Hieu Vo

Unix project

Default project

Helsinki Metropolia University of Applied Sciences
Smart Systems

Unix operating system course

16.03.2016

Contents

1 Introduction

This project is a part of Unix operating system course. The goal for the project is to figure out an application for Unix in creating a website.

At first the students decided what is the project that they can do, which is related to Unix operating system. After choosing the topic, students must send a proposal project and waiting for the teacher to reply.

If the answer is yes, then they can continue to work with their project. Alternatively, if the students do not have any ideas about Unix operating system yet, then they can choose the default project which is given by the teacher.

The preliminary plan for the project is to start with research and at the same time do the labs assigned on the lessons. No further plan was discussed at this point because of the little knowledge about the subject, so I need to do more research.

Later on, when I realized that this default project is mostly related to building a web site, I focus on making a web first then find the way to relate it to Unix operating system. There were a lot of difficulty because I do not have any background on building web or using Unix operating system. Therefore, I did it as much as I could research step by step in my ability.

It might not fulfill all the expectation of this default project; however, I try to response to all the basic requirement for this project as much as I can and I will discuss it more detail after this introduction part.

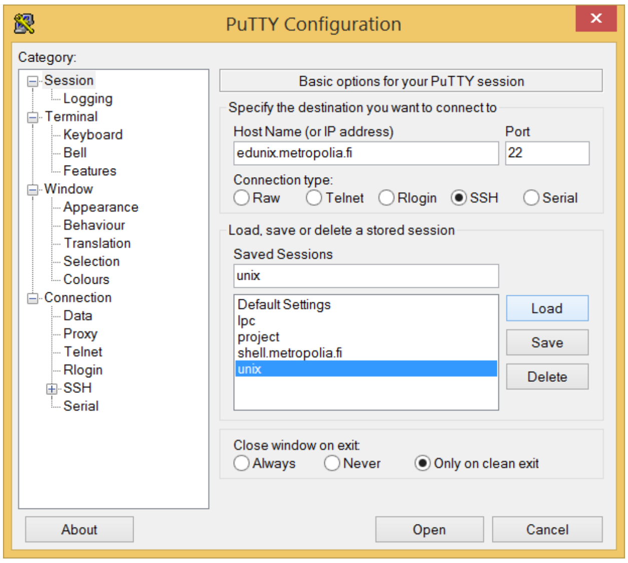
My own aims for this project is that I will build the web first through all environments that the school have supplied for me. Then I will try to make some command and bash script to do some specific tasks. This will show that how the shell scripts do something useful for some specific tasks as the project required.

2 Initialization

In this initializing part, I need to check up for many part before building a simple website. The LAMP (Linux Apache MySQL PHP) need to be checked. As I do not have much knowledge about Unix operating system, so I use WinSCP and web page to interact with the database. In addition, I tried to using Unix environment as much as I can if it is possible.

2.1 Checking LAMP (Linux Apache MySQL PHP)

To using Unix environment, Putty is an appropriate software to connect with the server. Next step is to access Metropolia Linux server edunix.metropolia.fi with port 22 and connection type is SSH.



