5 Chapter 5: Testing and Implementation

CHAPTER 5

TESTING AND IMPLEMENTATION

In this chapter, testing and analysis will be explained, along side with implementation guide of the application. Testing will be done to arrive at analysis data and support the thesis in its believe on educational games. Implementation guide will show readers how to install and run the application.

5.1 System Specification

5.1.1 Macromedia Flash Professional 8.0

Flash player is used to play the application for the implementation. There are specifications listed for flash.

- Very flexible to being exported or anything being imported
- Compatible with server PHP
- Is able to load outsourced images, movie clips, etc
- Is able to be loaded into internet

System requirements for flash are:

- OS Required:

  - Apple MacOS X 10.3, Apple MacOS X 10.4, Microsoft Windows XP, Microsoft Windows 2000

- Min Processor Type:

  - 800 MHz, 600 MHz, 600 MHz

- Peripheral / Interface Devices:

  - CD-ROM, VGA monitor

- System Requirements Details:

  - Microsoft Windows 2000 / XP - Pentium III - RAM 256 MB - HD 710 MB, PowerPC G3 - RAM 256 MB - HD 360 MB, PowerPC G3 - RAM 256 MB - HD 360 MB

5.1.2 PHP

PHP is used as the middle part that connects flash and database. It is most commonly used to load a flash banner onto any website. It can also be used for a case such as this thesis. PHP specifications are:

- is able to connect to flash and mySQL
- is programmable to suit the needs of any flash programs
- connects to a server and access database on the server
- is very easy to understand
- offers flexibility for any behaviour of the program

5.1.3 MySQL

MySQL is an open source database. It supports many platforms to be used on:

- Linux (RedHat, SuSE, Mandrake, Debian)
- Embedded Linux (MontaVista, LynuxWorks BlueCat)

- Unix (Solaris, HP-UX, AIX)
- BSD (Mac OS X, FreeBSD)
- Windows (Windows 2000, Windows NT, Windows XP)
- RTOS (QNX)
- Handheld (Windows CE)

5.1.4 NaviCat for MySQL

NaviCat for MySQL is used in the development of the application for this thesis mainly to monitor MySQL database. The entire database is accessed using NaviCat which then makes database easier to manipulate, create, construct and destroy. Specifications for MySQL are:

- NaviCat provides auto connection to MySQL
- Displays of tables in MySQL is clear and easier to understand
- Tables with primary keys and foreign keys are easily built in NaviCat
- Imports and exports of data in NaviCat is easy and widely varied in formats
- Support for back up and restore for MySQL databases
- Has well designed visual builder/editor

5.2 Operational Procedures

5.2.1 Installation guides

This section will help readers to install the needed softwares to run the application “Tero’s Adventure”. Those who are familiar and are

above the standard computer literacy are advised to skip this sub section and proceed to other sub-section for better understanding of the application testing.

5.2.1.1 Installing Apache

Apache can be downloaded for free since it is an open source software tool. It is easily found in internet. Some needs to be paid for some doesn't. It is best to find one that is suited to the users. In this thesis, Apache 2.0.59 is used. It is the latest Apache release when it is installed into the author of the thesis's PC.

What we will need:

Apache 2.0.59 Win32 Binary (apache_2.0.59-win32-x86-no_ssl.msi)

Locate the Apache installer from files and double click it. The first few screens are Apache's configuration, configuring windows to install Apache. This will only take a second and then we'll be ready to begin.

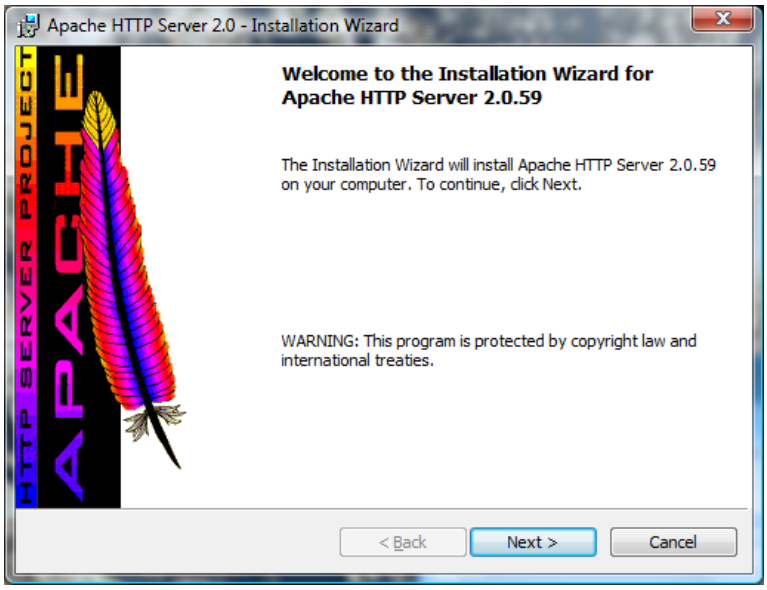


Figure 29 - Welcome page of Apache installation system

Click ‘Next’ button. Next we’ll have a license agreement that we must accept to continue the installation of Apache.

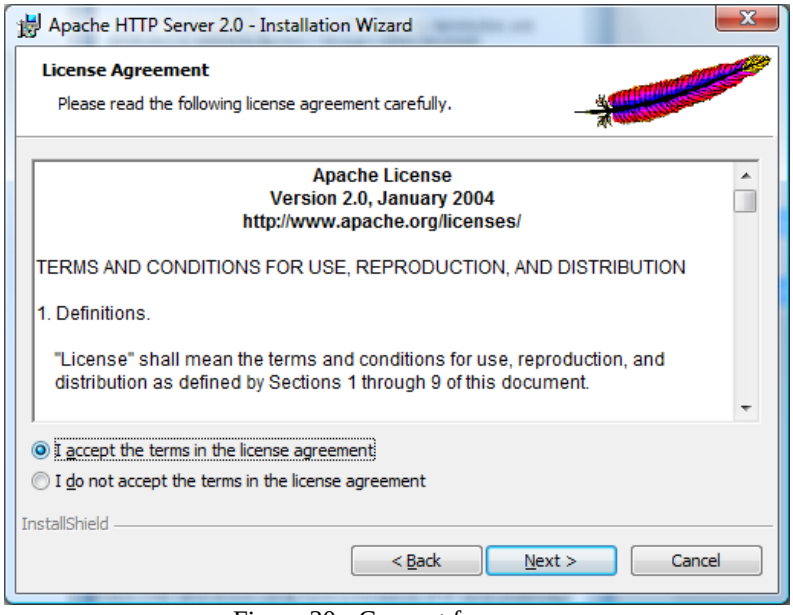


Figure 30 - Consent form

Select “I accept the terms in the license agreement” and click ‘Next’ button.

The next screen we have basically explains what we're installing and some useful information about the application and where we can locate updates and so on. Click the 'Next' button to continue.

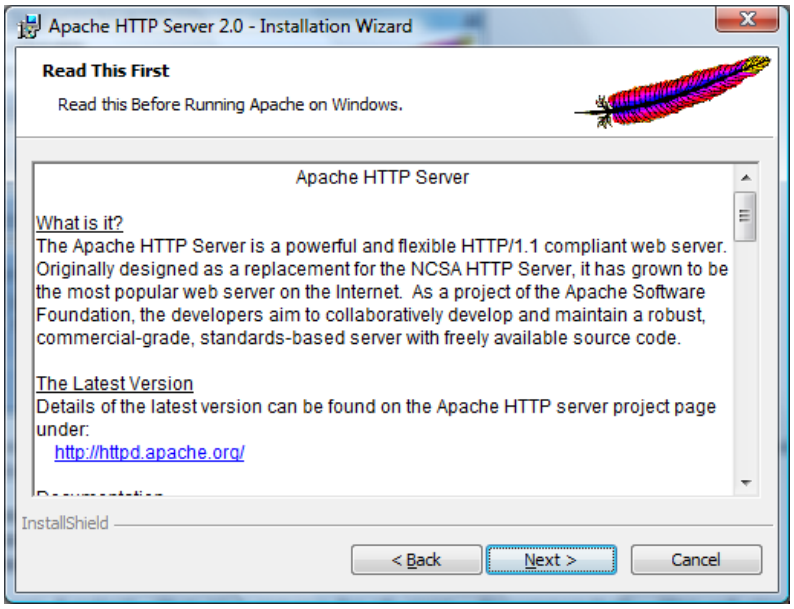


Figure 31 - Apache explanation

Next we have the server information. This is just basic configurations for the server and doesn't take any time at all.

Network Domain (e.g. somenet.com): localhost

Server Name (e.g. www.somenet.com): localhost

Administrator's Email Address: Your email here

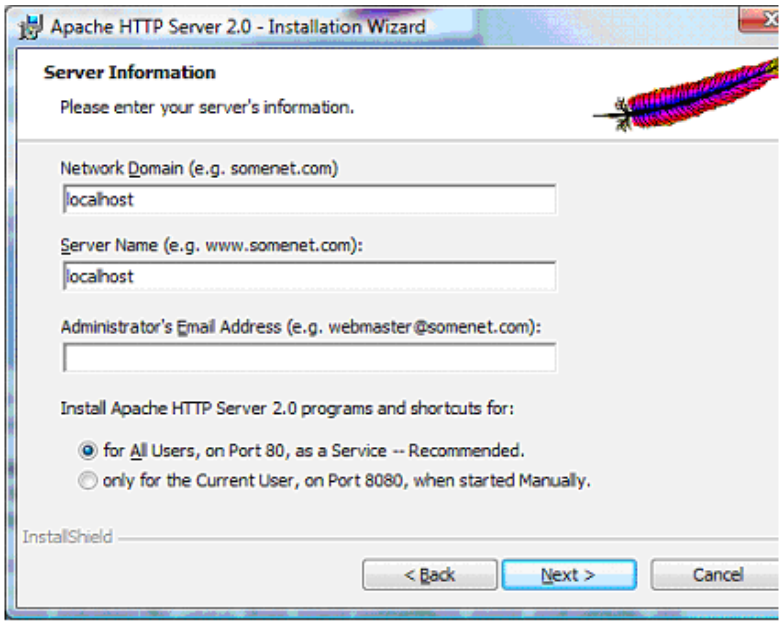


Figure 32 - Server information screen

Then we'll select "Run as a service for all users – Recommended". This will start Apache as a service and run it automatically when the system reboots. This would be the best solution for users that want something quick and simple, no maintenance needed. Click the 'Next' button.

Next we have the selection of how we'd like to install it. Most users would simple use the default selection which is Complete. If you prefer your documents to be in another location or folder, you will have to select Custom for this project and continue. Click the 'Next' button to continue.

