Shi Yang

Session control using PHP and MySQL

Bachelor's Thesis

Information Technology

May 2011

MIKKELIN AMMATTIKORKEAKOULU

Mikkeli University of Applied Sciences

DESCRIPTION

MIKKELIN AMMATTIKORKEAKOULU Mikkeli University of Applied Sciences	**Date of the bachelor's thesis** 16 May 2011
Author(s) Shi Yang	**Degree programme and option** Information Technology

Name of the bachelor's thesis

Session control using PHP and MySQL

Abstract

My project is to build an online shopping website. Recently, it has become a popular trend for an increasing amount of people to choose online shopping. It is very convenient and suitable for people's fast-paced lifestyle nowadays. The main function I want to implement is shopping cart.

Session control is the core technology to support me achieving shopping cart. The idea of it is to be able to track a user during a single session on a website. HTTP protocol has no built-in way of maintaining state between two transactions. (Welling & Thomson 2009, 509). So session control is essential component for me to complete the project.

In this thesis, I will explain session control and involved programming languages in detail. Besides, I will also expound my design ideas to the project, and some considerable codes in the whole program.

Subject headings, (keywords)

Session Control, Online Shopping Website

Pages 56 pages	**Language** English	**URN**

Remarks, notes on appendices

Tutor Timo Mynttinen	**Employer of the bachelor's thesis** Mikkeli Univerisity of Applied Science

ACKNOWLEDGEMENT

I would like to express my great appreciation to all those people who gave me a lot of help in completing the final thesis. Without their help, I could not have it completed in a relatively short period of time.

First of all, I want to thank my parents. When I encounter difficulties, when I feel tired, they always give me the greatest support. And my university, the Beijing University of Technology, provides the exchange opportunity to me, so that I can study in Finland.

Then, I would like to thank my tutor Timo Mynttinen, the language teacher Elina Kiviranta, and my mental tutor Matti Koivisto; Also I am appreciative to all the teachers here in Mikkeli University of Applied Sciences who gave me precious knowledge and help.

Finally, I want to thank all my friends who helped me with professional document resources providing, especially those ones who brought great interests to me in web-programming field. All the suggestions on some significant parts of the project led me to this final success, are essential and considerable.

CONTENTS

1 INTRODUCTION

With the development of science and technology, as well as the improvement of life standards, internet has become an indispensable part in people's life. Many people use internet to facilitate their study and work in early years. Students can easily search information to help their studying and get more knowledge. Businessmen can easily communicate with their remote work partners to promote cooperation. In recent few years, there was increasing number of people starting to care about aspects of life though the internet.

As the variety of shopping sites has emerged, "online shopping" became a very popular idea especially for young people. All of a sudden, the people who are interested in online shopping set off a wave of fashion. They can get what they need through the internet, such as clothes, food and even electrical equipment. Nowadays, people can buy almost anything that occurs in the market through the internet. There are lots of advantages, online shopping is very convenient to the buyer and the price is usually lower than it in market. To the society, it can increase the employment opportunities. From marketing view point, technical support of achieving online shopping is how to build up a shopping site.

My project is to build a website like that; the actual problem is that the HTTP protocol has no built-in way of maintaining state between two transactions. It means when a user requests one page, followed by another, HTTP does not provide a way for you to tell that both requests came from the same user. (Welling & Thomson 2009, 509). So I need session control to help me solve the problem. It can help to identify the user and track the user's behavior. Besides, it can also help to implement shopping carts (Welling & Thomson 2009, 509). Session control is nowadays a very wide range of applications.

It will involve a great deal of knowledge during the process of doing the project. To complete the project, I need to understand the basic format and syntax in Web production, like judgment and loop. Additionally, I must be proficient in PHP language, understand MySQL; know how to connect PHP and MySQL. The software NetBeans IDE 6.9.1 and WampServer will be used. The aim of the project is to consolidate the knowledge has been learned, assimilate more fresh knowledge in. It is focusing on improving the ability of using theoretical knowledge into the practice flexibly, so that the project will be more consummate.

From the specific terms, I will build a website in selling cosmetics. Cosmetics and skin care products are the things women need very much. They are all high prices and high-margin products; therefore I think the website will be useful and particularly liked by women. In my project, I will mainly complete several parts including products added to catalog for selection, users' registration and authentication, achieve temporary shopping cart using session control.

The structure of my final thesis is divided into several main chapters. In Chapter 2, I will introduce the related programming languages, containing the history and some basic concepts. In Chapter 3, I will have a brief introduction of the programming environment. In Chapter 4, I am going to first elaborate session control, and then declare some codes of my project. It will involve the initial creation of database, the codes about HTML forms and interactivity, some significant codes about PHP and MySQL, and part of codes about beautifying the Web page. In Chapter 5, I will give the conclusion and some other ideas about the project that I have not done.

2   INTRODUCE PROGRAMMING LANGUAGES

My project will involve HTML, CSS, PHP, and SQL language. I will introduce them respectively.

2.1 HTML

2.1.1 The History of HTML

HTML is one member of a family of markup language called SGML, which stands for Standard Generalized Markup Language. SGML was developed by the International Organization for Standards in 1986 to define markup language designed for various different purposes. (Mack & Platt 1997, 29).

HTML and the Web were first conceived in 1989 by a researcher named Tim Berners-Lee who worked for CERN, the European Laboratory for Particle Physics in Geneva, Switzerland. CERN researchers developed the first World Wide Web programs in 1990 and started to release programs and specifications to Internet users in 1991. In December 1994, the World Wide Web Consortium (W3C), which is now responsible for the standards of HTTP, HTML, and other Web technologies, was regrouped. And vendors such as Microsoft, Netscape, SGI, and Sun work with W3C to develop HTML. (Mack & Platt 1997, 28).