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): edunix.metropolia.fi

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

Saved Sessions: unix

Default Settings
lpc
project
shell.metropolia.fi
unix

Close window on exit: ○ Always ○ Never ● Only on clean exit

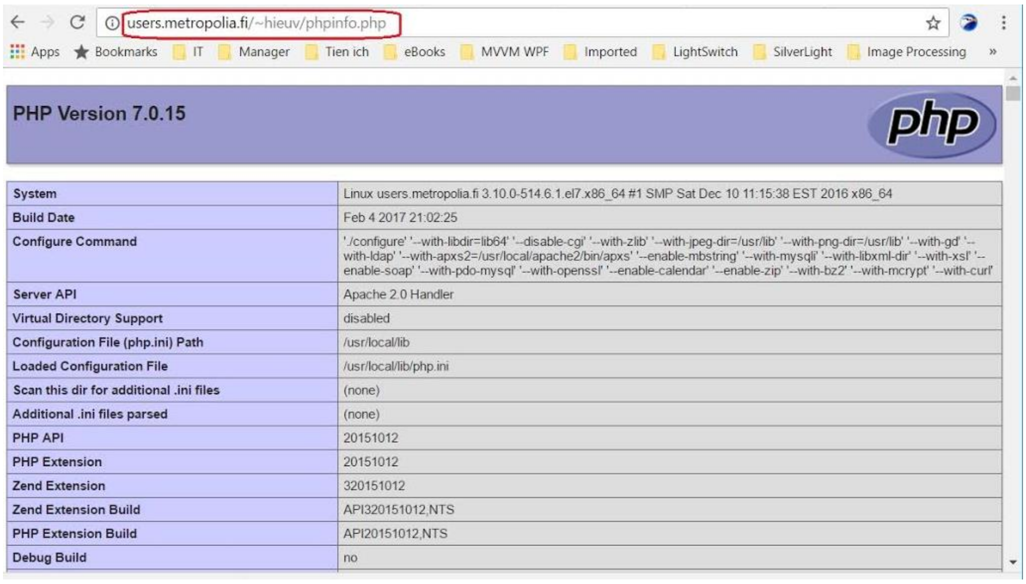
Figure 1: Putty configuration

Parallel with using putty, an easier software to interact and control is WinSCP. I could download it from school website and login with my user and password. Then I can already access to my folder at school and manage the public_html folder, which is for my webpage.

To test how my public_html work, I write some simple code and put it in the public_html to check.

According to the default project, I can use the school environment and server for MySQL and phpMyAdimin.

Furthermore, because I want to check more detail on the PHP server, I access by the link: users.metropolia.fi/~hieuv/phpinfo.php



users.metropolia.fi/~hieuv/phpinfo.php

PHP Version 7.0.15

System	Linux users.metropolia.fi 3.10.0-514.6.1.el7.x86_64 #1 SMP Sat Dec 10 11:15:38 EST 2016 x86_64
Build Date	Feb 4 2017 21:02:25
Configure Command	'./configure' '--with-libdir=lib64' '--disable-cgi' '--with-zlib' '--with-jpeg-dir=/usr/lib' '--with-png-dir=/usr/lib' '--with-gd' '--with-ldap' '--with-apxs2=/usr/local/apache2/bin/apxs' '--enable-mbstring' '--with-mysqli' '--with-libxml-dir' '--with-xsl' '--enable-soap' '--with-pdo-mysql' '--with-openssl' '--enable-calendar' '--enable-zip' '--with-bz2' '--with-mcrypt' '--with-curl'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/lib
Loaded Configuration File	/usr/local/lib/php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20151012
PHP Extension	20151012
Zend Extension	320151012
Zend Extension Build	API320151012,NTS
PHP Extension Build	API20151012,NTS
Debug Build	no

Figure 2: PHP server

With this, I can check all the information from Apache to PHP and now I am quite certain that it will works well.

3 Building project

Because this is a Unix project and Unix course, I am not focus on discussing deeply about web and writing web code. However, I still talk about it briefly and point out all the important issues and files, which is used to build the website.

3.1 Constructing website.

First, I created a simple product table model. In phpMyAdmin, I create a product table with column: name, description, price and action.

As I do not have any background on web, to do it faster, I borrowed the code from a website: code.tutsplus.com to build a shopping cart, from HTML guide form.com to build register form and from sitepoint.com to create role base access control. Then I read to understand what is the function of the code and edited it so that it will compatible with the Metropolia server.

All the important files and directories are listed below:

The file index.php is the one will be open when I access to the link user.metropolia.fi/~username. It acts like a template for the file products.php and cart.php to run. In this file, later on, I decided to extend it to control the users' privilege and category depending on the level of the users, who sign in to the website.

File products.php display the list of products for the website, which can be sort in category.

File cart.php has the basic function of a shopping cart.

File users.php which to display the list of the users (if user sign in as admin -> level 0) and allow the admin to adjust the level of the other users (give more or less right to the other users).