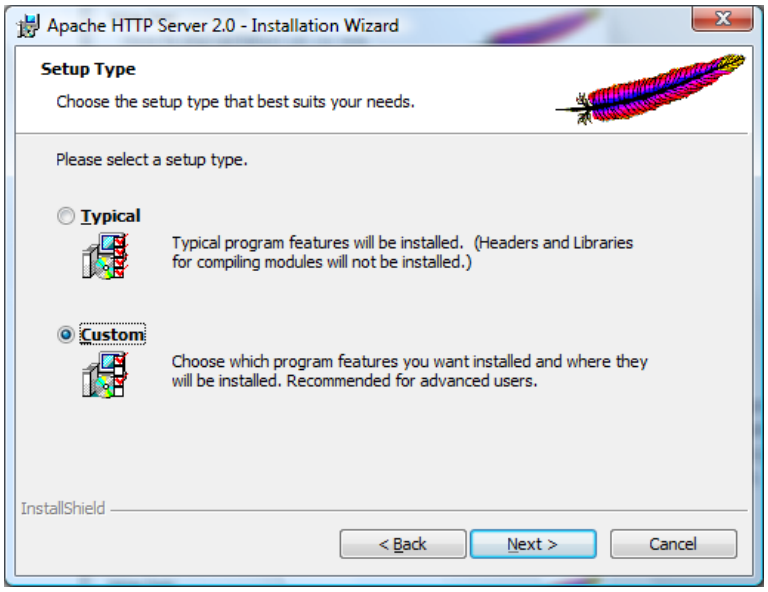


Figure 33 - Setup selection screen

From the next window that appears, select “Apache HTTP Server” and click on the “Change…” button located in the bottom right.

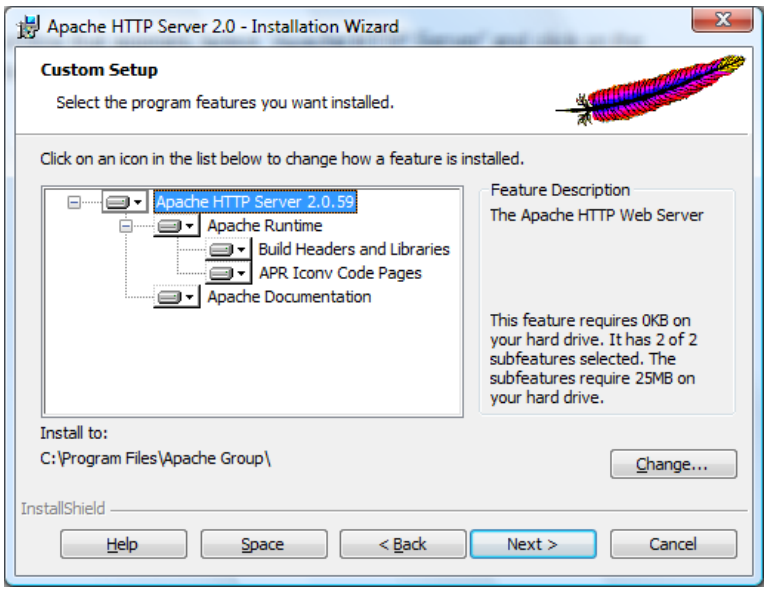


Figure 34 - Custom setup

This will bring up a window similar to this one where we’re able to change the location of the installation folder. You can put all of the server applications to be in “C:\tero adventure” so it will be organized in

one folder. If you want to do this, you need to change the default of "C:\Program Files\Apache Group\" to "C:\tero adventure \". Remember to use an ending back slash. Click 'Ok' and then click the 'Next' button.

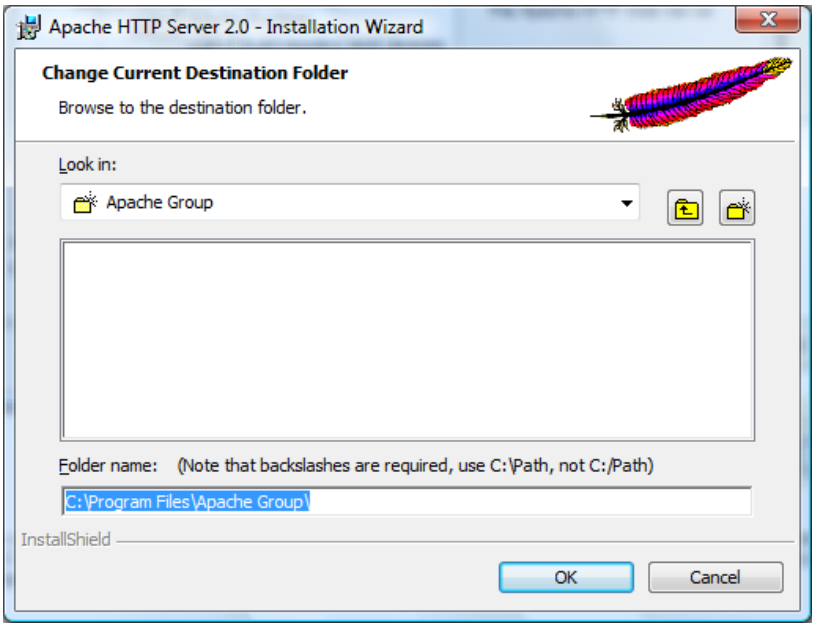


Figure 35 - Changing the default installation folder

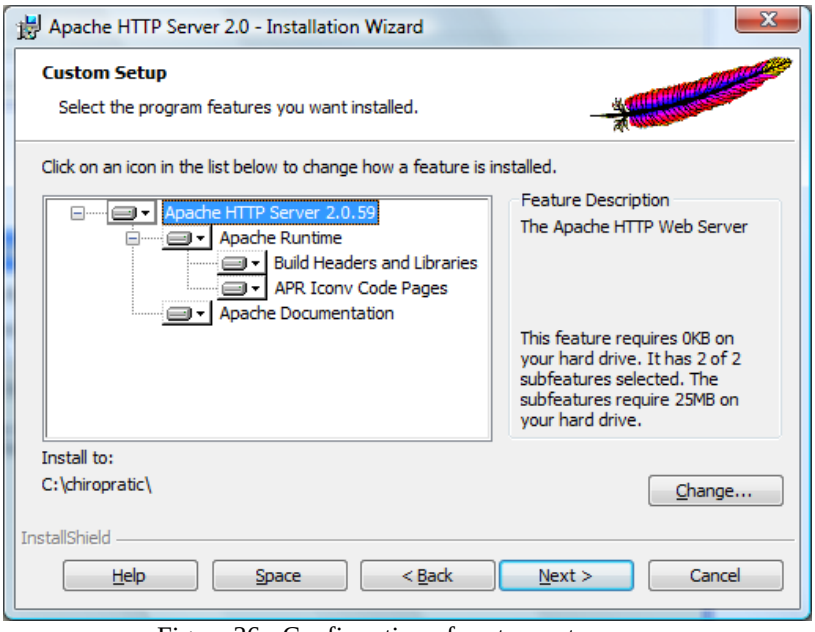


Figure 36 - Confirmation of custom setup

The application will bring us to ready to install screen. Simply click on Install and the installer will begin the installation of Apache using our custom configurations

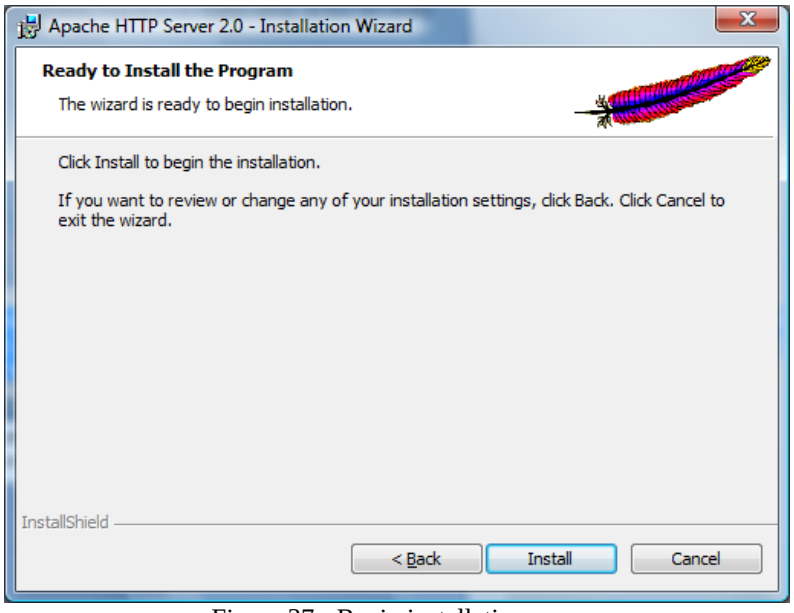


Figure 37 - Begin installation

The Apache is being installed into your computer. If after the installation, there's a popup window (it's a windows firewall warning) that asks whether we want to block or unblock the Apache Port, just choose ‘Unblock’ option.

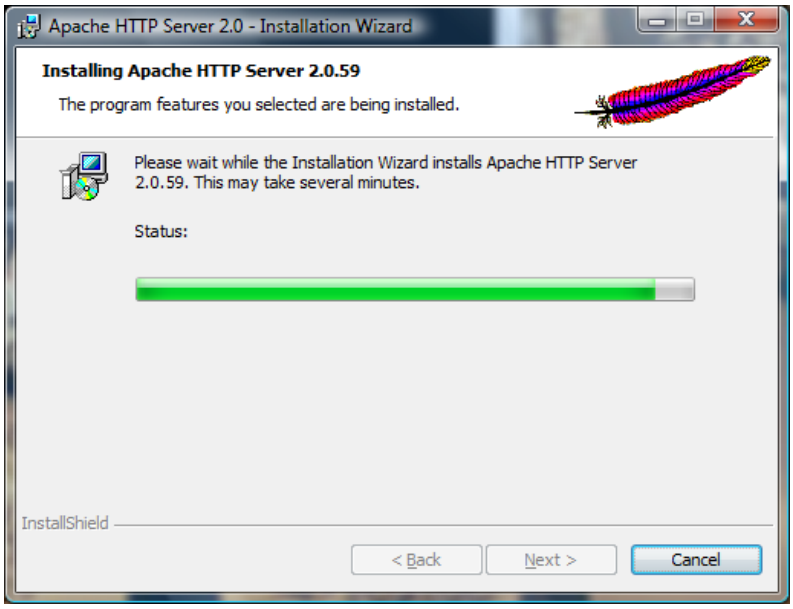


Figure 38 - Installation is in progress

Click the 'Finish' button and we're done. Apache is now installed in the location we provided and configured using the settings we specified during installation.

