browse Structure Search Insert Empty Drop	10	InnoDB	latin1_swedish_ci	16 KiB	-
customers	Browse Structure Search Insert Empty Drop	4	InnoDB	latin1_swedish_ci	16 KiB	-
orders	Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	16 KiB	-
payments	Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	16 KiB	-
products	Browse Structure Search Insert Empty Drop	5	InnoDB	utf8_bin	48 KiB	-
8 tables	Sum	34	InnoDB	latin1_swedish_ci	192 KiB	0 B

Figure 14: Database diagram and tables created for our Xampp (phpMyAdmin)

A database table has its own unique name and consists of columns and rows.

For each table we need to mark at least one field as primary key. The primary key is always unique in a specific table.

The table “admins” is created to store all the administrators’ information. It consists of 5 different fields called columns to store admin id, first name, last name, email, and password. The email and the password are really important because they are the requirement to be able to login into the system.

“admin_id” is the PRIMARY KEY.

Many columns of one table can be defined as PRIMARY KEY.

Each column has a name, a datatype and other optional attributes.

The screenshot shows the phpMyAdmin interface for the 'admins' table. The table structure is as follows:

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	<u>admin_id</u>	int(11)			No	None	AUTO_INCREMENT	Change Drop More
2	admin_fname	varchar(100)	latin1_swedish_ci		No	None		Change Drop More
3	admin_lname	varchar(100)	latin1_swedish_ci		No	None		Change Drop More
4	admin_email	varchar(255)	latin1_swedish_ci		No	None		Change Drop More
5	admin_password	char(128)	latin1_swedish_ci		No	None		Change Drop More
6	admin_username	varchar(50)	latin1_swedish_ci		No	None		Change Drop More
7	admin_address	varchar(255)	latin1_swedish_ci		No	None		Change Drop More
8	admin_phone	varchar(255)	latin1_swedish_ci		No	None		Change Drop More
9	Activation	varchar(40)	latin1_swedish_ci		Yes	NULL		Change Drop More

Figure 15: Administrator database table

The table “brands” is created to store all the information related to brands of the products. It has 3 columns to store brand id, brand name and brand description.

“brand_id” is the PRIMARY KEY.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	<u>brand_id</u>	int(100)			No	None	AUTO_INCREMENT	Change Drop Primary More
2	brand_name	varchar(100)	latin1_swedish_ci		No	None		Change Drop Primary More
3	brand_desc	longtext	latin1_swedish_ci		No	None		Change Drop Primary More

Figure 16: Products brand database table

5.2 User Interface (GUI) Design

The project has been divided into two main interfaces: User Interface and Administrator Interface. The user will be able to visit the website pages and search products, add products to the shopping cart and edit the number of products in the cart; the administrator will be able to add, edit, update and delete products;

In order to buy the user will be redirected to registration page, and after that will be able to order and pay products; and will be able to have access to customer page where own information can be edited, track own orders and orders history.

- **Home Page**

The home page is the index page that will be accessed when connected to the domain name of the company. It shows the basic information as products images, products

name, registration link, login link, cart, selling company contact information, features products, social Medias links as shown below.

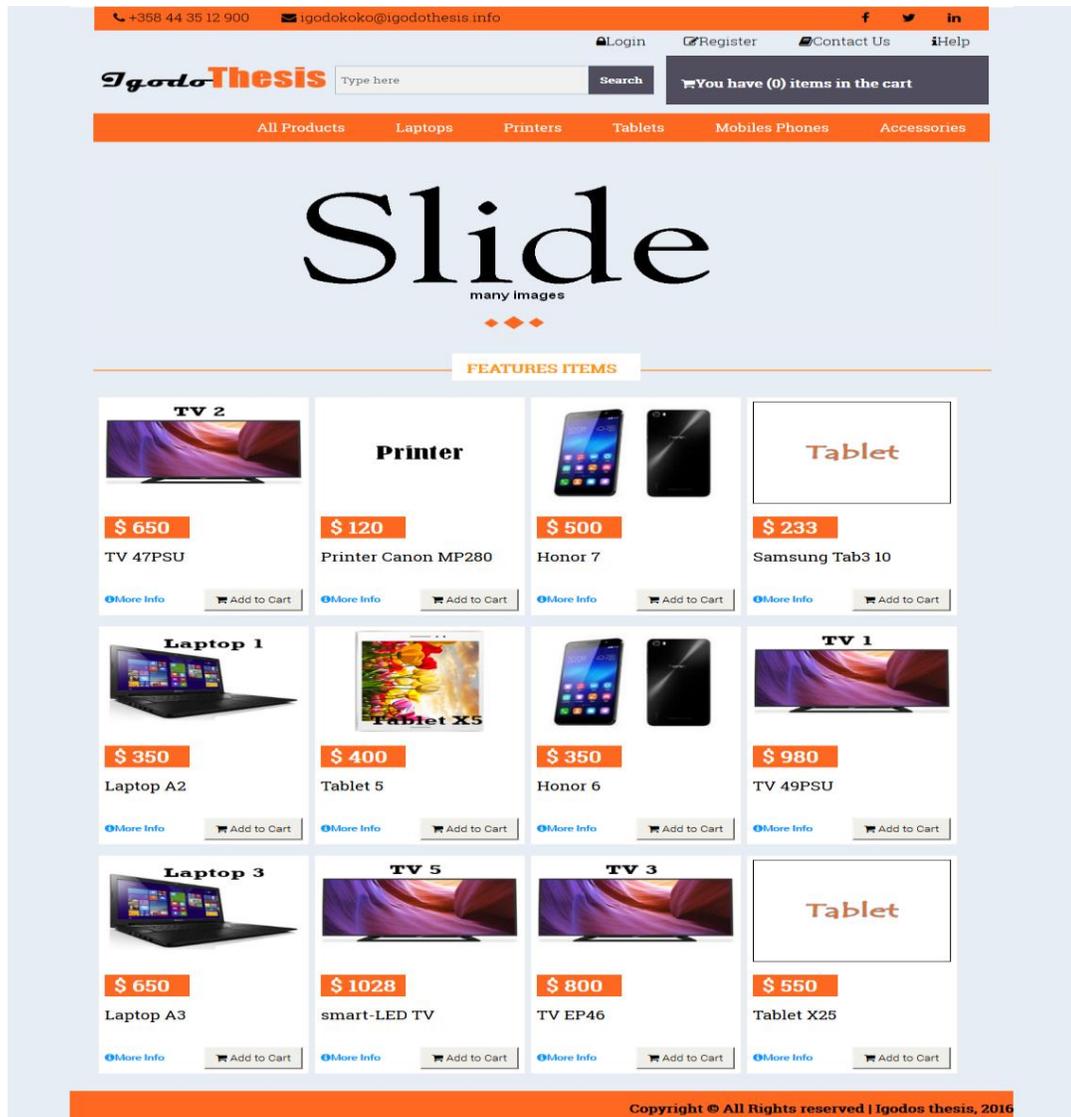


Figure 17: Home Page

- **Registration Page**

When a user clicks on the registration page, he will be redirected to the registration page where he will fill the fields on the registration form show below. After filling the

required fields, the user will click on register button and the browser will send all the information into the registration database that is called “customers”.

An php file will execute the necessary function in order to insert all the data in the appropriate table. The php file that will execute the registration is called “customer_registration.php”. The registration page is linked to the server by the customer_registration.php”.

The screenshot displays the 'IqodoThesis' website's customer registration page. At the top, there is a navigation bar with links for 'Login', 'Register', 'Contact Us', and 'Help'. Below this is a search bar with the placeholder text 'Type here' and a 'Search' button. To the right of the search bar, it indicates 'You have (0) items in the cart'. The main navigation bar includes links for 'All Products', 'Laptops', 'Printers', 'Tablets', 'Mobiles Phones', and 'Accessories'. The page is divided into two main sections: 'CATEGORIES' on the left and 'CREATE AN ACCOUNT' on the right. The 'CATEGORIES' section lists various product types like TABLET, LAPTOPS, TV, DIGIBOX, BLU-RAY, CAMERA, PRINTERS, PHONES, and ROUTERS. The 'BRANDS' section lists manufacturers such as ACER, TOSHIBA, SAMSUNG, LG, SONY, HP, ASUS, APPLE, CANON, NOKIA, and HUAWEI. The 'CREATE AN ACCOUNT' section contains a registration form with the following fields: First Name (*), Last Name Name (*), E-mail (*), Password (*), Username, City (*), Address (*), Postal code (*), City (*), Phone Number, and Profile Image. The Profile Image field includes a 'Browse...' button and the text 'No file selected.'. A 'Register' button is located at the bottom right of the form. A note indicates that (*) means obligatory field.