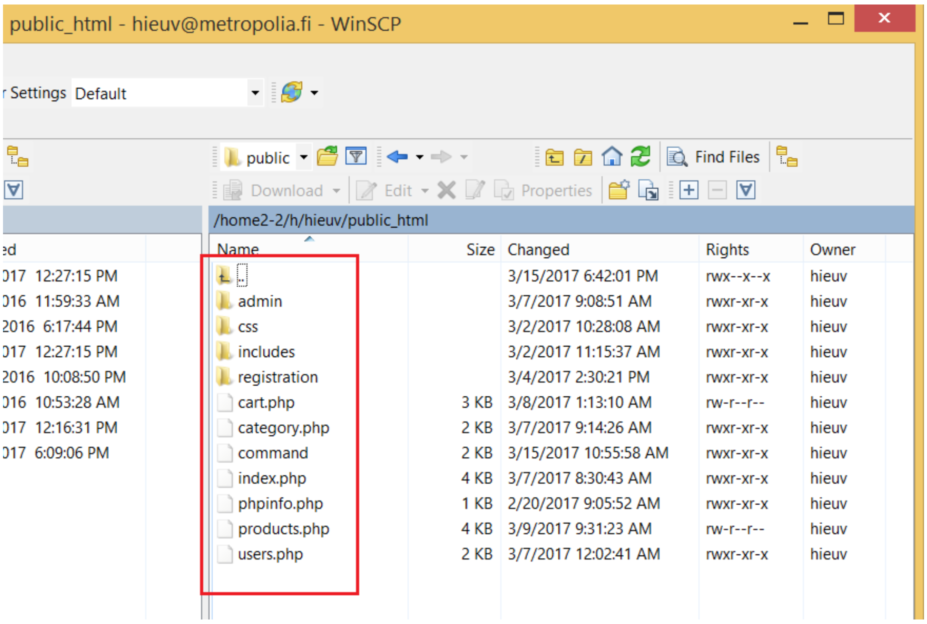
File catergory.php is used to organize the products. However, there is no function delete a category because after you delete it, the product may lose that category and not belong to anywhere.

The directory administrator contains the php files, which is needed to manage the product, category, user (add, edit, delete) (if there is administrator privileges)

Register the directory is used to manage the registration, login, logout and password changes

The directory includes the files needed to connect to MySQL

Css collection which contains css style files to format html tags for clear, good design.



public_html - hieuv@metropolia.fi - WinSCP

/home2-2/h/hieuv/public_html

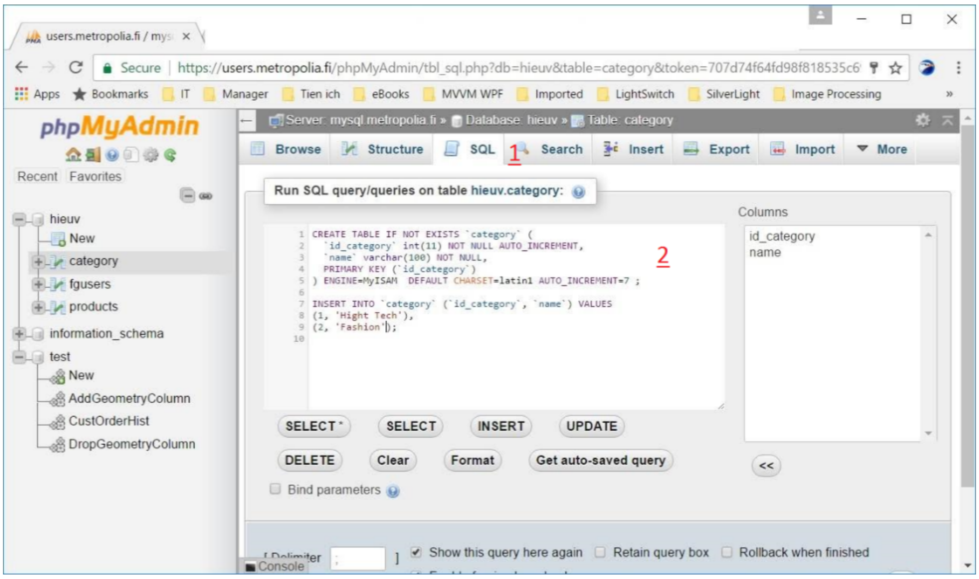
Name	Size	Changed	Rights	Owner
..		3/15/2017 6:42:01 PM	rwx--x--x	hieuv
admin		3/7/2017 9:08:51 AM	rwxr-xr-x	hieuv
css		3/2/2017 10:28:08 AM	rwxr-xr-x	hieuv
includes		3/2/2017 11:15:37 AM	rwxr-xr-x	hieuv
registration		3/4/2017 2:30:21 PM	rwxr-xr-x	hieuv
cart.php	3 KB	3/8/2017 1:13:10 AM	rw-r--r--	hieuv
category.php	2 KB	3/7/2017 9:14:26 AM	rwxr-xr-x	hieuv
command	2 KB	3/15/2017 10:55:58 AM	rwxr-xr-x	hieuv
index.php	4 KB	3/7/2017 8:30:43 AM	rwxr-xr-x	hieuv
phpinfo.php	1 KB	2/20/2017 9:05:52 AM	rwxr-xr-x	hieuv
products.php	4 KB	3/9/2017 9:31:23 AM	rw-r--r--	hieuv
users.php	2 KB	3/7/2017 12:02:41 AM	rwxr-xr-x	hieuv

Figure 3: Important files and folder in public_html

3.2 Creating database

In this section, I take one example from create a products table to briefly demonstrate. The other table: products and fgusers were created almost the same way.