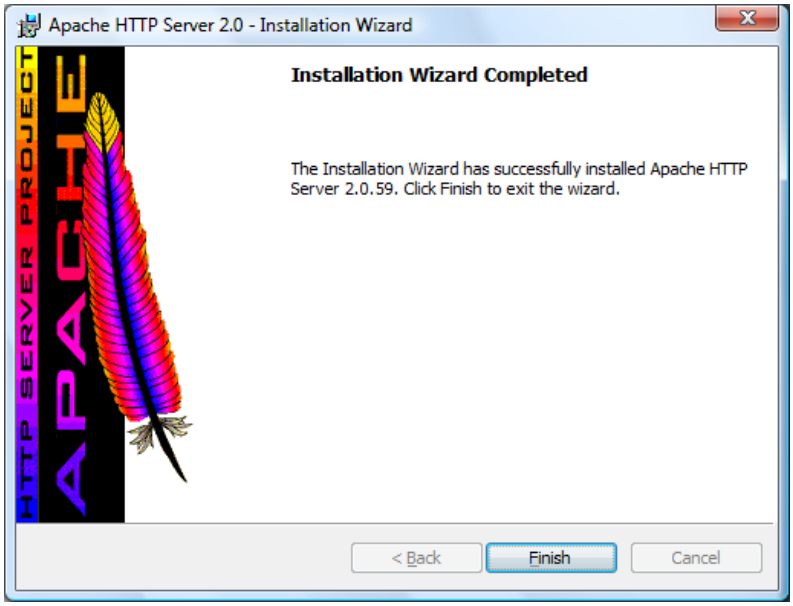


Figure 39 - Finish installation

To test whether the Apache is installed and run correctly, open your internet browser and type in http://localhost/ as the URL.

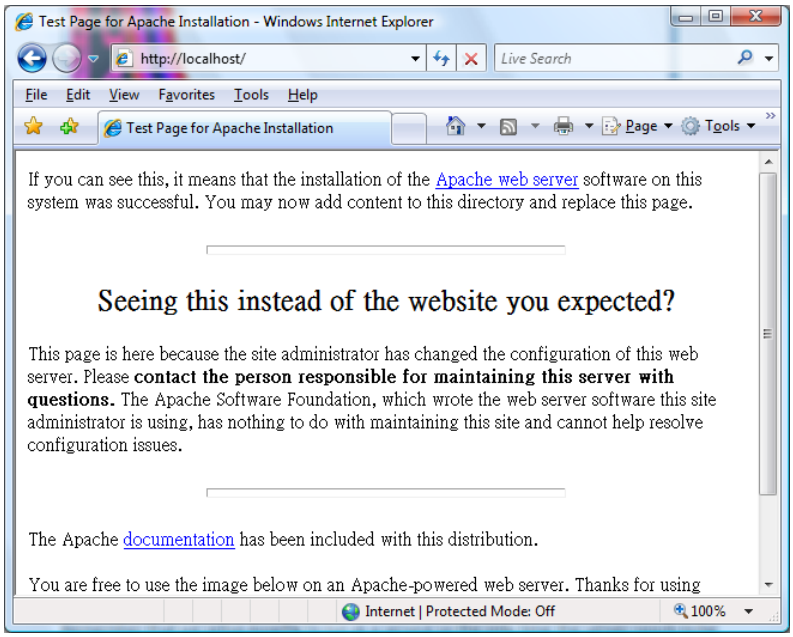


Figure 40 - Testing the Apache program

If you get a page similar to the figure 40 above, that means everything went fine and Apache is now installed and working.

5.2.1.1.1 Configuring Apache Setting to Work with PHP

To be able to connect the pieces of programs (MySQL, PHP, and Apache) to work accordingly, the Apache localhost need to be configured first. This part of document will explain the configuration step by step.

- Go to the folder where you installed the Apache program. In this example, it's the C:\tero adventure folder.

- Open the text file named 'httpd.conf.txt'

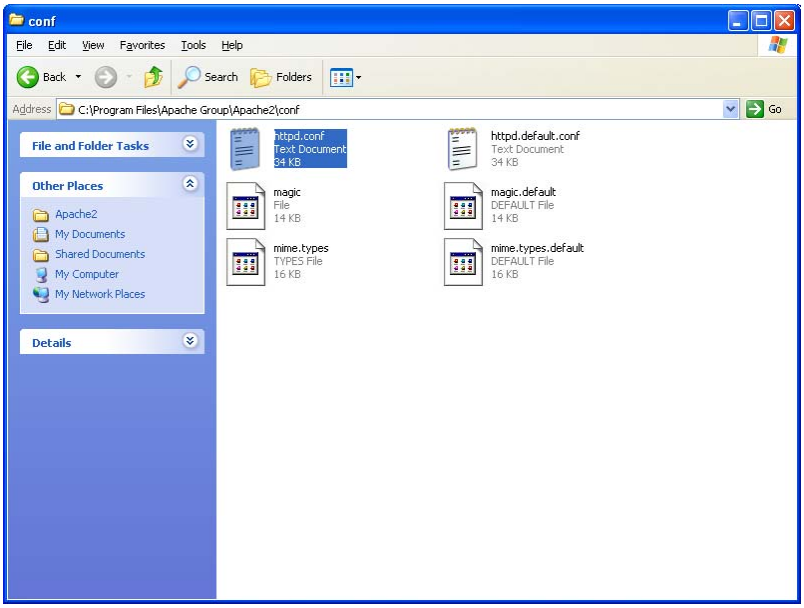


Figure 41 - Location of the 'httpd.conf.txt'

- Open the file

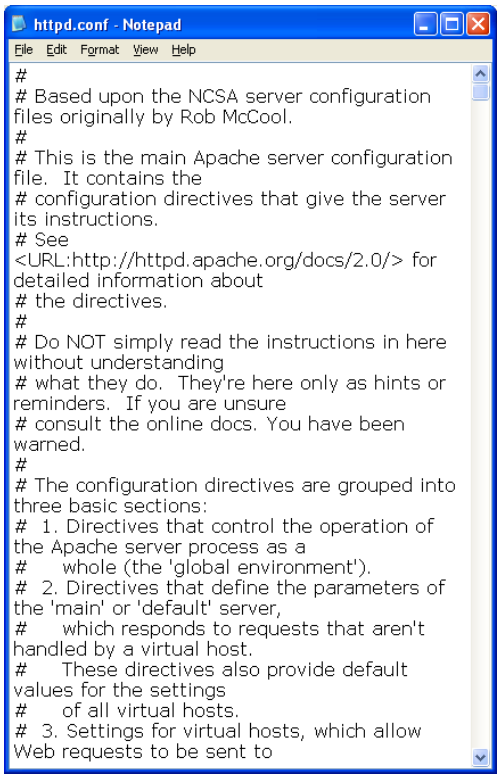


Figure 42 - Content of 'httpd.conf.txt'

- Press 'Ctrl+f' to start searching for a text inside the file. We're going to find 'documentroot' inside the 'httpd.conf.txt' file.

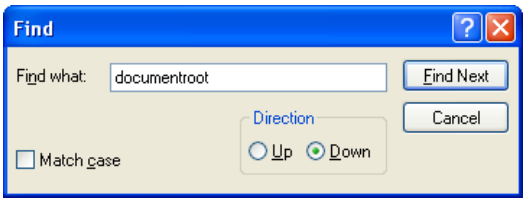


Figure 43 - Searching the ‘documentroot’ word

- You will find the word highlighted.

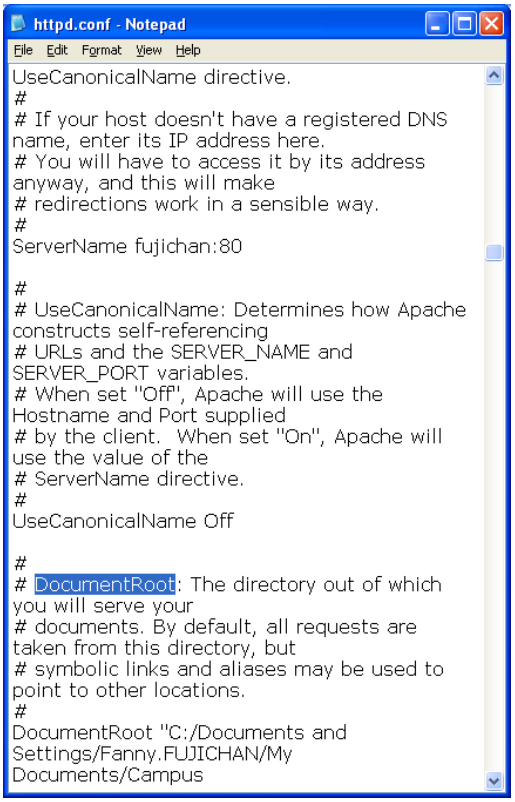


Figure 44 - Finding the ‘documentroot’ word

- Scroll to find the documentroot with a C directory written such as this:

DocumentRoot "C:/Tero/Apache Group/Apache2/htdocs"

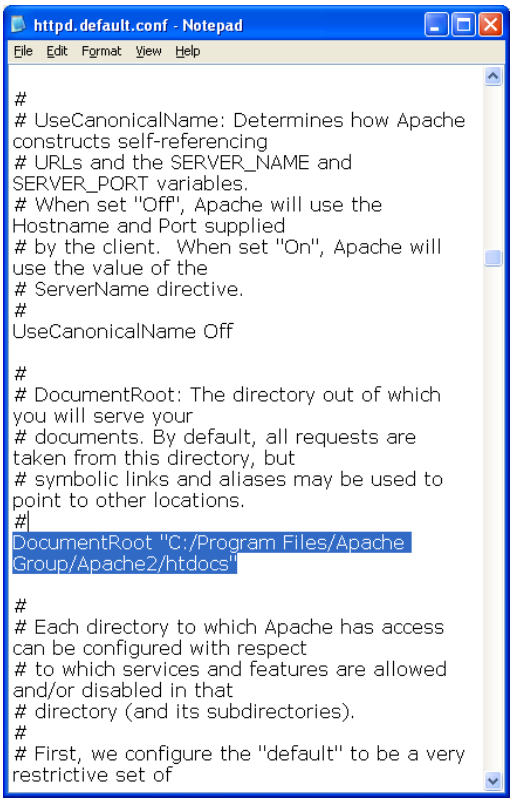


Figure 45 - Finding the path of directory

Replace the directory to the directory in which you will keep all the PHP pages and the program “Tero Adventure 1.7.swf”.

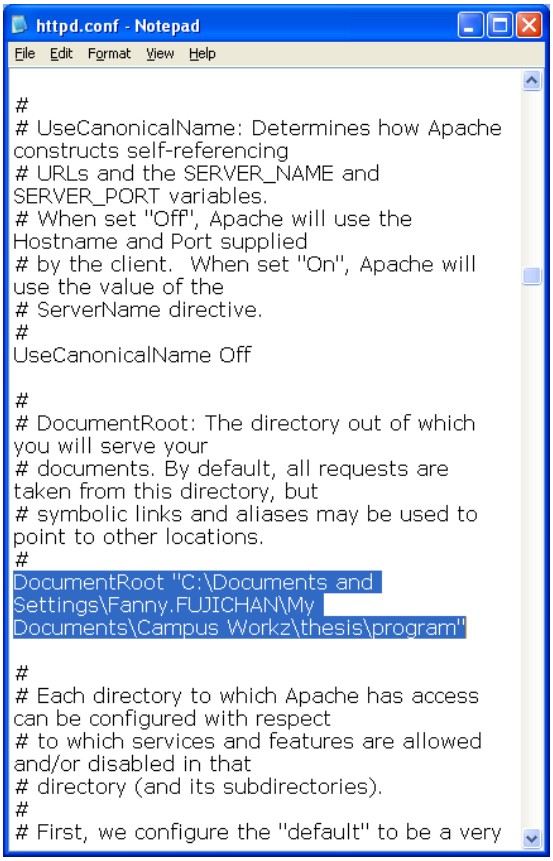


Figure 46 - Changing the location of the directory

- Scroll down to find another lines containing “<directory”. It is not far from the ‘documentroot’ you have edited earlier.