Figure 18: Customer Registration page

- **Login Page**

Any user who will register, will be redirected to the customer page. Any customer will be able to login with his confidential information; email and password. This login page is linked to the server using a php file called: “customer_login.php”.

Figure 19: Customer login page

When the email and password are provided, the login form will check all the fields in order to be sure that they respect the fields’ requirements. After that step, “customer_login.php” file will connect to the server and check if the confidential information is correct with the one stored in the table “customers”. If the information is correct, the customer will be redirected to the customer page; otherwise he will be asked to give the confidential information (email and password) again.

- **Customer Page**

This page is only accessed if customer credentials are authenticated. The customer can see on this page link to edit his own information, track his orders, orders, change password, logout.

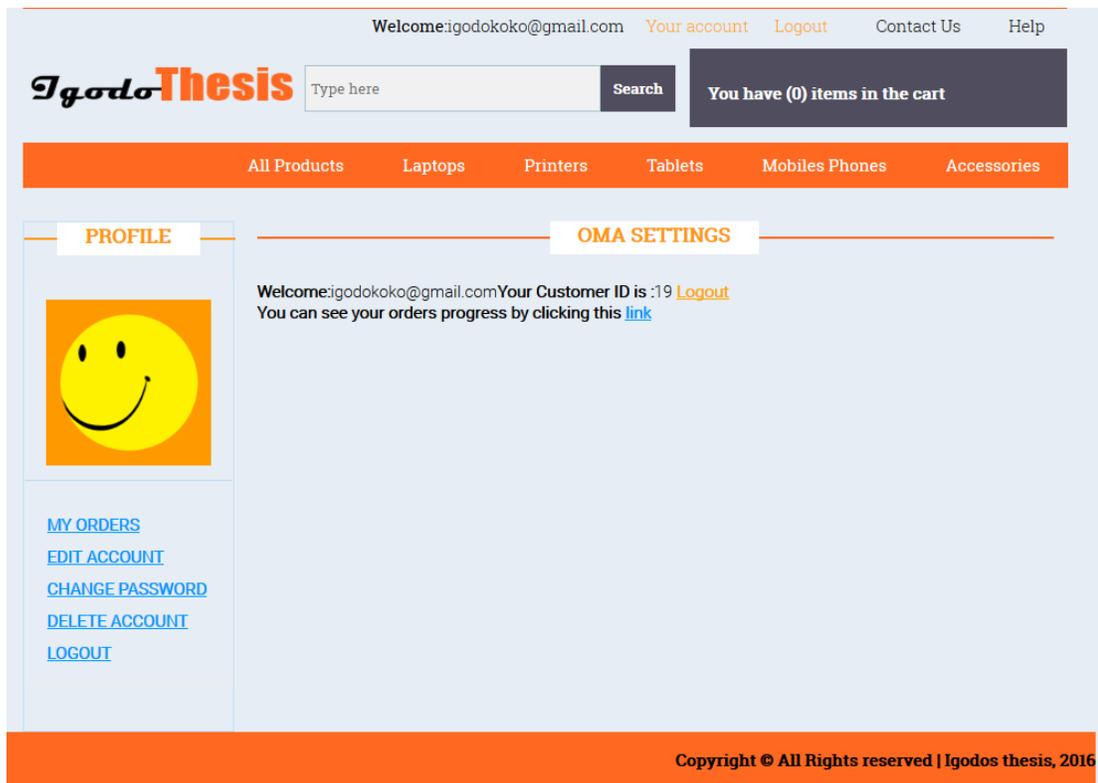


Figure 20: Customer Account Page

- **Shopping cart Design**

The user can add products in the shopping cart and checkout. All the data in the shopping are stored in the table “shopping” in the databse “mythesis”.

IgodoThesis Type here Search You have (1) items in the cart

All Products Laptops Printers Tablets Mobiles Phones Accessories

Slide

many images

Welcome:igodokoko@gmail.com || Below are the products in Your Shopping Cart || [Back to Shop](#) || [Logout](#)

CATEGORIES

- [TABLET](#)
- [LAPTOPS](#)
- [TV](#)
- [DIGIBOX](#)
- [BLU-RAY](#)
- [CAMERA](#)
- [PRINTERS](#)
- [PHONES](#)
- [ROUTERS](#)

Remove	Product(S)	Quantity	Price Per Item
<input type="checkbox"/>	Tablet 5 	2	\$400
Sub Total:			\$800

Delete Product Continue Shopping Update Cart

Checkout

Figure 21: Shopping cart

- **Payment**

The customer will be able to pay all the products in the shopping cart with PayPal.



Figure 22: Checkout page with PayPal

5.3 Administration Interface (GUI) Design

- **Administration registration Page**

We can have many administrators but the main administrator is the only one who can add other administrators. In order to add any administrator, he need to get personal information such as firstname, lastname, username, password, address and phone number from that new administrator. After he will fill the admin registration form with those information as shown below. After a successful registration, an email with a confirmation link is sent into that new administrator's email box for activation as shown below. After the activation, the administrator will be able to login to the administrator area.

THIS IS THE ADMIN REGISTRATION FORM

The screenshot shows a web form titled "Registration Form" with a sub-header "Create A new ADMIN Account" and a link "Already a ADMIN?". The form contains the following fields and values:

First Name :	Vamok
Last Name Name :	Vamk
E-mail :	vamk@localhost.com
Password:	●●●●●●●●●●
Username :	vamok
Address :	Verkkokat u
Phone Number :	0443512906

A "Register" button is located at the bottom right of the form.

Figure 233: Admin Registration Page

Thank you for registering! A confirmation email has been sent to vamk@localhost.com Please click on the Activation Link to Activate your account

THIS IS THE ADMIN REGISTRATION FORM

The screenshot shows the same "Registration Form" as in Figure 233, but with all input fields empty. The fields are:

- First Name :
- Last Name Name :
- E-mail :
- Password:
- Username :
- Address :
- Phone Number :

The "Register" button is still present at the bottom right.

Figure 24: Admin Successful Registration Page

We can see from the email box that the new administrator has received the confirmation email with the activation link shown below. He just will activate his account by clicking the link.

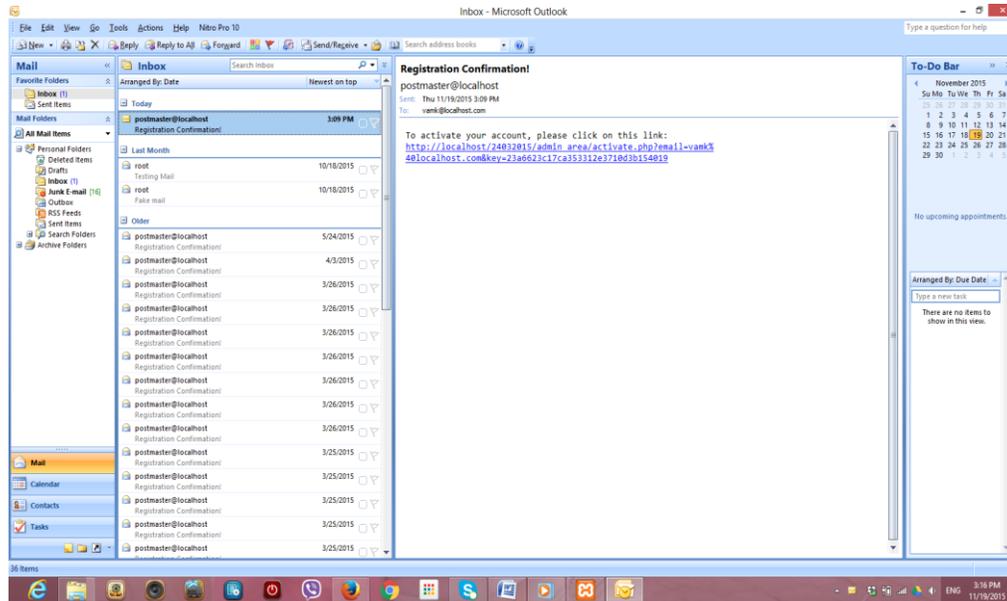


Figure 24: Email received in the new administrator Email box

The figure below shows the data in the “admins” table. We can see clearly that the account is not activated yet.