As the figure, first, click on the SQL tab to open the window. After that, at number 2, write the codes which construct the database table. You can also enter some initial data for the table. Finally, after everything is ready, click “Go” at the right corner of the window to generate the table.



Run SQL query/queries on table hieuv.category:

```
CREATE TABLE IF NOT EXISTS `category` (
  `id_category` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(100) NOT NULL,
  PRIMARY KEY (`id_category`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=7 ;

INSERT INTO `category` (`id_category`, `name`) VALUES
(1, 'Hight Tech'),
(2, 'Fashion');
```

Figure 4: Creating database and 2 steps.

4 Relating to Unix operating system

After building database successfully, with what I have learnt from the labs and the course, I did some research and write a command and bash script to communicate with the data base, as the project require. It is needed to return list of users, their usernames and passwords and documentation. Below are all commands and scrips, which do all the tasks.

4.1 Bash command lines and script.

4.1.1 Read and write file

Usually, to add a file, I can edit it with cat command. With $ cat > <file name>, I can create and write to any file that I want and also can read from it $ cat <file name>, or to copy file $ cat - file1 > file2. In further point, there is a small issue that to run the script, I need to write the script in the Unix environment. If I do not write it in the Unix environment, the script will not work.

In addition, I need to give the permission for the file with command $ chmod <a+x> <filename> so that I can run that file. Depending on the right that you want to give, you can change the information in the bracket (such as chmod 777 to give all the right).

4.1.2 Accessing database

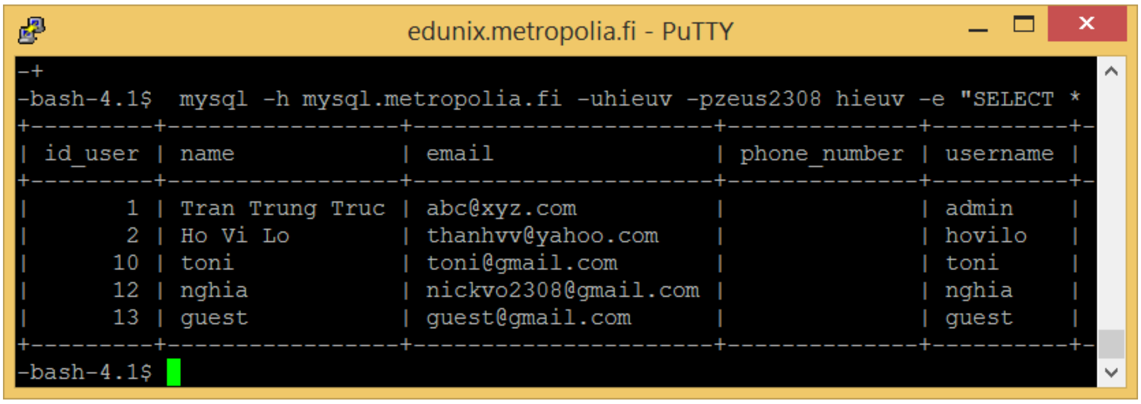
4.1.2.1 Command for extracting small information

I tried with some commands to list all the users' name and information of the user. To list all email and users' name:

```
mysql -h mysql.metropolia.fi -u<username> -p<password> <username> -e
"SELECT name,email,username FROM fgusers;"
```

To list all the information of the user:

```
mysql -h mysql.metropolia.fi -u<username> -p<password> <username> -e
"SELECT * FROM fgusers;"
```



```
-+
-bash-4.1$  mysql -h mysql.metropolia.fi -uhieuv -pzeus2308 hieuv -e "SELECT *
+---------+-----------------+----------------------+--------------+----------+-
| id_user | name            | email                | phone_number | username |
+---------+-----------------+----------------------+--------------+----------+-
|       1 | Tran Trung Truc | abc@xyz.com          |              | admin    |
|       2 | Ho Vi Lo        | thanhvv@yahoo.com    |              | hovilo   |
|      10 | toni            | toni@gmail.com       |              | toni     |
|      12 | nghia           | nickvo2308@gmail.com |              | nghia    |
|      13 | guest           | guest@gmail.com      |              | guest    |
+---------+-----------------+----------------------+--------------+----------+-
-bash-4.1$
```

Figure 5. Extract users’ infomation

4.1.2.2 Add new item.

In this part, I have made simple scripts for adding new item. There are 2 adding script, one add new user and one add new product. However, they are quite similar to each other.