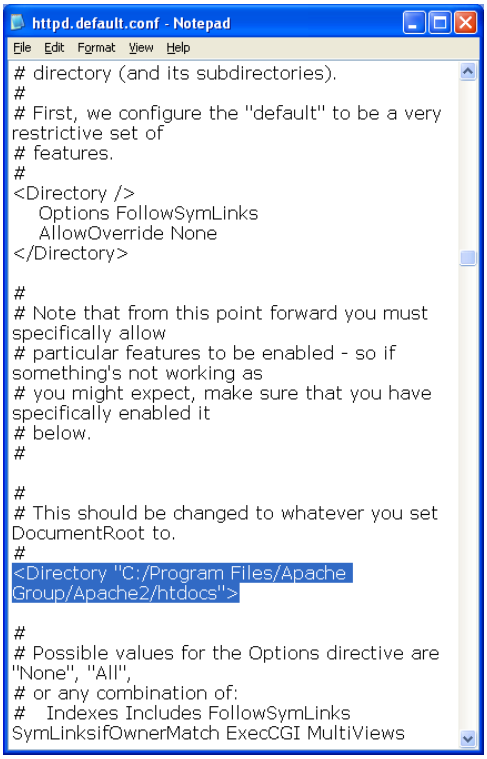


Figure 47 - Find the matching text

- Replace the directory with the directory in the folder which holds the PHP pages and program.

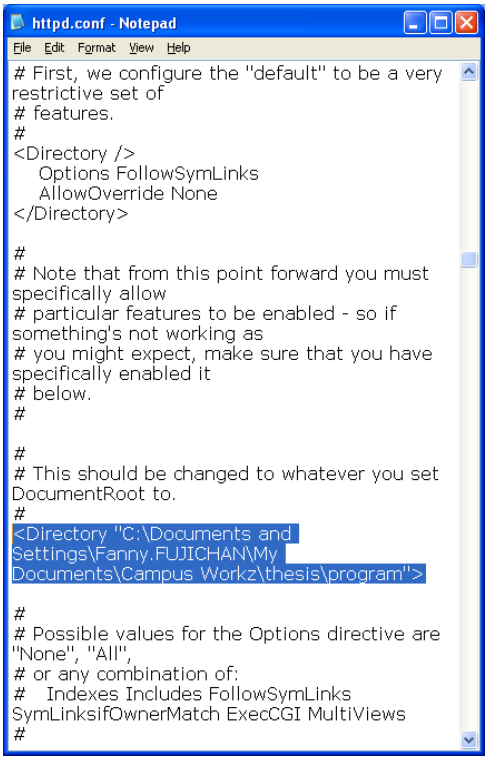


Figure 48 - Change the directory to be the same as the ‘documentroot’ directory

- Go to the bottom of the http.conf to find “#</VirtualHost>” and paste the following lines below the virtualhost

Codes to be placed under #</VirtualHost>
ScriptAlias /php/ "c:/php/"
AddType application/x-httpd-php .php
Action application/x-httpd-php "/php/php-cgi.exe"

Figure 49 - Placing the codes

- Now that you are done, you could test the Apache connection you have by opening internet browser and type http://localhost in the address. The page will show you what is inside the directory of which you store the program and its PHP pages.

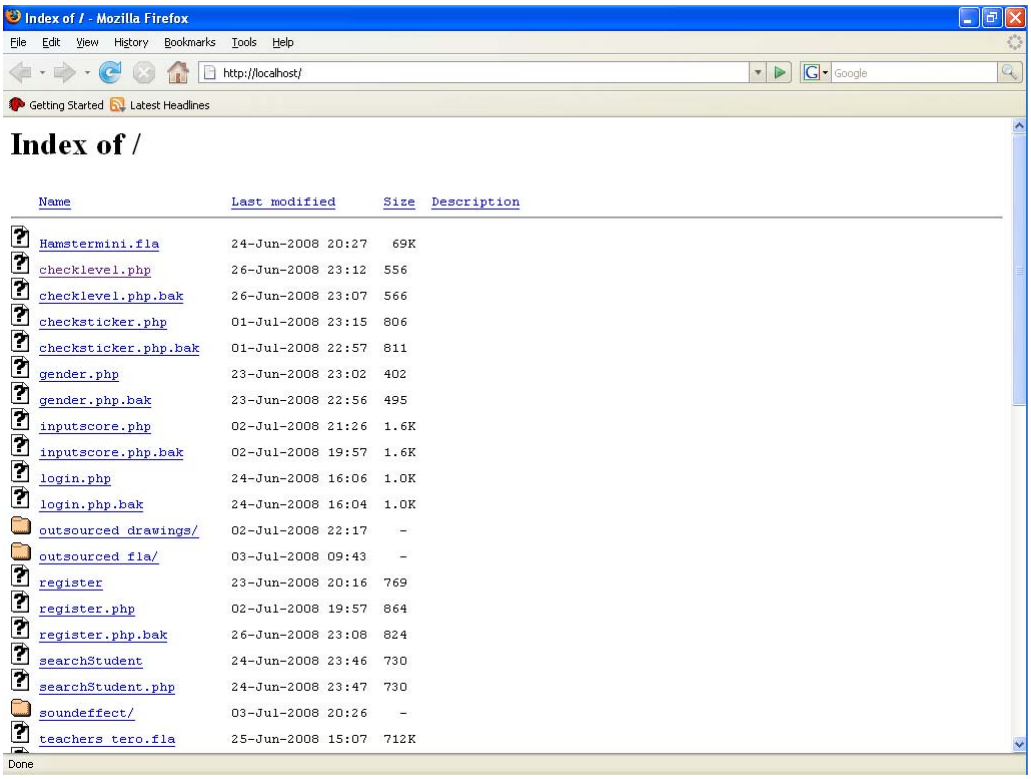


Figure 50 – localhost

Note that the above pictures are examples given to you from another machine that holds different directory in which the program is stored. You need to have your own address directory in which you store the program and its PHP pages.

5.2.1.2 Installing PHP

To help installing PHP easier and faster, I would suggest users get a folder of ready-installed PHP from me. It is far easier than installing PHP on your own. What you need to do is to just copy the PHP folder and put it directly under C:/ directory. You can, however, decide to install PHP from its installer; however, it will not be included in the operational system installation guide in this thesis.

5.2.1.3 Installing MySQL

What you will need:

MySQL 6.0.0 installer (mysql-6.0.0-alpha-win32.zip)

Run an installer called Setup.exe. Double click on that Setup.exe to run the installer.

The figure 2.2.2. below is the first screen that we will see after we double click the Setup.exe. Click the ‘Next’ button to continue the installation.

Figure 51 - Welcome screen of MySQL installation

Next, you are given a choice to choose the setup type. We can install in typical mode, complete mode, or custom mode. For general use, it is recommended to choose Typical. Select "Typical" if it's not automatically selected and click 'Next'.

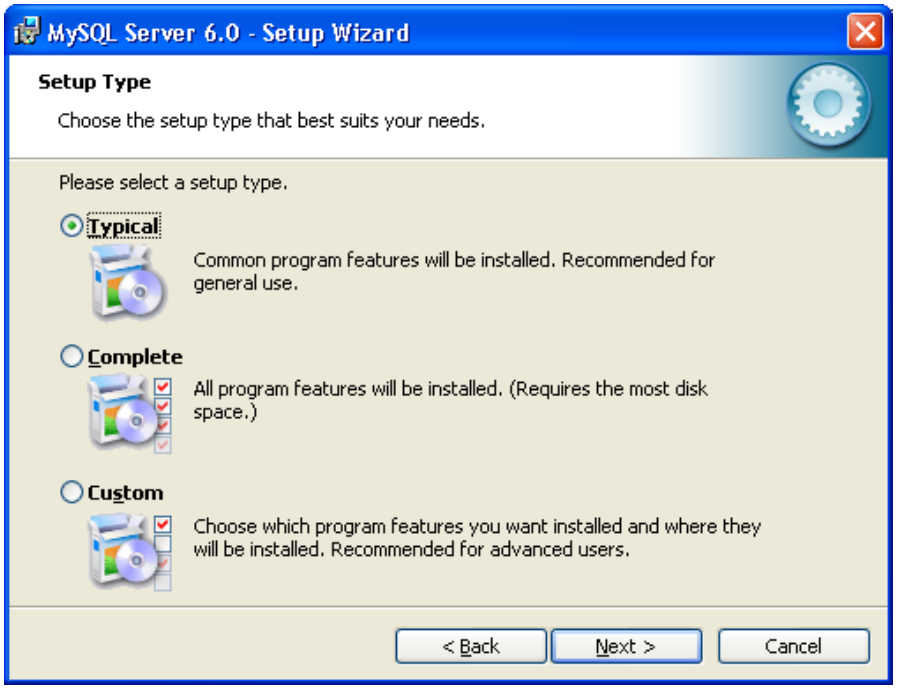


Figure 52 - Setup type selection

The next window will give us a chance to go back to the previous page to change our setup type. Since we are going to install it in typical mode, we can just ignore this page and press Install button directly. Click 'Install' button to install the program.

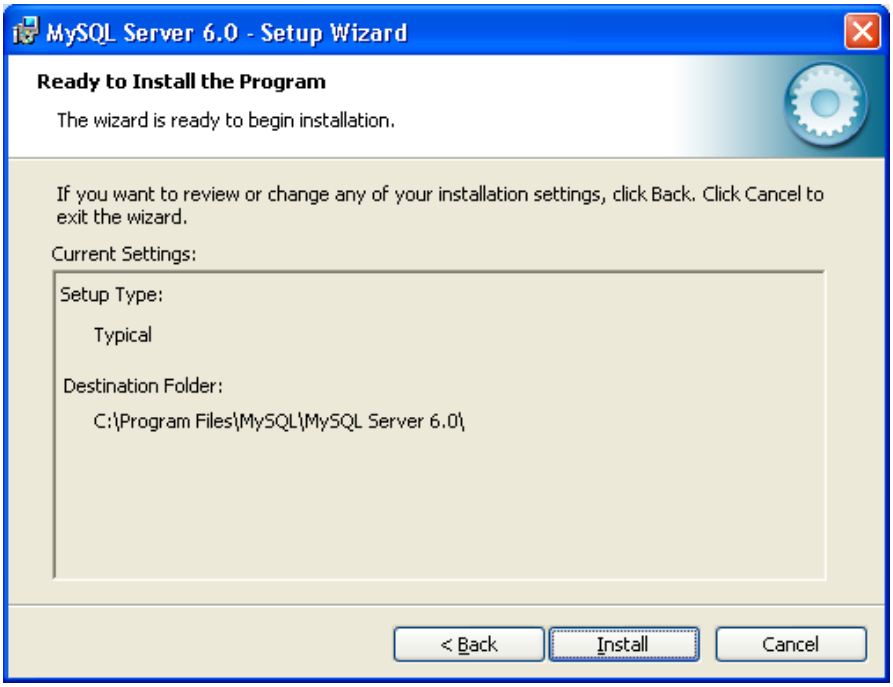


Figure 53 - Starting the MySQL installation

All we need to do in this step is to wait for the installation process to finish. The installation process should not take a long time, but it really depends on your computer speed.