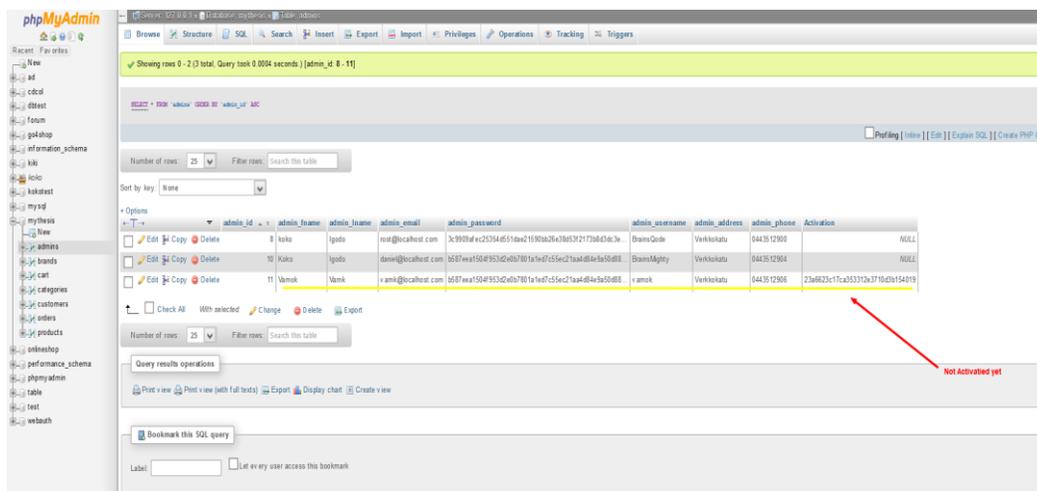


Figure 25: New administrator data in the table, not activated yet

In the database it also shows that the account is activated as shown below.

The screenshot shows the phpMyAdmin interface for a MySQL database named 'admins'. A query has been executed, displaying a table with the following data:

admin_id	admin_fname	admin_lname	admin_email	admin_password	admin_username	admin_address	admin_phone	Activation
8	Ikele	Igodo	root@localhost.com	3c5903afe23354451dae21590ba26e384530217268c36c4e	BrainsCode	Verleketu	0443512900	MULL
10	Kole	Igodo	daniel@localhost.com	b587ee51504f953c2e7b78011aed7c55e21ea4404e9e50480	BrainsMighty	Verleketu	0443512904	MULL
11	Vamok	Vamok	vamok@localhost.com	b587ee51504f953c2e7b78011aed7c55e21ea4404e9e50480	vamok	Verleketu	0443512906	account activated

A red arrow points to the 'Activation' column of the third row, which contains the text 'account activated'.

Figure 267: New Administrator Account activated in the database

- **Administration login page**

The administrator will be able to login with his confidential information (email and password).

THIS IS THE ADMIN LOGIN PAGE

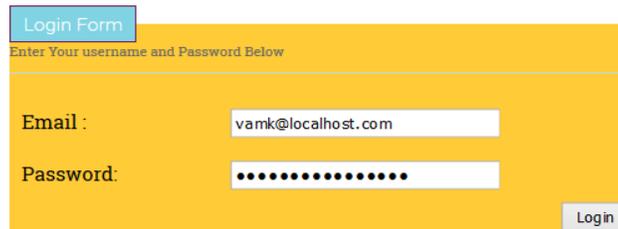


Figure 278 shows a screenshot of the administrator login page. The page has a yellow background. At the top left, there is a blue box labeled "Login Form". Below it, the text "Enter Your username and Password Below" is displayed. There are two input fields: "Email" with the value "vamk@localhost.com" and "Password" with a masked password represented by dots. A "Login" button is located at the bottom right of the form.

Figure 278: Administrator login page

- **Administration Management page**

When email and password provided by the administrator are correct, the php file named "login_admins.php" will send the request to the database server which will check the information, if everything is correct, it will redirect the administrator to the "login_success.php" file.

On this page the administrator will be able to insert, add, edit, delete, and update products information. He will be able to view, edit, and update customers' information. He will be able to view orders and view customers' payments.

The administrator will be able to process the orders made by customers. By confirming the orders, a customer will be able to get the products ordered.

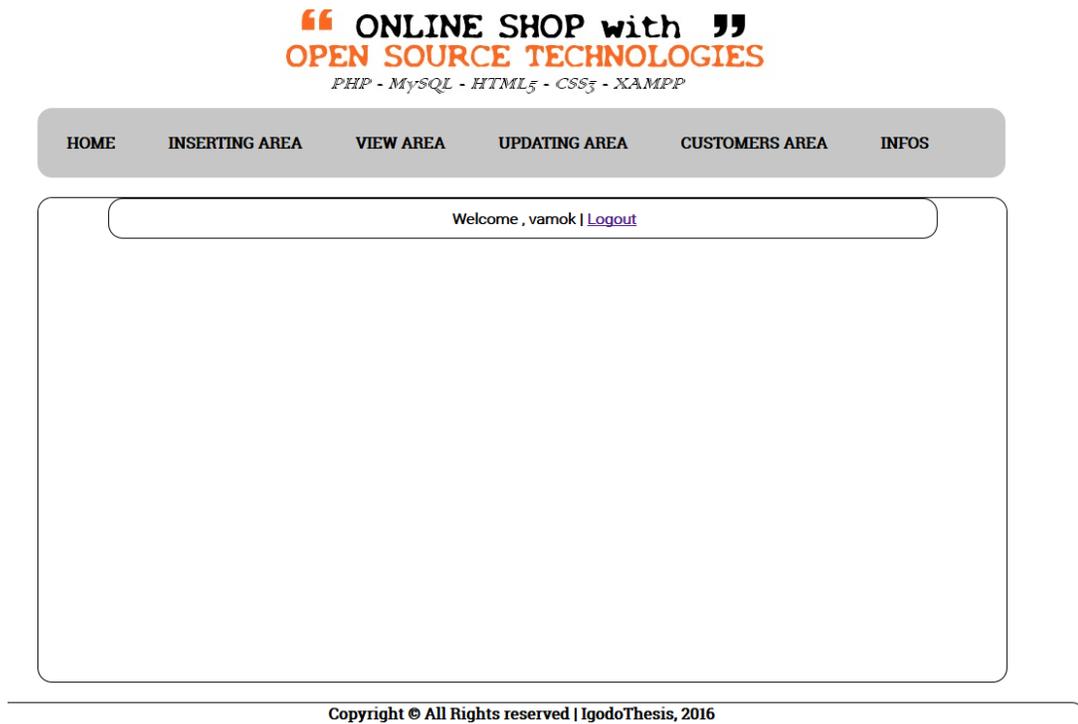


Figure 29: Online shop management page (Administration page)

6 IMPLEMENTATION

This chapter contains the functions implementation codes.

6.1 Description

Our online shop system is based on open source applications and in order to make it work. All the functionalities need to be done in an orderly. The customer or user interfaces are important as the administrator interface. As the whole project is based on HTML5, CSS3, PHP and MySQL technologies, which were focused for the implementation of the project. All the codes are explained below.

6.2 Implementation of Functions

The implementation can be divided into two main blocks: customer interface and administrator interface. The user interface is implemented with HTML5 and for styling we used CSS3. In this part, we will mainly explain the code for the implementation of the pages, of the functions used in the whole project.

6.2.1 Setting up Mercury Mail Transport.

As we are working on a local server we need a mail server in order to send mail from any CMS placed in Xampp. Mercury is already a mail server included in Xampp server so the only thing needed here is to configure it.

When registering or adding any administrator, some information needs to be added like First name, last name, email and password etc.

The email and the password are really important because the administrator will use them to login into the system. After registering a confirmation email is sent automatically to the administrator's email provided during the registration. And this can be done only if there is a mail transport server.

We also need a mail client in order to receive mail from the local host. And in this project we are going to use outlook express as it is the default client mail available in the windows system.

First of all, we need to launch the Xampp application. After that we start the applications: apache, MySQL and also Mercury as shown on the picture below.