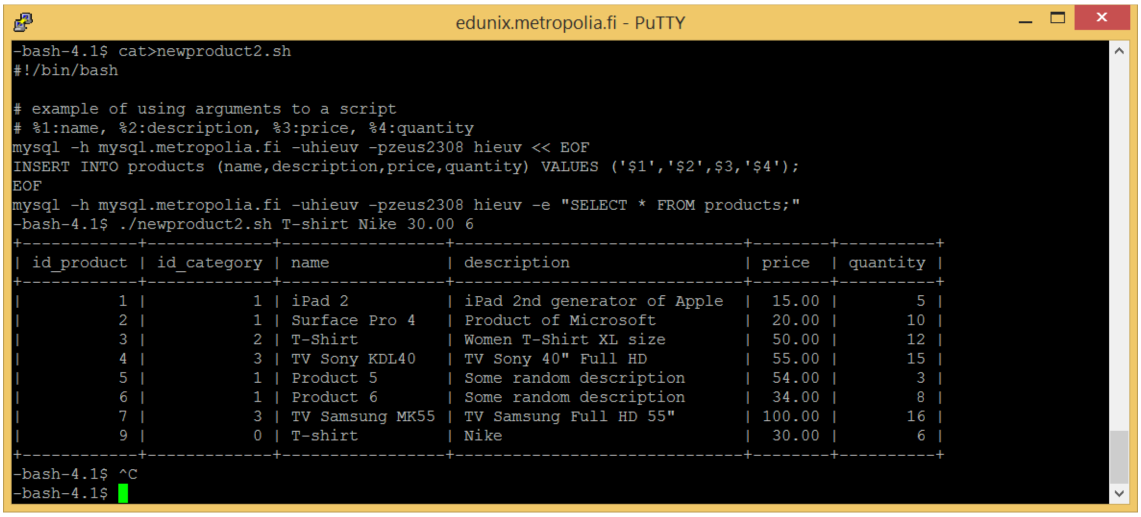
Scripts to add new user:

```
#!/bin/bash

# example of using arguments to a script
# %1: username, %2:password, %3:level, %4:name, %5:email
mysql -h mysql.metropolia.fi -uhieuv -pzeus2308 hieuv << EOF
INSERT INTO fgusers (username,password,level,name,email) VALUES
('$1','$2',$3,'$4','$5');
EOF
mysql -h mysql.metropolia.fi -uhieuv -pzeus2308 hieuv -e "SELECT *
FROM fgusers;"
```

Script to add new product:

```
#!/bin/bash

# example of using arguments to a script
# %1:name, %2:description, %3:price, %4:quantity
mysql -h mysql.metropolia.fi -uhieuv -pzeus2308 hieuv << EOF
INSERT INTO products (name,description,price,quantity) VALUES
('$1','$2',$3,'$4');
EOF
mysql -h mysql.metropolia.fi -uhieuv -pzeus2308 hieuv -e "SELECT *
FROM products;"
```

Below is the result of running adding new product script. Adding new user script got quite same result.



```
-bash-4.1$ cat>newproduct2.sh
#!/bin/bash

# example of using arguments to a script
# %1:name, %2:description, %3:price, %4:quantity
mysql -h mysql.metropolia.fi -uhieuv -pzeus2308 hieuv << EOF
INSERT INTO products (name,description,price,quantity) VALUES ('$1','$2',$3,'$4');
EOF
mysql -h mysql.metropolia.fi -uhieuv -pzeus2308 hieuv -e "SELECT * FROM products;"
-bash-4.1$ ./newproduct2.sh T-shirt Nike 30.00 6
+------------+-------------+-----------------+------------------------------+--------+----------+
| id_product | id_category | name            | description                  | price  | quantity |
+------------+-------------+-----------------+------------------------------+--------+----------+
|          1 |           1 | iPad 2          | iPad 2nd generator of Apple  |  15.00 |        5 |
|          2 |           1 | Surface Pro 4   | Product of Microsoft         |  20.00 |       10 |
|          3 |           2 | T-Shirt         | Women T-Shirt XL size        |  50.00 |       12 |
|          4 |           3 | TV Sony KDL40   | TV Sony 40" Full HD          |  55.00 |       15 |
|          5 |           1 | Product 5       | Some random description      |  54.00 |        3 |
|          6 |           1 | Product 6       | Some random description      |  34.00 |        8 |
|          7 |           3 | TV Samsung MK55 | TV Samsung Full HD 55"       | 100.00 |       16 |
|          9 |           0 | T-shirt         | Nike                         |  30.00 |        6 |
+------------+-------------+-----------------+------------------------------+--------+----------+
-bash-4.1$ ^C
-bash-4.1$
```