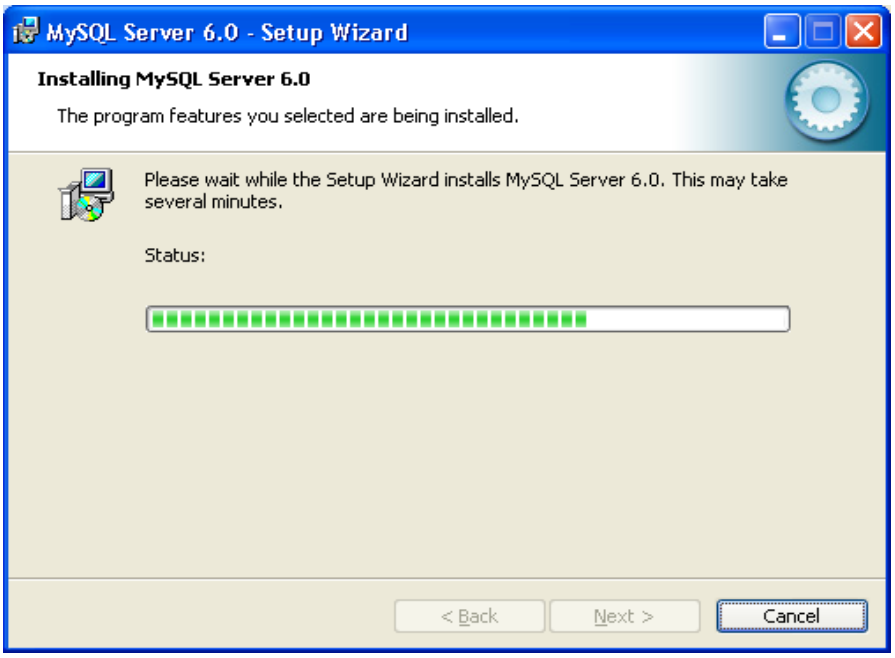


Figure 54 - Installation is in progress

Next, the installation process will describes some information about MySQL itself. To proceed to the next step, simply click 'Next' button.

Figure 55 - MySQL explanation(i)

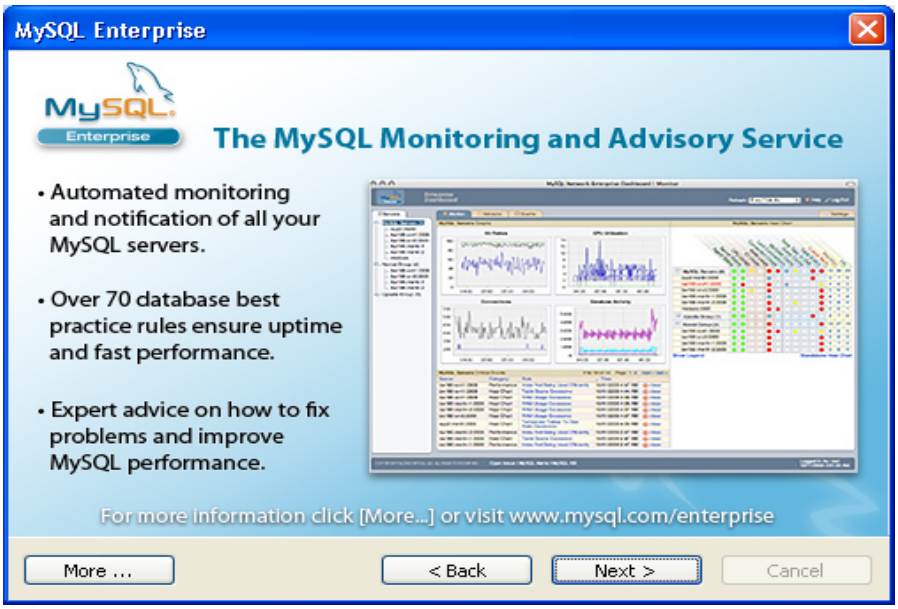


Figure 56 - MySQL explanation (ii)

Now that we have finished the installation process, the next step is to configure the MySQL server. Tick "Configure the MySQL Server now" checkbox if it's not already been ticked, and press 'Finish' button like figure 2.2.8. below.

Figure 57 - Finish installation

Click the 'Next' button to start configuring MySQL.

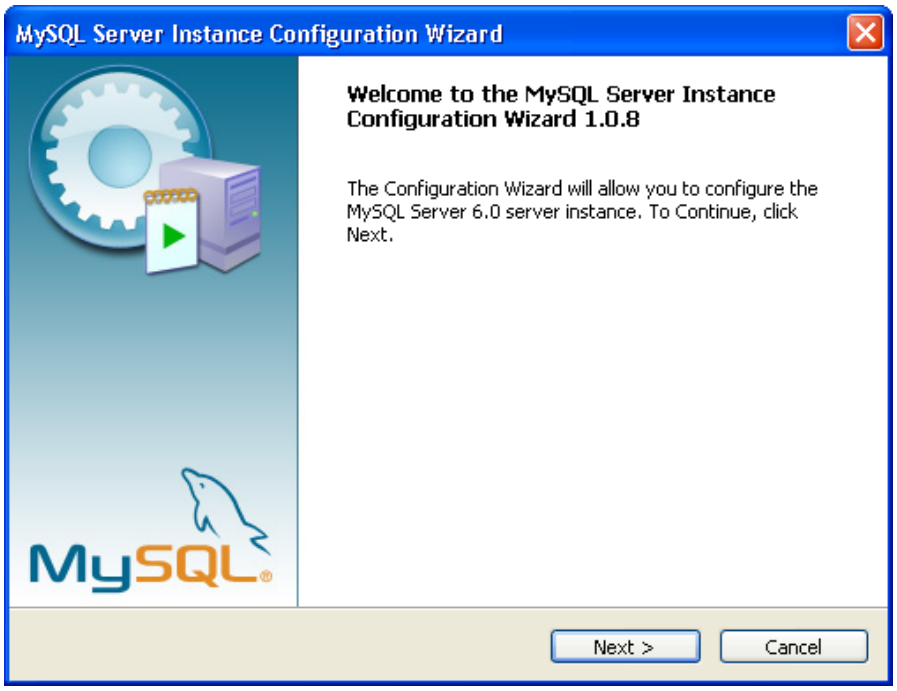


Figure 58 - Starting MySQL configuration

Since we have installed the MySQL Server previously on our machine, choose 'Detailed Configuration', and press 'Next' button. As what has been described on the windows itself, the Standard Configuration is for machine that doesn't have MySQL Server installed in it. Thus, since we already have MySQL Server installed, choose Detailed Configuration.

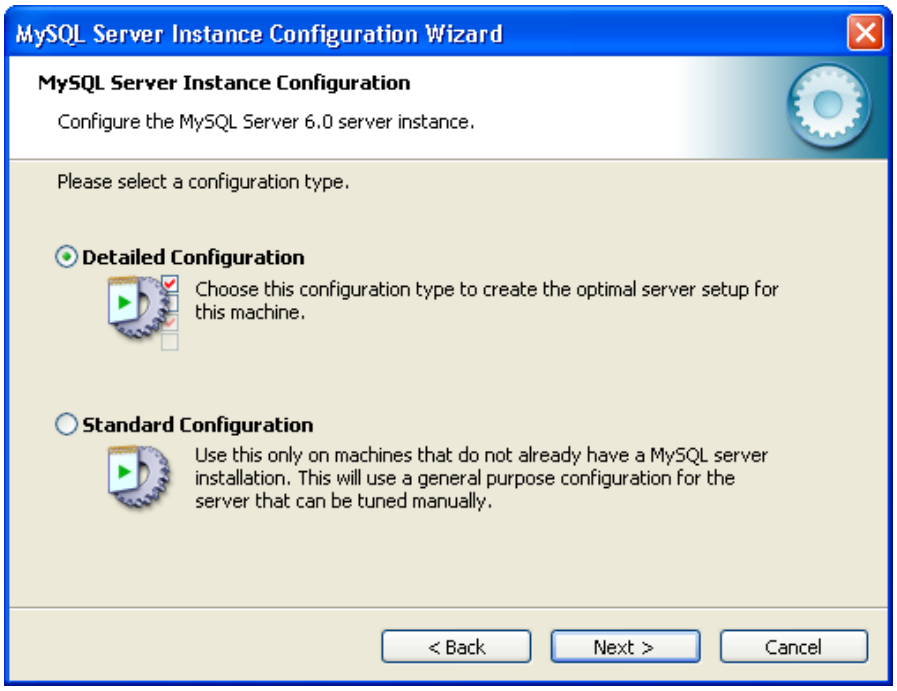


Figure 59 - Selecting configuration type

In the next screen, we are given a choice to choose our server type. Developer machine requires the minimum amount of memory of the computer, thus, we recommend choosing this since we don't really need MySQL Server for handling large application like web or application servers. Besides, we want to reduce the amount of storage needed. Click the 'Next' button to continue.

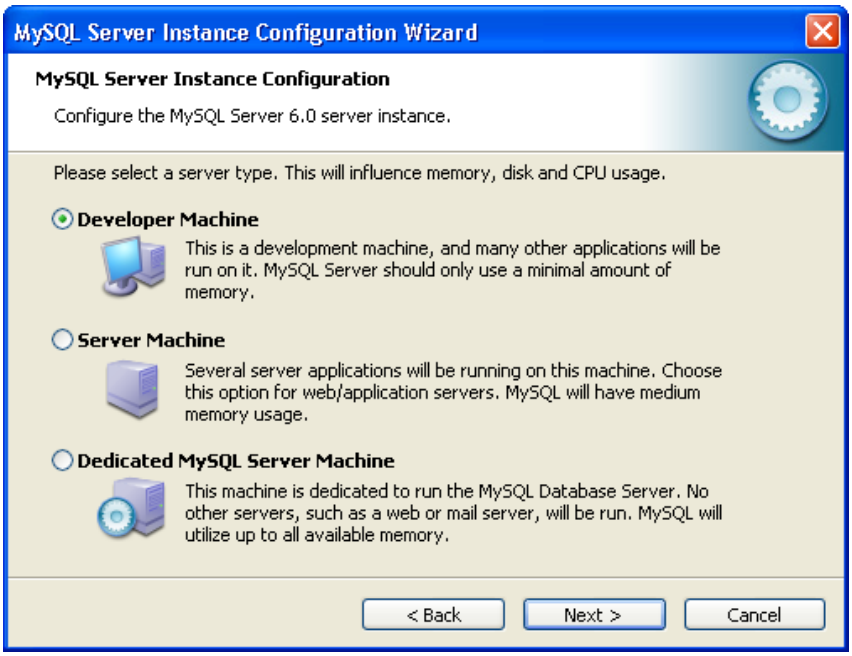


Figure 60 - Selecting the server type

Next, we need to select the type of database we are going to create on the server. Since we are only using the database to store data, it is recommended to choose “Multifunctional Database”. The other two options can be considered if we are going to make larger application database. To continue to the next step, select the ‘Next’ button.