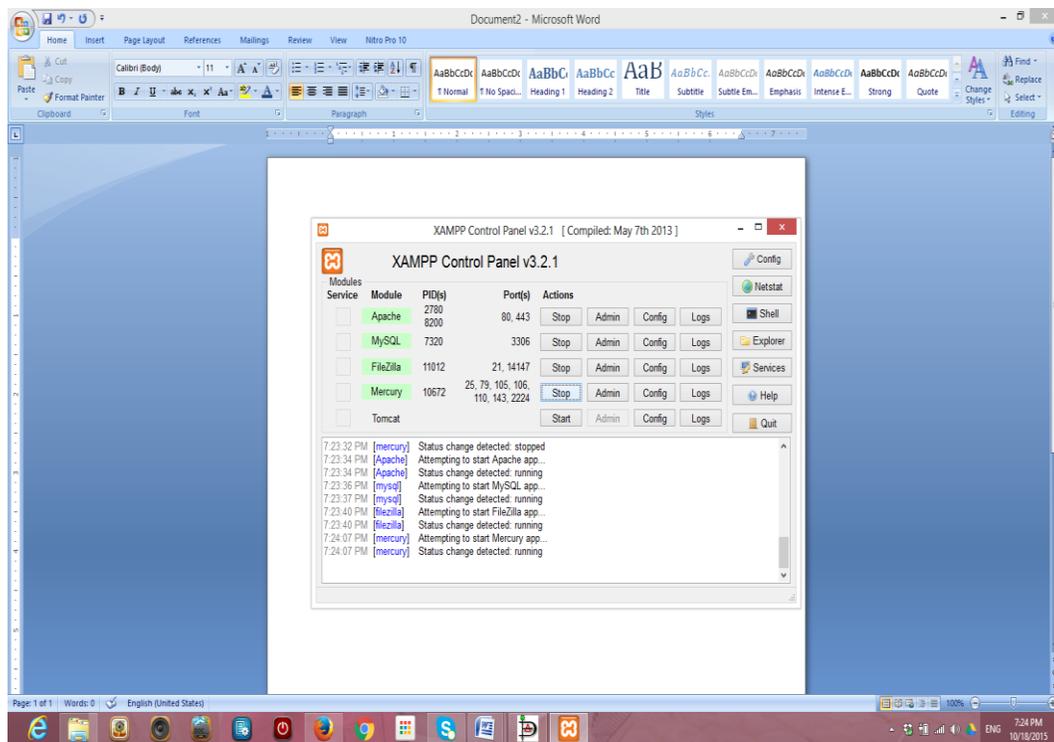


Figure 28: Running Xampp, Mercury, Apache and MySQL

Now we click on the Admin button that is next to Mercury and it will open many windows as shown on the figure below.

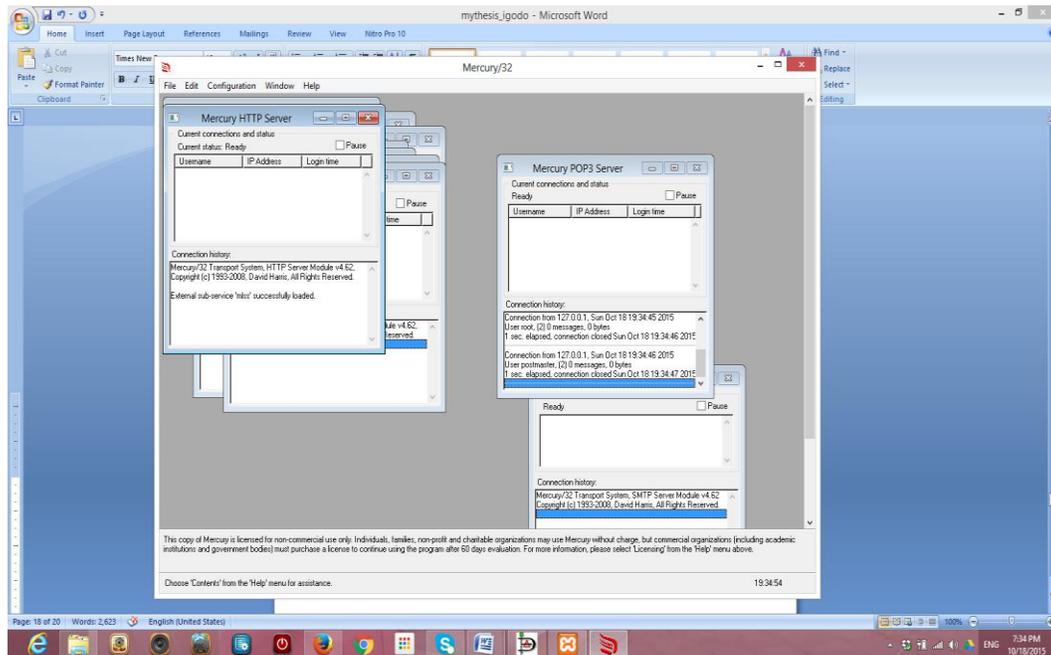


Figure 29: Admin Panel of Mercury Mail Server

From the admin panel, we click on the Configuration (Menu) and from the sub menu we click Manage local users. It will show all the users defined in the system. We can now add the first user with these details:

Username: vamk

Password: vamkpass

Now we need to click Configuration again and choose Mercury SMTP Server.

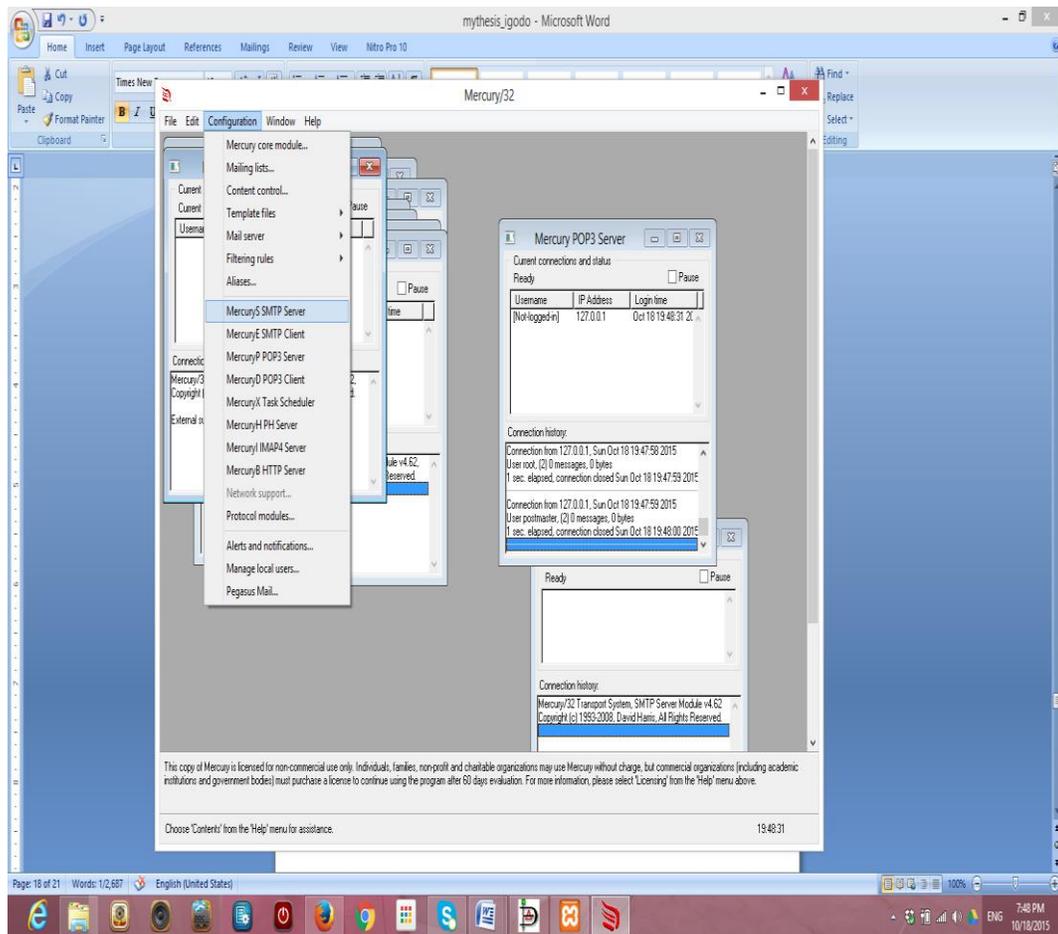


Figure 30: Mercury Configuration Sub Menu

We configure the Mercury SMTP Server with these details as shown:

IP Interface to use: 127.0.0.1

Listen to TCP/IP port: 25

Announce Myself as: 127.0.0.1

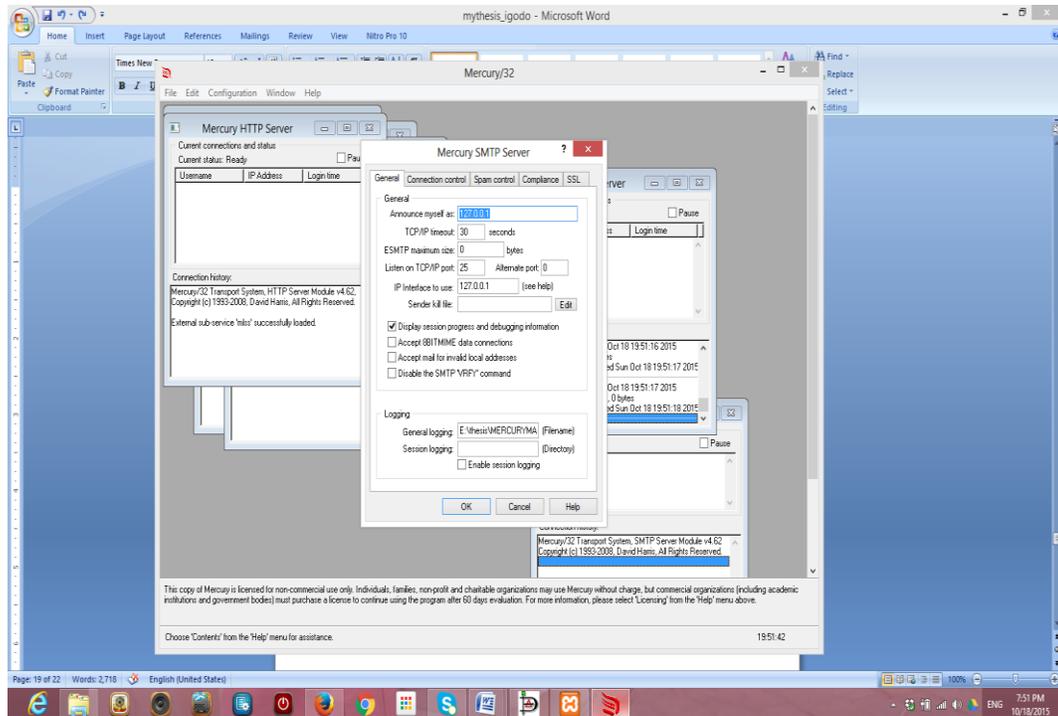


Figure 31: Mercury SMTP Server

We do the same with the Mercury POP Server by clicking Configuration → Mercury POP Server with these details:

Listen to TCP port: 110

IP Interface to use: 127.0.0.1

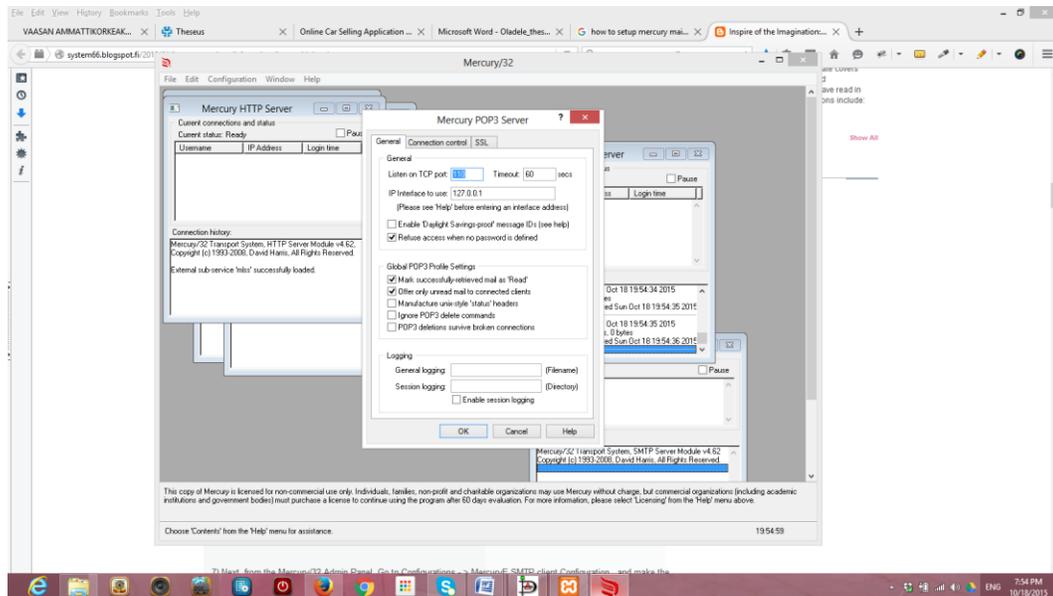


Figure 32: Mercury POP Server Configuration

From Mercury Admin panel we now configure MercuryE SMTP Client with these details:

Identify Myself as: 127.0.0.1

Name server: 127.0.0.1

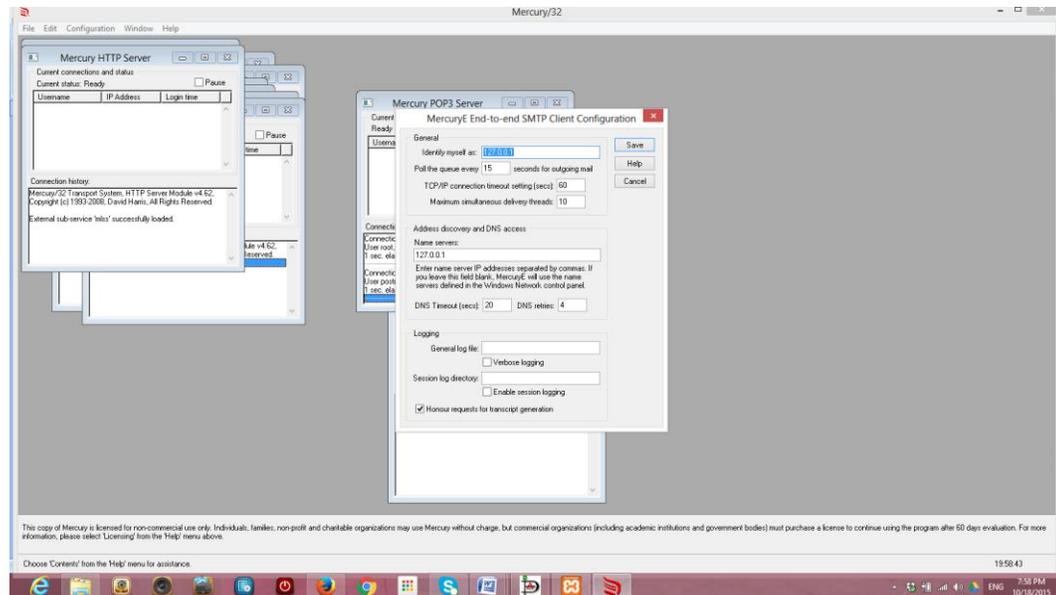


Figure 33: MercuryE SMTP Client

Next step is to configure MercuryD POP Client by creating a new account with these details and we save it:

POP3 Host: 127.0.0.1

Username: vamk

Password: vamkpass

The Mercury transport system on Xampp is ready to be used.

Now we are going to configure Microsoft Outlook.

After launching Microsoft Outlook, we go to Tools->Account Settings and we add new Account.

We configure manually Microsoft Outlook as shown:

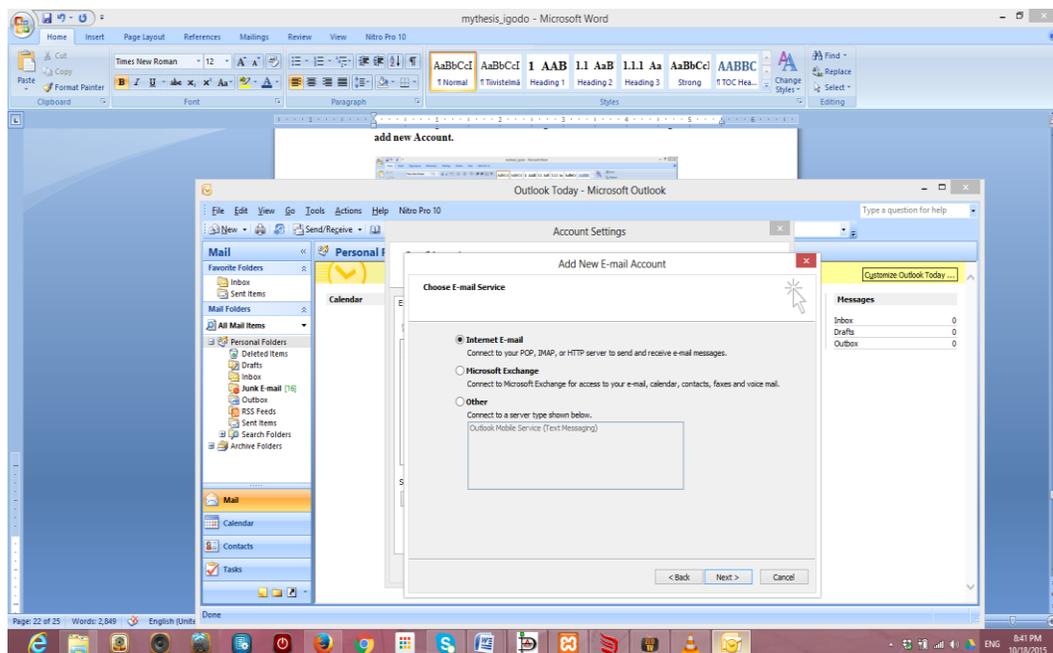


Figure 354: Email address creation

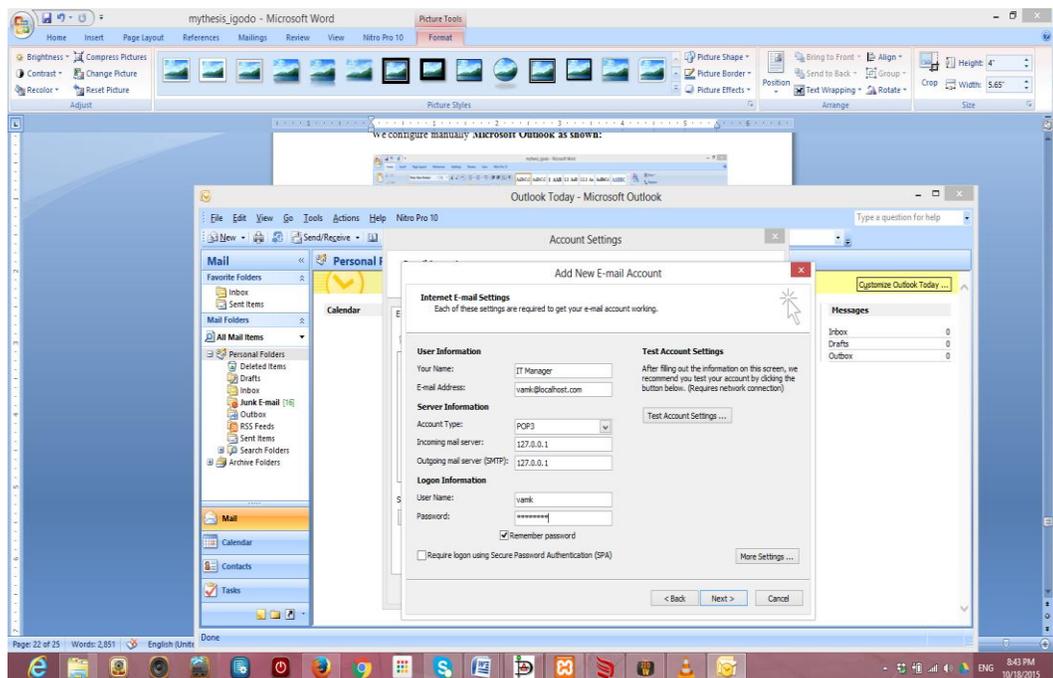


Figure 345: Setup Email address

The following step is to send a fake mail to check if everything is well configured.

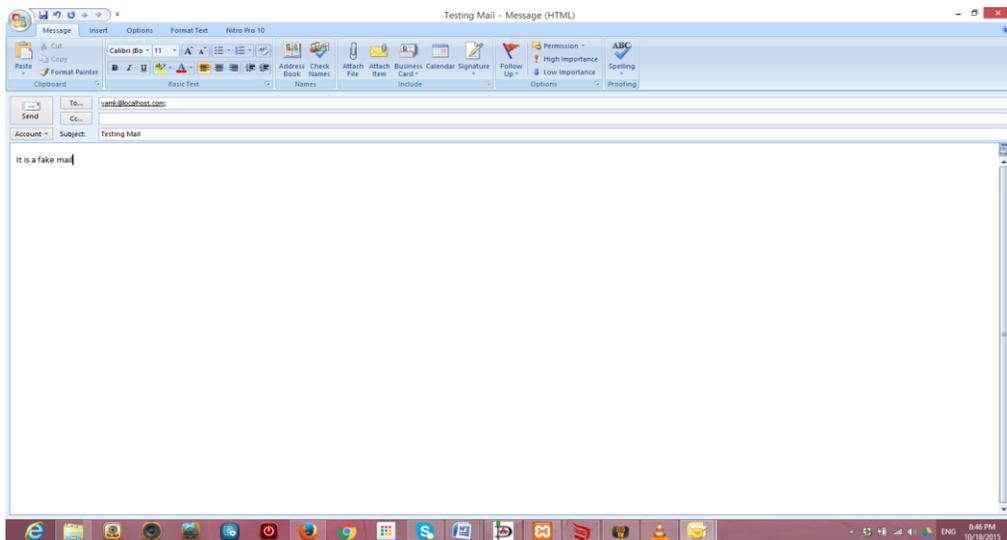


Figure 36: Fake email sending

Mail sent is received in the inbox.

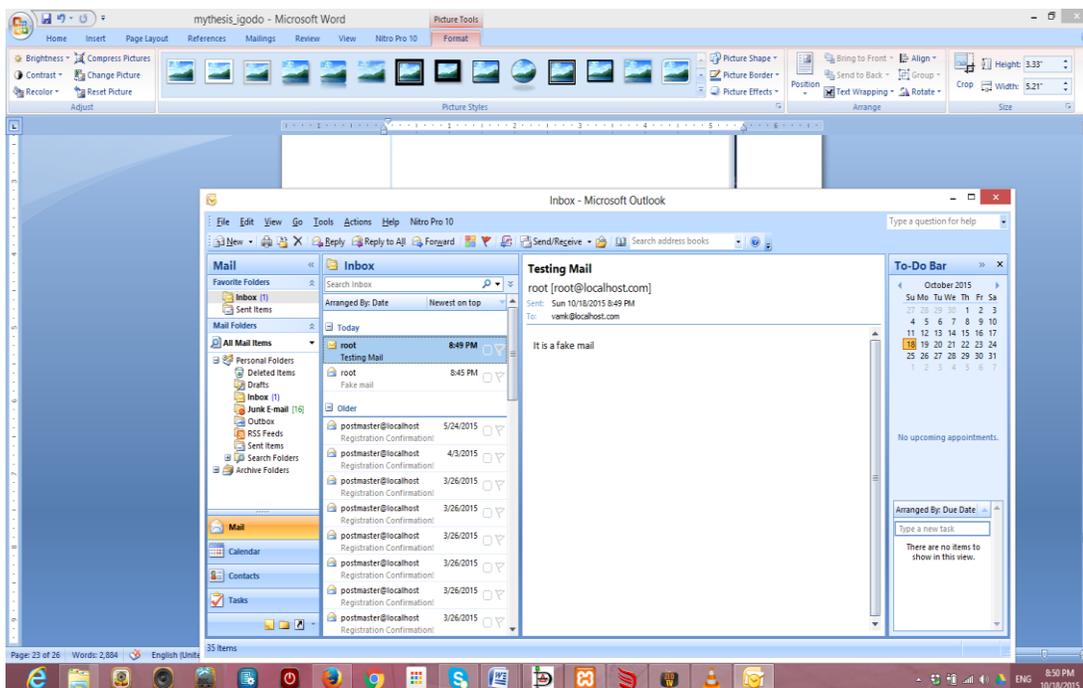


Figure 37: Fake email received

For the implementation of all the online shop pages, we used `$_SESSION` that is an array that stores information across pages requested by the user. Each user accesses the page with his own assigned session. The session is stored on the server.

Below code explained how we used open session in our online shop.

```
<?php

session_start();

$_SESSION["user"] = $row[customer_email];
```

Snippet 1: Customer Session

`session_start()` starts the connection between the server and the user and the values stored in the session can be accessed by other pages and scripts. Data can be retrieved from session.

The code below showed how to close session.

```
<?php

session_start();

unset($_SESSION["user"]);

session_unset();
```

Snippet 2: Session closing

`session_unset()` deletes all the session data. But to destroy the session we used `session_destroy()`, It destroy the whole session because `session_unset()` delete only the data stored.

6.2.2 Available products in the system

On the index page of our project, any user can browse or search products. All the products are showing by category, or by brands. On this page the user can even see the feature products of the week. As the products available in the system are stored into the database, we need to open the connection of the database with the confidential information before having access to it.

We created a function called “getprod()” and within that function as the connection to the database already made, we retrieved from the table “products” the available products with the statement:

We limited the number of the products to display to 12, and also with display randomly as shown in the code.

```
$get_pro = "select * from products order by RAND()
LIMIT 0,12";
```

In the next step we chose from the tables the attributes that we want to display from a single product by using the code below

```
$run_pro = mysqli_query($dbc, $get_pro);
while($row_pro=mysqli_fetch_array($run_pro)){
$pro_id = $row_pro['prod_id'];
$product_cat = $row_pro['product_cat'];
$product_brand = $row_pro['product_brand'];
$pro_name = $row_pro['prod_name'];
```

```
$pro_price = $row_pro['prod_price'];
```

```
$pro_image = $row_pro['prod_image'];
```

Snippet 3: Product attributes from database

We will use "echo" to display the products and attributes on our online shop as shown below.

```
echo "

<div id='single_product'>

    <img src='admin_area/product_images/$pro_image'
width='210' height='150' />

    <div id='leprix'><h3> $ $pro_price </h3></div>
    <div id='letitre'>$pro_name</div><br />

    <a href='details.php?pro_id=$pro_id' id='more_info'><i
class='fa fa-info-circle'></i>More Info</a>

    <a href='index.php?add_cart=$pro_id'><button
id='button_single'><i class='fa fa-shopping-cart'></i> Add to
Cart</button></a>

</div>";
```

Snippet 4: Code for product display

6.2.3 User Registration

Any user can search products and add also add products to the cart but need to register before checking out. For that we have the registration page form. Through the form, the user will fill the fields with his/her personal information. After that he will click on the button “Register”.