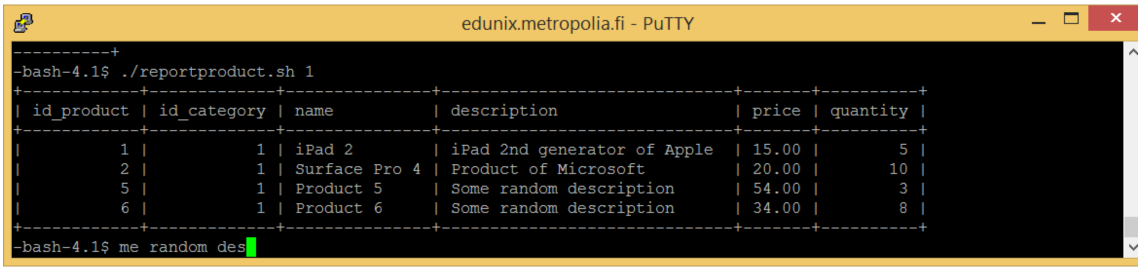
Figure 6. Add new item

The input to run this script is: ./newproduct.sh {name} {description} {price} {quantity}. Depeding on diferent type of item that you want to add (for example: new user or new product) the input is needed to be modified a little, but the structure for it mostly stays the same.

4.1.2.3 Script for making a report

To explore more about how access and list all of the items in product table, I develop the script with if statement. The input to run the script is: ./reportproduct.sh <number>. The number here is the category number. Depending on what number you type in all of the products in that category will be displayed like the figure bellow.

```
#!/bin/bash

# example of using arguments to a script
# $1: id_category
# if there's parameter => filter to id_category=$1
if [ ${#} = 0 ]; then
mysql -h mysql.metropolia.fi -uhieuv -pzeus2308 hieuv -e "SELECT *
FROM products;"
else
mysql -h mysql.metropolia.fi -uhieuv -pzeus2308 hieuv -e "SELECT *
FROM products WHERE id_category=$1;"
fi
```



```
----------+
-bash-4.1$ ./reportproduct.sh 1
+------------+-------------+---------------+------------------------------+-------+----------+
| id_product | id_category | name          | description                  | price | quantity |
+------------+-------------+---------------+------------------------------+-------+----------+
|          1 |           1 | iPad 2        | iPad 2nd generator of Apple  | 15.00 |        5 |
|          2 |           1 | Surface Pro 4 | Product of Microsoft         | 20.00 |       10 |
|          5 |           1 | Product 5     | Some random description      | 54.00 |        3 |
|          6 |           1 | Product 6     | Some random description      | 34.00 |        8 |
+------------+-------------+---------------+------------------------------+-------+----------+
-bash-4.1$ me random des
```

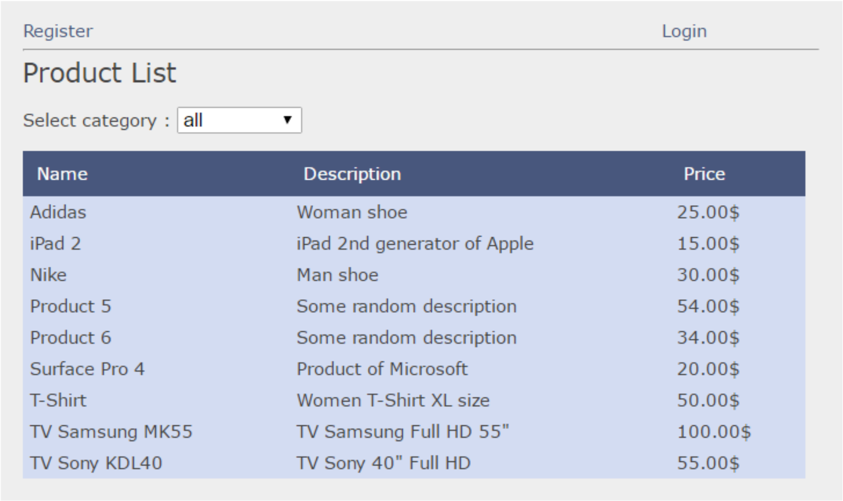
Figure 7. Report list.