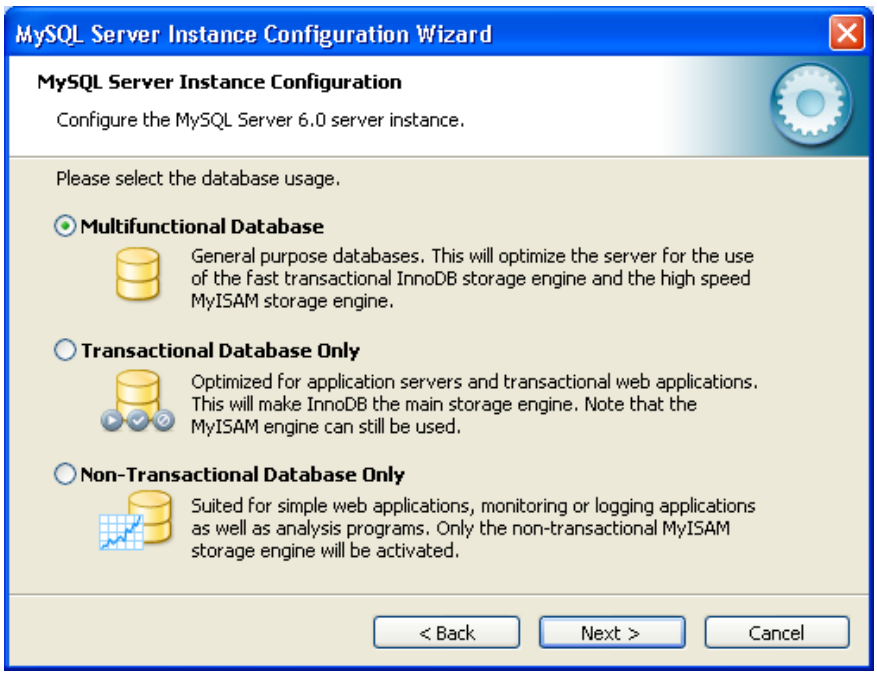


Figure 61 - Selecting database type

Figure 62 below shows the step where we have to set the path for the database. Press next directly without defining the path will give you default setting, which means that the database will be stored to some default folder. For general user, it will be problematic if you cannot find the default folder for the database. Thus, it is recommended to define your own path, so you can keep track of all the data.

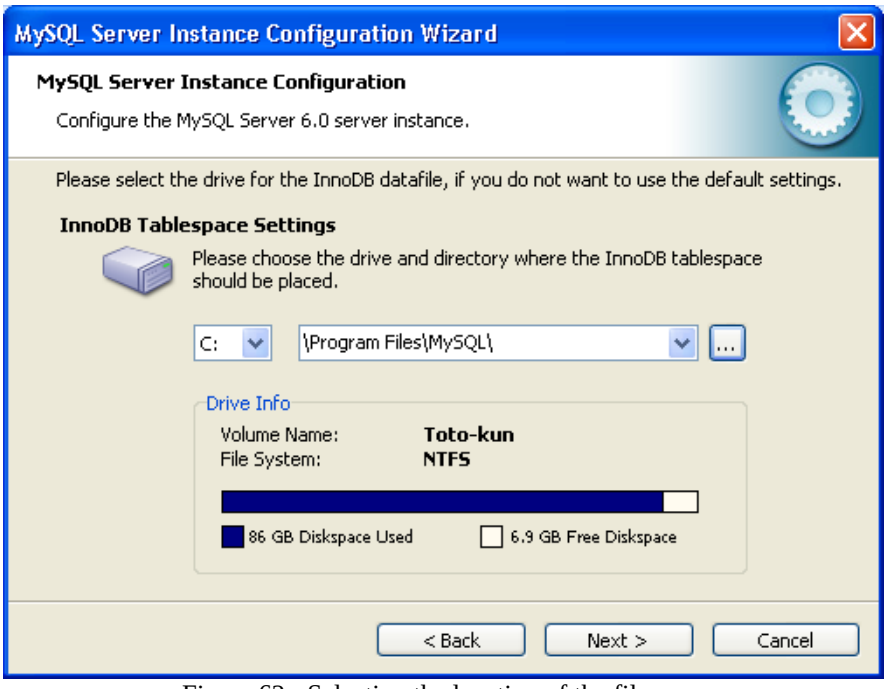


Figure 62 - Selecting the location of the file

The next step is to define the connections to the server. For Tero's Adventure, we recommend choosing Decision Support (DSS) / OLAP because the system doesn't need a lot of connection with the database. Choose 'Next' to continue.

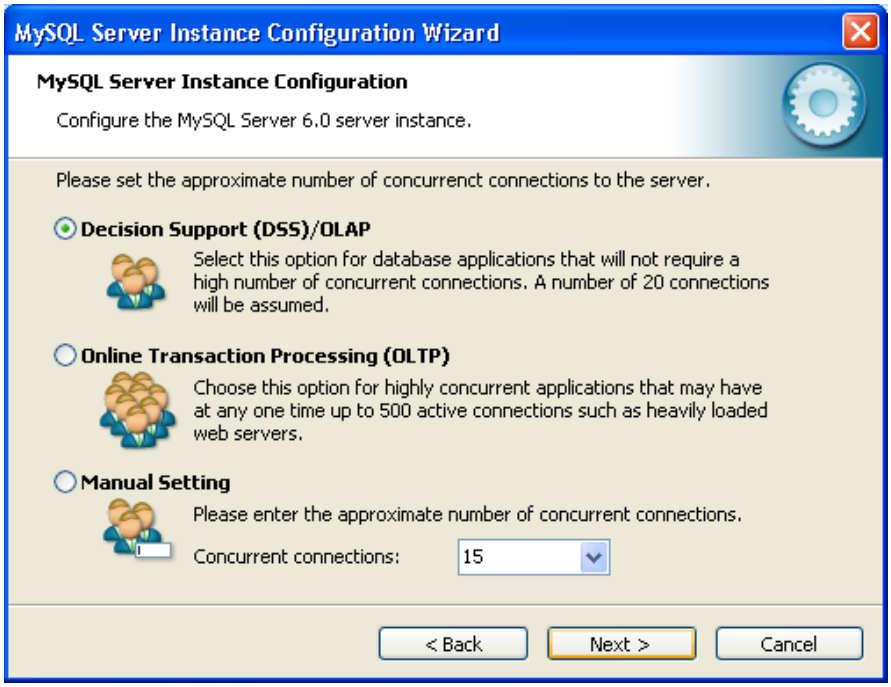


Figure 63 - Selecting the concurrent connections

Click ‘Next’ button to continue.

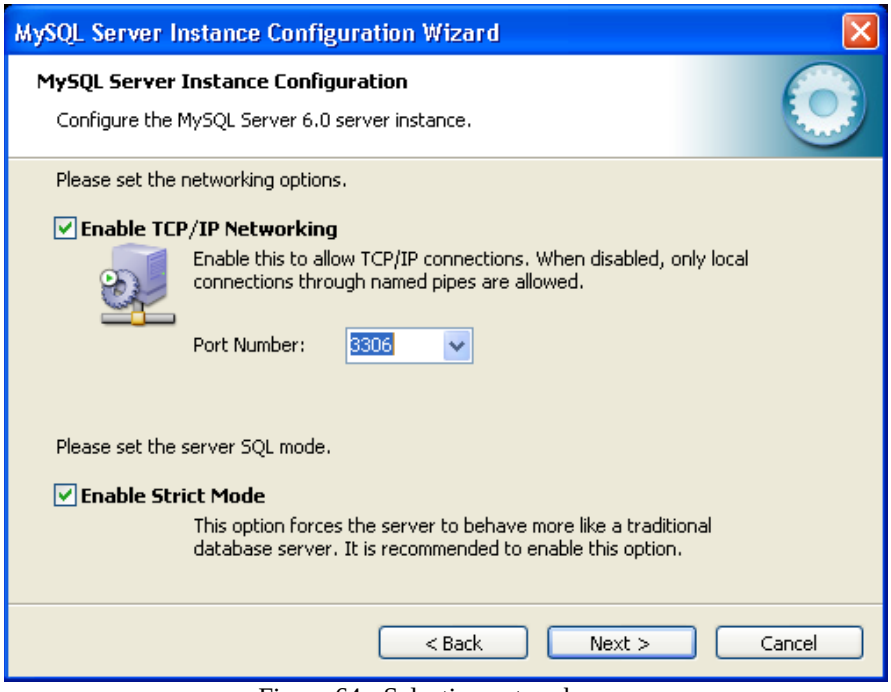


Figure 64 - Selecting network

In the next step, we need to configure the character set. Choose Standard Character set (default) and press ‘Next’ button.

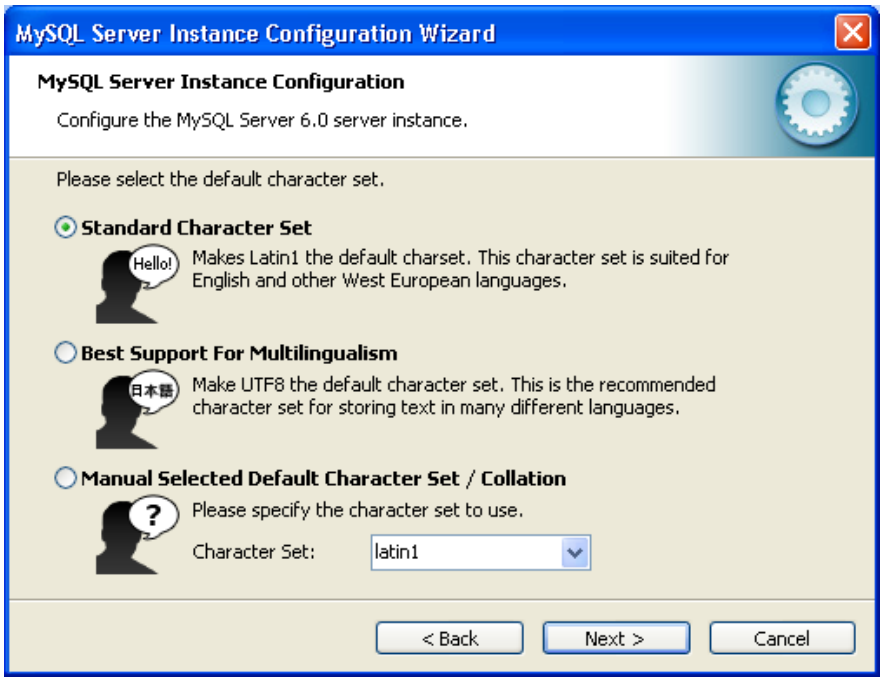


Figure 65 - Selecting character set

Leave the next setting as it is, press 'Next' and proceed to the next configuration just like the figure 66 below.