There is a PHP file called “customer_registration.php” that will process all the data before sending it to the database (table “Customers”).

The customer will use his email and password provided when registering, so as the attribute “customer_email” should be unique in the table, before inserting the user data into the table, the PHP file will check if there is any customer with the same email provided by the user. In case there is no customer with the same email, the user data will be inserted into the database (table “Customers”) and will be redirected to customer page or checkout page if his already has products in the shopping cart, if there is a customer with that email, the user will be notified to change the email or to login with his confidential information if he already customer.

```

    if (isset($_POST['insert_customer'])) {

        // $error = array(); // Declare An Array to store any
        error message

        $ip = getIp();

        $fname = $_POST['fname']; // else assign it a variable

        $lname = $_POST['lname']; // else assign it a variable

```

```
$Email = $_POST['email'];//else assign it a variable

$Password = $_POST['Password'];//else assign it a
variable

$username = $_POST['username'];//else assign it a
variable

$country = $_POST['country'];//else assign it a
variable

$address = $_POST['address'];//else assign it a
variable

$postcode = $_POST['p_code'];//else assign it a
variable

$city = $_POST['city'];//else assign it a variable

$phone = $_POST['phone'];//else assign it a variable

$customer_image = $_FILES['customer_image']['name'];

$customer_image_tmp =
$_FILES['customer_image']['tmp_name'];

move_uploaded_file($customer_image_tmp, "./customer/cu
stomer_images/$customer_image");

if (isset($_POST['insert_customer'])) {

//$error = array();//Declare An Array to store any
```

Snippet 5 : User Registration function

6.2.4 Customer Login

For the customer to access the customer page or to checkout, he needs to provide his email and password, the HTML5 validation on the page will validate the fields data and then the PHP file will first check in the database (table “Customers”), if the inputs provided by the customers are the same with the one in the database. If it is the same the customer will be redirected to the customer page.

```
$email = $_POST['email'];

>Password = $_POST['Password'];

$hashPassword = hash('sha512',$Password);

$sel_customer = "select * from customers where
customer_email='$email' AND
customer_password='$hashPassword'";

$run_customer = mysqli_query($dbc, $sel_customer);

$num_rows = mysqli_num_rows($run_customer);

//echo $num_rows;

$row = mysqli_fetch_assoc($run_customer);
```

Snippet 6: Customer Login checking

```
$ip = getIp();

$sel_cart = "select * from cart where ip_addr='$ip'";

$run_cart = mysqli_query($dbc, $sel_cart);

$check_cart = mysqli_num_rows($run_cart);

if($row['customer_password']==$hashPassword AND
$check_customer>0 AND $check_cart==0){

    $_SESSION['customer_email'] = $email;

    echo

"<script>window.open('./customer/my_account.php','_self')<
/script>";

}
```

Snippet 7: Code for accessing customer account

7 TESTING

In order to verify the quality of the online shop, tests have been made, at different levels of the project; test on the component, and also to check the reliability of all the functions. A test on the integration system has been made too.

Tests are made on all the customer and administration modules. As the project is mainly done in PHP, all the modules PHP files are tested. The customer test shown here will be the customer login, account and shopping cart pages' tests and the administrator modules tests shown here will be admin login, customer and products management pages' tests.

7.1 Customer Shopping Cart Testing

The PHP file *cart.php* is responsible of adding product to the cart. It gets all the product parameters (product name, quantity) from the database as the *product_id* is known, and calls the function *cart ()* which adds the product to the cart database with the customer ip address. After add the product to the cart database, its uses *view_cart ()* function to display the product on the shopping cart page.

On the shopping cart page, the customer can delete a product, and *delete_pro()* function is responsible of deleting the product in the cart, also for quantity updating, *updatecart()* function is used to update the product quantity from the cart database.

7.2 Customer Login Testing

This test is done to be sure that the login function is implemented well, *customer_login.php* is responsible page for customer login. We will see how the function will react when different inputs are into the login forms and the customer click on Login button.

- The input email does not exist

When the customer forgets to write the email address and click the login button, a message is shown to him/her (*please fill out the field*), At this level we used HTML 5 validation function. It is a client side validation. It does not use JavaScript.

The test result:

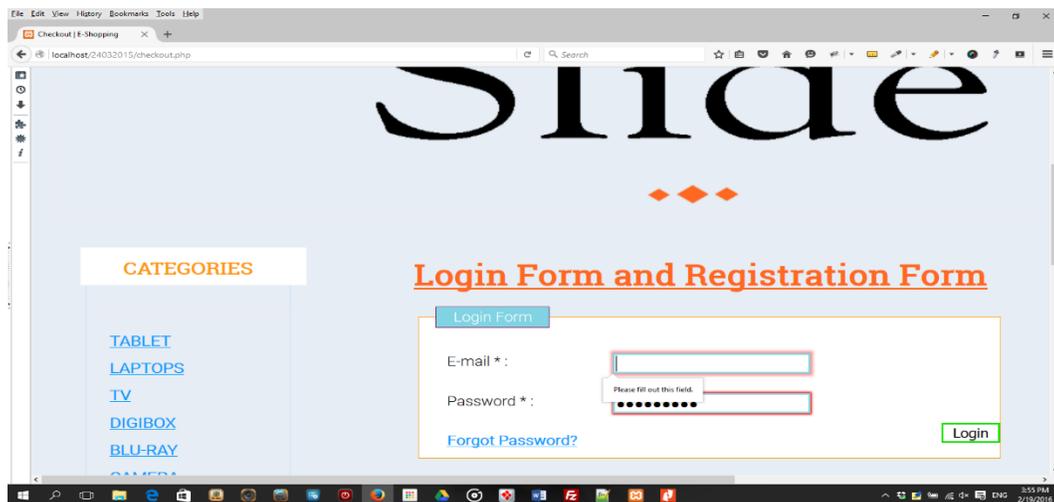


Figure 38: Test customer login result

- Customer Login test with input but not email form

When the customer writes an input in the email field but not in the form of email(xyz@xyz.com) and click on the login button, an error message is also shown as show in the figure below

The test result:

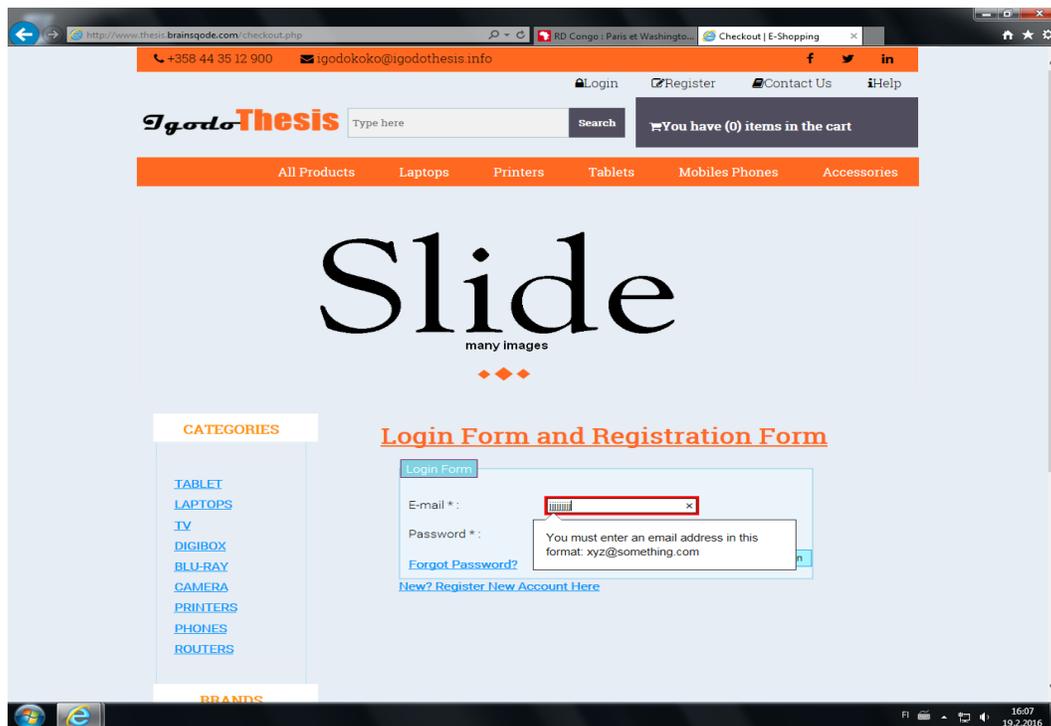


Figure 39: Customer login test 2

7.3 Administration Login Testing

The administration testing is the same as the customer test.