5 Instruction and note for web site model

5.1 Instructions

In this project, I used Metropolia environment and server. Therefore, users have to use my school link to access my web site. The link is: http://users.metropolia.fi/~hieuv/

When users access the web site, the users will see a product table. First, user need to register , the register button on the left upper conner. The user highly recommend to use real Metropolia email to register because it will send to user a confirm code and user need to click to the link to activate it. In case user fail in registing, user can use a guest account , username : guest , pasword: password .



Register Login

Product List

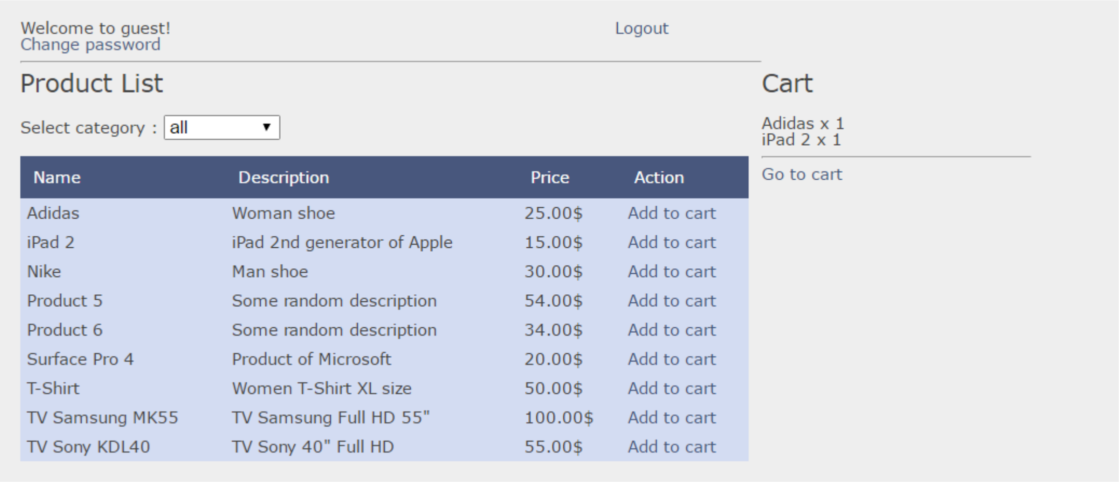
Select category : all ▾

Name	Description	Price
Adidas	Woman shoe	25.00$
iPad 2	iPad 2nd generator of Apple	15.00$
Nike	Man shoe	30.00$
Product 5	Some random description	54.00$
Product 6	Some random description	34.00$
Surface Pro 4	Product of Microsoft	20.00$
T-Shirt	Women T-Shirt XL size	50.00$
TV Samsung MK55	TV Samsung Full HD 55"	100.00$
TV Sony KDL40	TV Sony 40" Full HD	55.00$

Figure 8. Product table when access by link

After the activation is successful, user can login in from the upper right conner button.

In the action row, you can click the item you want to add to the cart. After adding some item to the cart , the button “Go to cart” will appear.



Welcome to guest!
Change password

Logout

Product List

Select category : all

Name	Description	Price	Action
Adidas	Woman shoe	25.00$	Add to cart
iPad 2	iPad 2nd generator of Apple	15.00$	Add to cart
Nike	Man shoe	30.00$	Add to cart
Product 5	Some random description	54.00$	Add to cart
Product 6	Some random description	34.00$	Add to cart
Surface Pro 4	Product of Microsoft	20.00$	Add to cart
T-Shirt	Women T-Shirt XL size	50.00$	Add to cart
TV Samsung MK55	TV Samsung Full HD 55"	100.00$	Add to cart
TV Sony KDL40	TV Sony 40" Full HD	55.00$	Add to cart