Figure 66 - Selecting option for Windows

Turn off modify security settings to make things easier by not selecting the option. The password that is in the security settings below have an impact as to the PHP pages that is developed to correspond to the game. I would suggest having the password: “root”. Press ‘Next’ button to continue.

Figure 67 - Selecting security option

After that, press the ‘Execute’ button to start executing the configuration.

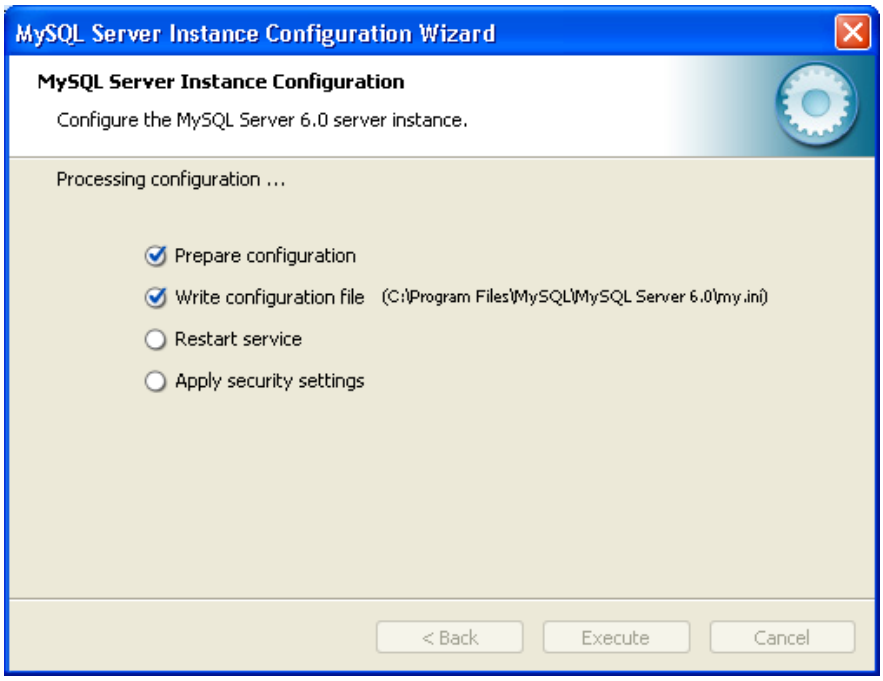


Figure 68 - Executing configuration

Click the 'Finish' button and you're done installing MySQL.

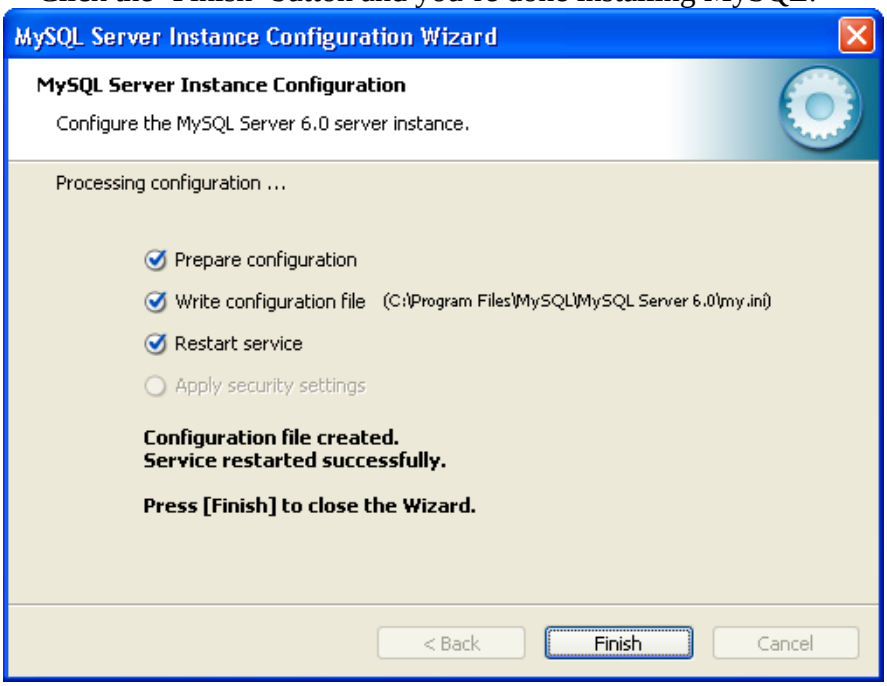


Figure 69 - Finish MySQL installation

Now that all softwares are installed and configured accordingly, application is ready to be used. Should any further problem exist, it is better to contact the author of the thesis for help.

5.3 Test Plan

In this sub chapter, more of the application testing would be described, pointed out and explained. The testing of this application would focus more on functionality testing and later on connection testing.

5.3.1 Functionality Testing

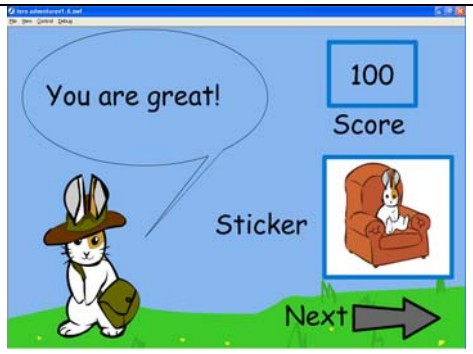
Functionality testing is done to check whether each of game function is working properly. User Acceptance Test (UAT) is used as testing method and also screenshot of the user interface. Functionality testing for the thesis will include: user register/ log in, scoring input system and displaying scores.

Aside from UAT, a special testing is used as well since it is provided in the flash action script for functionality testing. It is called “trace” method; where on a special identified frame, an action script code of *trace (“message”);* is used and this will result in a pop up window generated by flash to alert the programmer as to whether the application is running according to the script. The trace will be included into the UAT as screen shots.

	User Register/Log in
No.	1
Test Case	Players register to be new students or logging in
Precondition	- Student has run the application - All connection configuration of Apache, PHP and NaviCat is set
Test Steps	1. Player select ‘New’ button / ‘Get in’ button 2. Player input name 3. Player press ‘enter’ button
Expected Results	Application connects to database and register the user, thus creating a new field in student’s table in the database. When recording data to database runs smoothly, PHP sends OK sign and application gets player to his/her player’s next frame. For logging in, PHP will send query to the database requesting a unique field with the username and when given only one unique filed from the query result, PHP sends OK sign to the application.
Status	OK
Screenshot	
Additional Info	It is easy to tell if the functionality is running smoothly because the

	database will always show the programmer as to whether or not the data is successfully recorded or not.

Table 23 - UAT: User register/log in

	Scoring Input System
No.	2
Test Case	Players plays the game and application input score into database
Precondition	- Student has run the application and is done playing any one stage - All connection configuration of Apache, PHP and NaviCat is set
Test Steps	1. Player plays any stage 2. Application calculates scores correctly 3. Stage play ends and goes to menu
Expected Results	Application connects to database and input score. If inputting score into database runs smoothly, PHP will not get any error from its query and so with no termination, PHP can send OK sign to the application, indicating a smooth input into the database.
Status	OK
Screenshot	

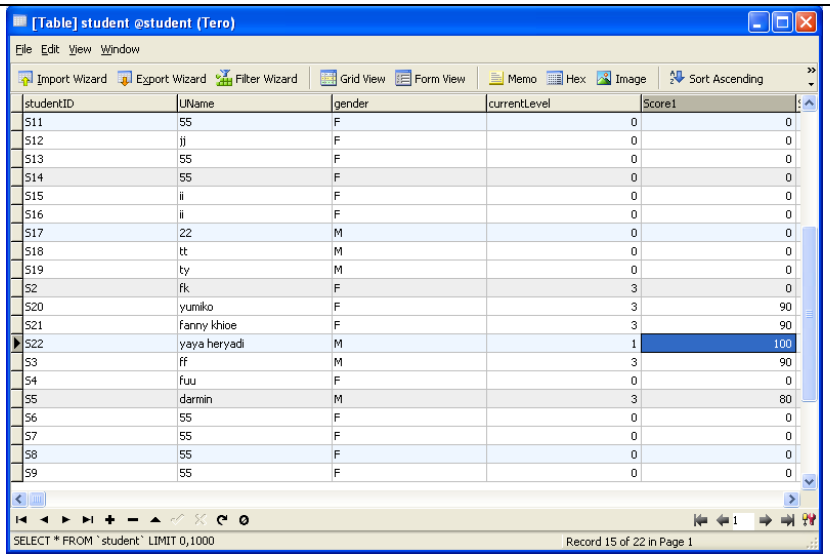
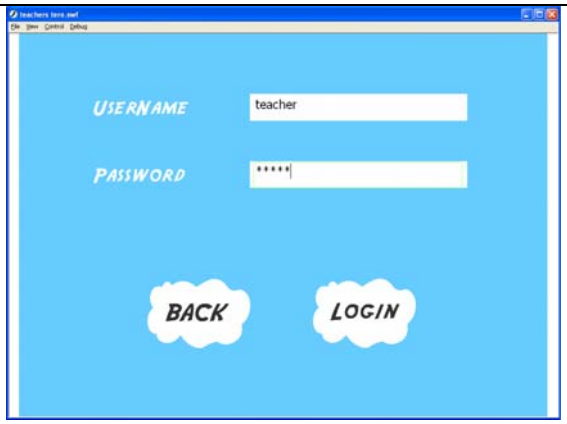
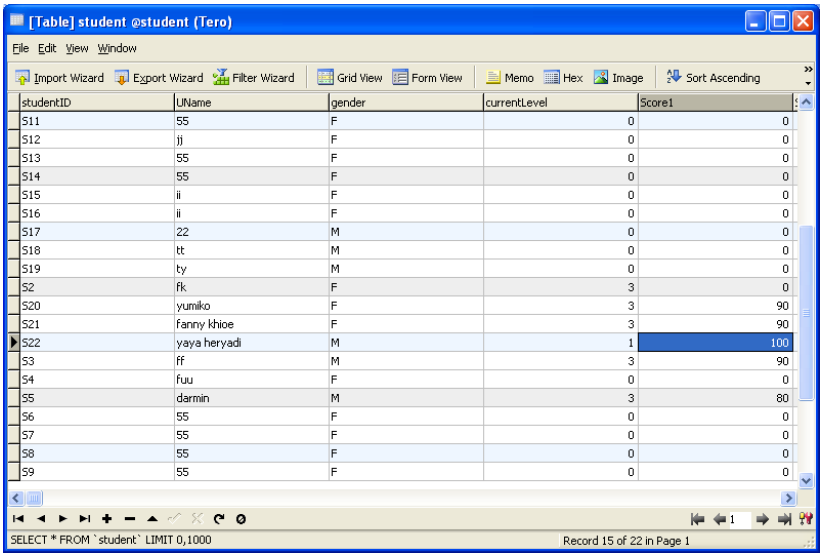
	
Additional Info	After 'Next' button in the particular page displayed above is pressed, it is then the application loads PHP to send query to the database for input.