- Login with no email or password inputs

The test result

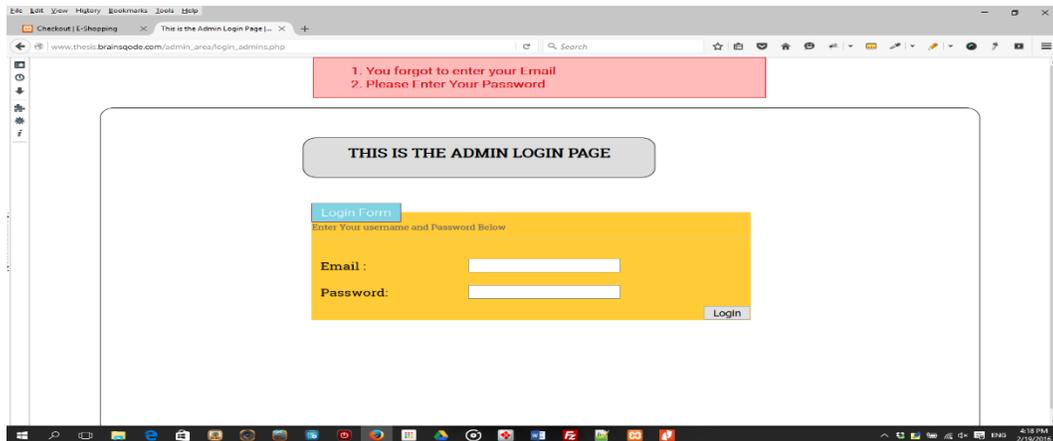


Figure 40: Administrator login test 1

7.4 Administrator Login Successful page

The test result

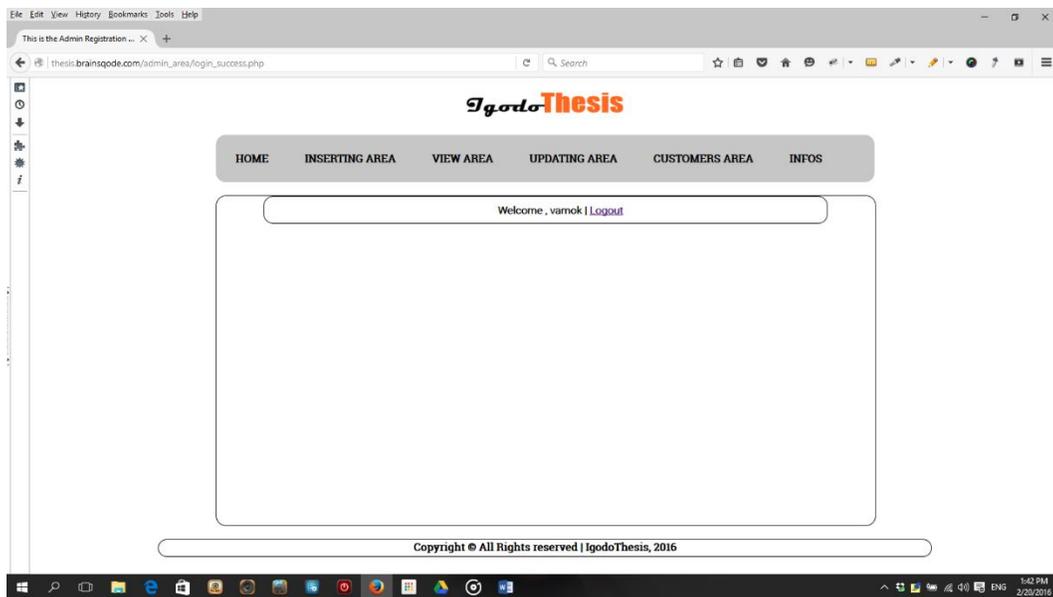


Figure 41: Administrator Login success page

- Administration page: Product Inserting page

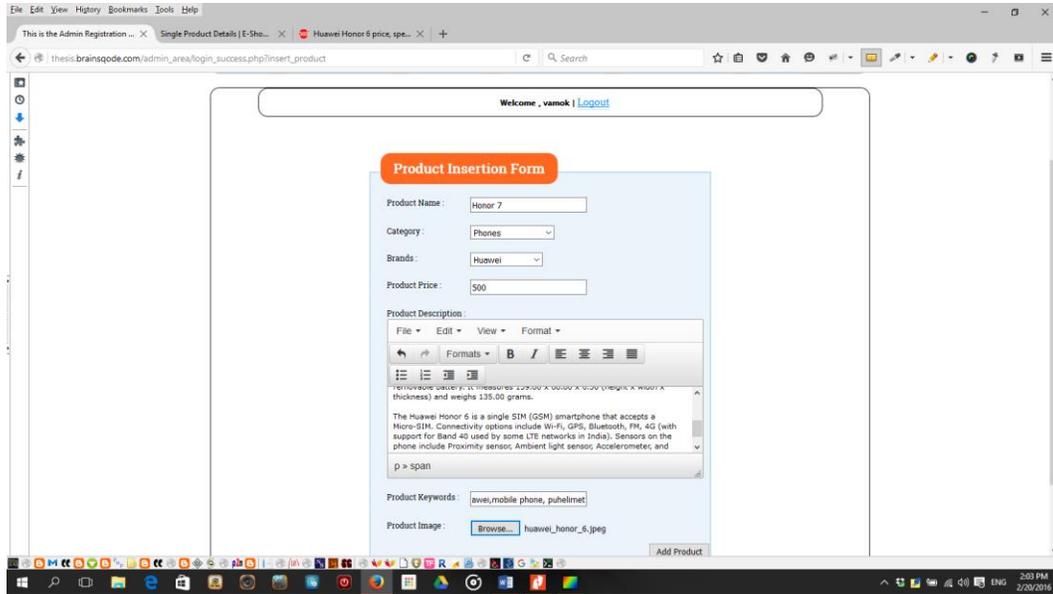


Figure 42: Product Inserting test

8 CONCLUSIONS

An online shop for electronics devices was developed using PHP, MySQL, HTML5 and CSS3 technologies. Any user can browse products, add, update or delete products from the cart. To be able to make a payment, the user needs to login, so he needs to sign up first of all with his personal details as well as his email address, password and address. Those data are saved in MySQL database.

After logs in successfully, user has access to own account where own information can be updated, see own orders and payments, change own password. Payments can be made for the products in the shopping cart using PayPal.

After making the payment with PayPal, the user will receive an automatic email message about his purchase. The administrator needs to confirm the order but the user can still see the status of own orders in own account. The orders and payment are stored also in a MySQL database.

Everything was done successfully based on requirements.

8.1 Future Work

For future work, administrators have to be divided into different groups; maintenance management group (department) who will only maintain the website, products management group (department) who will only edit, update and delete products, Order management group (department) who will only check all the orders and process them, invoice and payment management group (department) who will only manage orders payments and invoices.

Each group will have certain rights based on needs of each department. The security aspects were not developed in this project. As security in online shop is also very important and I think if this project will be used, it would need improvements on database securities and server.

REFERENCE

/1/ Understanding the proper way to lay out a page with html5. Accessed 11.11.2015

<http://www.developer.com/lang/understanding-the-proper-way-to-lay-out-a-page-with-html5.html>

/2/ CSS3 feature and attributes. Accessed 10.12.2015

www.w3school.com

/3/ PHP functions. Accessed 08.03.2016

<http://php.net/manual/en/book.session.php>

/4/ Javascript tutorial. Accessed 09.03.2016

<http://www.w3schools.com/js/default.asp>

/5/ Documentation on MySQL Database. Accessed 05.01.2016

<https://www.mysql.com/>

/6/ UML concept. Access 10.02.2016

https://en.wikipedia.org/wiki/Unified_Modeling_Language

/7/ XAMPP. Access 15.03.2016

<https://www.apachefriends.org/index.html>

/8/ Mercury Mail Transport. Accessed 20.02.2016

http://www.pmail.com/overviews/ovw_mercwin.htm

/9/ PhpMyAdmin. Accessed 20.02.2016

<https://www.phpmyadmin.net/>

/10/ Notepad++. Accessed 20.02.2016

<https://notepad-plus-plus.org/>

/11/ MySQL Workbench

<http://dev.mysql.com/downloads/workbench/>

/12/ PayPal developer and Sandbox account. Accessed 20.01.2016

www.paypal.com

/13/ MVC Diagram explanation. Accessed 25.01.2016

https://commons.wikimedia.org/wiki/File:MVC_Diagram_%28Model-View-Controller%29.svg