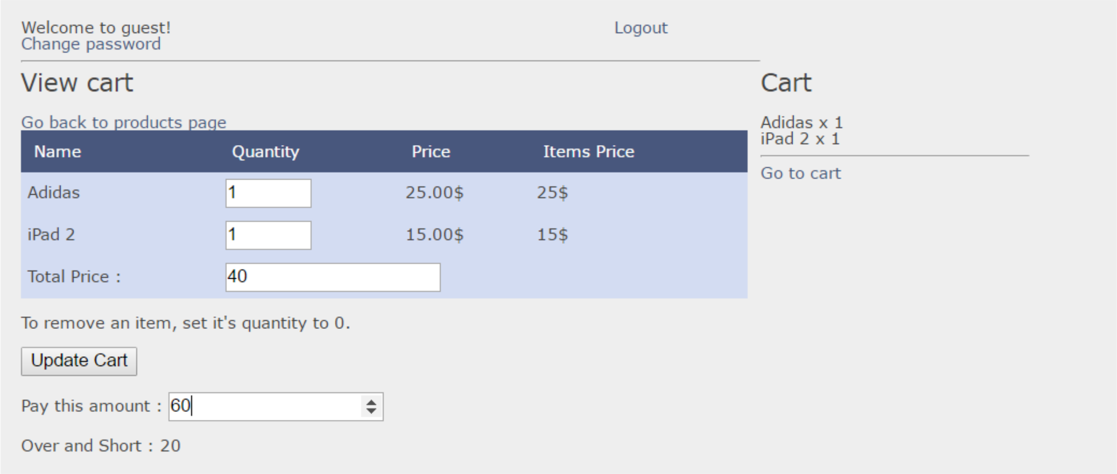
Cart

Adidas x 1
iPad 2 x 1

Go to cart

Figure 9. After login window

After click on “Go to cart”, you will go tho a page View cart , where you can view and modify you shopping cart. You also can add an amount of money to pay. It will inform you that you have pay more or less than it should be at line “Over and Short ” at the bottom of the page.



Welcome to guest!
Change password

Logout

View cart

Go back to products page

Name	Quantity	Price	Items Price
Adidas	1	25.00$	25$
iPad 2	1	15.00$	15$
Total Price :	40		

To remove an item, set it's quantity to 0.

Update Cart

Pay this amount : 60

Over and Short : 20

Cart

Adidas x 1
iPad 2 x 1

Go to cart

Figure 10. Shopping cart page.

5.2 Note

Above is the link to access the web.However, in past, when I was building the website there is a problem happened. I lost all the database in phpMyAdmin and only the template for the website was left. It took me some time to recover that page , as I forgot to export the data base and I have to build it all again like in the beginning. I make some guess one of the reason is that the school might clear the server once in a while. Therefore, to deal with this case, which might happen again, I made a short video clip screen shot and upload it to youtube.

The link is : https://youtu.be/r5kMnYcTvvU

Alternatively, It is possible to contact me via email : hieuv@metropolia.fi If there is anyproblems so that I could fix it.

6 Conclusions

In this project, I need to overcome a lot problems, which I have never been deal with it. The most difficult issue is that I have no background about web or Unix operating system. Therefore, it took me a long time to get used to all the concept and then start to research for it.

That means I have less time to build this project. Therefore, it not really at a completely high level yet (such as better back and front for the web), but I am sure that at least my project has response to mostly all basic requirement. There are also many problems such as cannot run the script if using notepad or losing all database. However, they are minor problems, it is not work to mention so much about it.

In conclusion, this project has helped me to understand more about Unix operating system and comprehending what I have learnt from Unix labs. This project show that you can make some commands and script to communicate with the web database, which you created. You can extract information or even modify it remotely.

Not only I can understand more about the bash script, but also I can learn how to create a basic website, which I have no ideas before. I learnt a lot about a structure of a php file and how to interpret it. It also important to learn how to create database on phpMyAdmin because without it you cannot connect all the files and run it as a website.

References

HTML guide form (2011), Creating a registration form using PHP, http://www.html-form-guide.com/php-form/php-registration-form.html

Psinas, M. (2012), Role Based Access Control in PHP, https://www.sitepoint.com/role-based-access-control-in-php/

Vlad, S. (2009), Build a shopping cart with PHP and MySQL, https://code.tutsplus.com/tutorials/build-a-shopping-cart-with-php-and-mysql--net-5144