Table 24 – UAT: Scoring Input system

	Displaying Scores
No.	3
Test Case	Teachers open teachers' site to view scores by students so far
Precondition	- Teacher has run the correct teachers' application - Teacher inputs his/her username and password correctly - All connection configuration of Apache, PHP and NaviCat is set
Test Steps	1. Teacher inputs username and password correctly 2. Teacher selects "view scores" button
Expected Results	Application connects to database and generate an array of all students and their scores. The array is then passed onto the application from PHP and values will be displayed in a table data.

	Table data allows resizing, ordering data ascending or descending and scrolling of the table. The data displayed should be identical to all data inside the database.
Status	OK
Screenshot	 

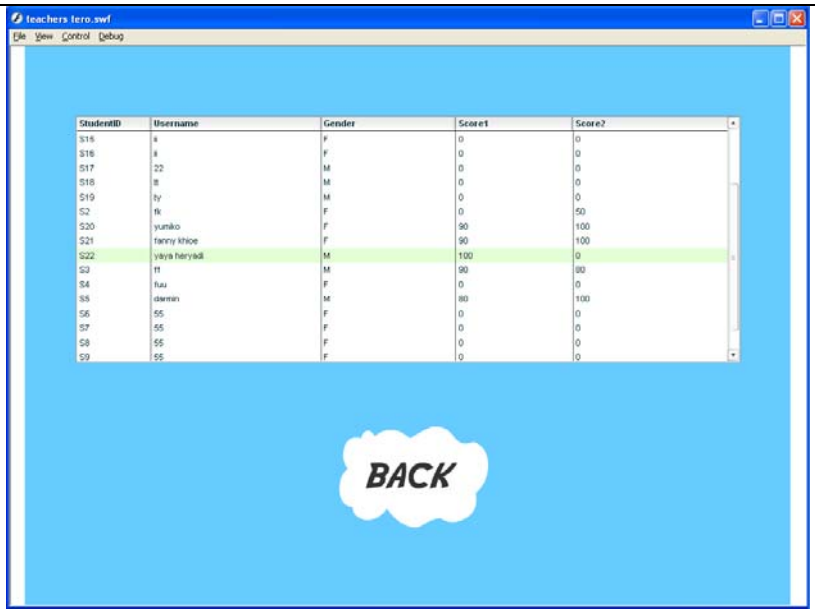
	
Additional Info	Notice that the data displayed is identical to the data in the database.

Table 25 – UAT: Displaying Scores

5.3.2 Connection Testing

Connection Testing is used to test the connection of several softwares and tools used to put the application to work. The application requires quite a number of connection or bridges between several software and tools such as PHP, Apache and MySQL. Thus, connection testing is essential and it is actually done prior to further development of the rest of the application. Once connection testings give good and positive result showing each tools and software are connected correctly and accordingly, the development of the program/application could then begin. The connection testing will test the connection of flash and Apache, Apache and PHP and Apache and MySQL. The testing would be

done by all kinds of methods that are simple yet sufficient enough to show a connection exist.

5.3.2.1 PHP and Apache Connection Testing

To check the connection between PHP and Apache, proper installation and configuration is needed. Once both PHP and Apache are installed and configured correctly and accordingly, testing is pretty easy. First, a php page is created to print out any sentence to be shown on the Internet Explorer page, indicating a connection is established.

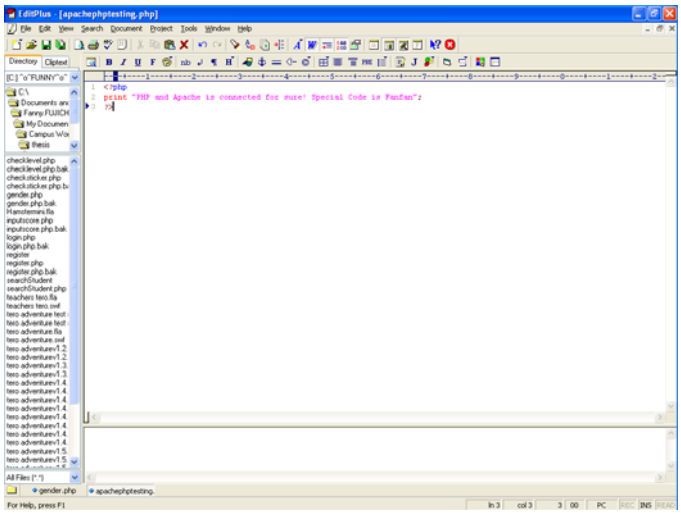


Figure 70 - PHP page trial

```
<?php
print "PHP and Apache is connected for sure! Special Code is Fanfan";
?>
```

For the testing, the above code is used. When the page is done, it is saved under the same directory of the application so Apache can connect to the php page. To test the connection,

Internet Explorer or the same kind of internet page tool can be used. Type http://localhost/(filename).php. In the testing, the filename is testingapachephp.php. So the URL would be http://localhost/testingapachephp.php. The result would be like the figure below.

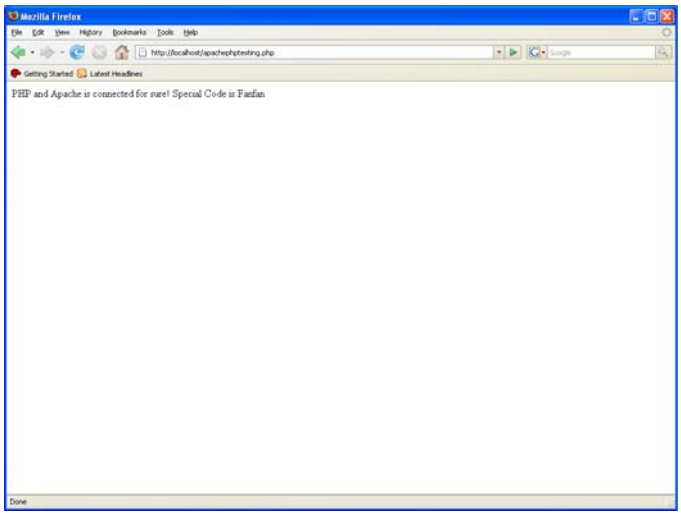


Figure 71 – PHP Testing Result

5.3.2.2 Apache and MySQL connection testing

Apache would connect to MySQL only if MySQL configuration is correct and PHP pages that calls database of MySQL is in the specified folder that interacts with Apache. Since we have covered PHP and Apache testing, it is almost the same to test Apache and MySQL because it is done via php pages. Instead of putting just print codes in the php page, we should connect the php to the database by coding in the php page syntaxes for database connection.

```
<?php
$db_username = "root";
```

```
$db_password = "root";
$db_name = "student";

$conn = mysql_connect("localhost", $db_username, $db_password);
mysql_select_db($db_name);
?>
```

To test the result of the database connection, a query to fetch some students' data is coded into the php page. The codes are as follow:

```
$myquery = "SELECT studentID, UName, gender, Score1, Score2 FROM student";
$result = mysql_query($myquery) or die('Query failed: ' . mysql_error());

print "&masuk=1";
$arraysize = mysql_num_rows($result);
print "&arraysize=$arraysize";
$mydeli = "~";
print "&mydata=";
//$count = 0;
while($row = mysql_fetch_array($result))
 {
        $myID = $row['studentID'];
        $myuser = $row['UName'];
        $gender = $row['gender'];
        $score1 = $row['Score1'];
        $score2 = $row['Score2'];
        $mydat=$myID.$mydeli.$myuser.$mydeli.$gender.$mydeli.$score1.$mydeli.$score2.$mydeli;
        print "$mydat";
 }
```

The above codes would display the students' data when the php page is called by a method similar to the Apache and PHP testing. The result would then be like the figure below.

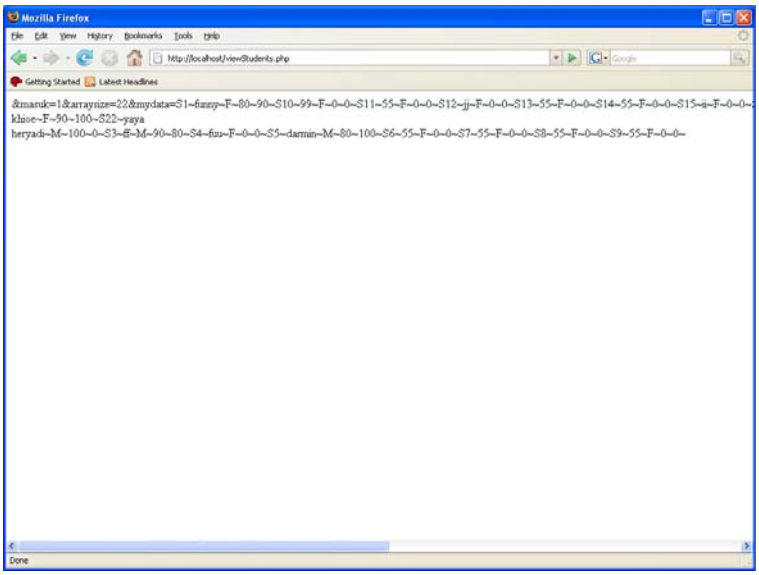


Figure 72 – MySQL Connection Test Result

Open up NaviCat to check whether the students' data are identical to those displayed in figure 72. The identical data is the sign that Apache has successfully connected to MySQL database via php pages. From now then, php pages can be developed to build the application.

5.3.2.3 Apache and Flash connection testing

Once Apache, PHP and MySQL have all successfully connected to each other accordingly, Flash program can now be developed and connected to Apache by calling the URL of the php pages needed. Flash testing requires action script such as below:

```
loadVariables("http://localhost/login.php", this, "POST");
this.onEnterFrame = function()
{
        if(_root.logIn.teachlogin == 2)
        {
                gotoAndStop("MainMenu");
        }
```

```
        if(_root.logIn.teachlogin == 1)
        {
                gotoAndStop("wrongPass");
        }
}
```

In login.php, the codes are below:

```
//teacher
$teacquery = "SELECT teacherID FROM teacher WHERE
UName='$user' AND passwords='$pass'";
$teacresult = mysql_query($teacquery) or die('Query failed: ' .
mysql_error());
$teacnumber = mysql_num_rows($teacresult);
if($teacnumber == 1)
{
        print "&teachlogin=2";
}
else
{
        print "&teachlogin=1";
}
```

Therefore, when a teacher with a username and password is found, teachlogin will be equals to 2 and in Flash, if the output from php page is detected, indicating a connection, the mainmenu frame would be called. When the testing results in what is expected, and then the connection of Flash and Apache is established.