For the version history, the first version of HTML was just called HTML. It was an excellent early attempt, and the foundation for subsequent versions. A successor to HTML was called HTML+ in 1993. (Mack & Platt 1997, 30). In 1995, HTML 2.0 was published as the first standard and popular version of HTML. It brought some added capabilities such as form-based file upload and tables. The next standard version was HTML 3.2, which was published in January 1997 as the first version developed and standardized exclusively by the W3C. It reconciled overlap among

various proprietary extensions and adopted most of Netscape's visual markup tags. (HTML Wikipedia 2011). Between HTML 2.0 and 3.2, there was an episode called HTML 3.0. Because of too big a change from previous version, it was replaced by HTML 3.2 completely. (Mack & Platt 1997, 30). In December 1997, HTML 4.0 was published as a W3C Recommendation. Initially code-named "Cougar", HTML 4.0 adopted many browser-specific element types and attributes, but at the same time sought to phase out Netscape's visual markup features by marking them as deprecated in favor of style sheets. In December 1999, HTML 4.01 was published; and as of mid-2008, HTML 4.01 is the most recent versions of HTML. (HTML Wikipedia 2011).

2.1.2 The Definition of HTML

HTML stands for Hypertext Markup Language; the filename extension is .html or .htm. It is the markup language with which World Wide Web hypertext documents are written and is what allows you to create hypertext links, fill-in forms, and etcetera (Graham 1998, xix).

Explain separately, Hypertext means the ordinary text that has been dressed up with extra features, such as formatting, images, multimedia, and links to other documents. (Mack & Platt 1997, 21).

Markup is the process of dealing with ordinary text and adding extra symbols. Each of the symbols used for markup in HTML is a command that tells a browser how to display the text, called elements or tags. (Mack & Platt 1997, 22). Elements or tags are to mark text as headings, paragraphs, lists, quotations, and so on. There also has tags for including images within the documents, for including fill-in forms that accept user input, and, most importantly, for including hypertext links connecting the document being read to other documents or Internet resources such as WAIS databases or anonymous FTP sites. (Graham 1998, xxi). So it means that when you write a HTML

page, you should start with the text of your page and add special tags around words and paragraphs. Those tags indicate the different parts of the page and produce different effects in the browser. (Lemay & Colburn 2006).

Language is actually a key point to remember about HTML. Definitely, HTML is a language; it has its own syntax, slang, and rules for appropriate communication, but it is not really a programming language. Instead, HTML is a markup language in essence; it has a different aim from creating a program. (Mack & Platt 1997, 22).

2.1.3 The Features of HTML

HTML is based on the Standard Generalized Markup Language (SGML), a much larger document-processing system. HTML as a computer language is to build Web pages. To write HTML pages, you need to know one of the main features of SGML is that it describes the general structure of the content inside documents rather than its actual appearance on the page or on screen does help. (Lemay & Colburn 2006).

For design principles, HTML was not designed to be the language of a "What You See Is What You Get" (WYSIWYG) word processor such as Word. Instead, HTML takes a "What You Get Is What You Meant" (WYGIWYM) approach, such that authors must construct documents with sections of page. (Graham 1998, 2).

For the language's limitations of HTML are that HTML is not a word processing tool, a desktop publishing solution, or even a programming language, for that matter. That's because its fundamental purpose is to define the structure and appearance of documents and document families so that they could be transferred quickly and simply to a user over a network for rendering on a variety of display devices. (Musciano & Kennedy 1997). In other words, HTML as a universal language, the only thing it does is classify the function of different sections of a document so that a browser can display it correctly. It means HTML defines different parts of your page, for example,

which part is a headline, which part is a table, which part needs to be emphasized, which part includes an image, and so forth. (Mack & Platt 1997, 22). So I think this is why HTML is the principal part of a Web page.

HTML is a markup language. A markup language is a way of describing, using instructions embedded within a document, what the different parts of the text mean or what they are supposed to look like. There are two basic markup language approaches. The first is known as physical markup. In this approach, the markup tags explicitly say how the document should look. The second approach is known as logical or semantic markup. Here, the markup language defines the meaning of the text but not how it looks. Although semantic formatting is more difficult because you have to think about what each part means when you add the markup instruction, it is much more powerful and flexible for describing text. For HTML, it is very much a semantic markup language, designed to specify the logical organization of a text document; there are very few physical formatting commands in HTML. (Graham 1998, 1 – 2).

Furthermore, HTML is an extensible language. This simply means that new features, commands, and functionality can be added to the language without "breaking" older documents that do not use these new features. There has some important extensions for instance allowing for hypertext links from one document to another, as well as allowing for user input and user interaction. (Graham 1998, 2 – 4).

There are still some left, the important thing to know is that HTML works on a wide variety of platforms (Mack & Platt 1997, 22).

2.1.4 The Structure of HTML

HTML as a structured language, has rules for where element can and cannot go. These rules are present to enforce an overall logical structure upon the document. (Graham 1998, 6).

The structure of HTML is a strict hierarchical nesting of elements structure (Bowers 2007, 37). HTML documents are composed entirely of HTML elements that, in their most general form have three components: a pair of tags , a "start tag" and "end tag"; some attributes within the start tag; and finally, any textual and graphical content between the start and end tags, perhaps including other nested elements. (HTML Wikipedia 2011). Elements can be nested within each other, but they cannot overlap each other. HTML organizes elements into three major categories: structural, block, and inline elements.

The core structural elements are <html>, <head>, and <body> (Figure 1). Information about document goes in <head> and document content goes in <body>. Besides, such as <link>, <meta>, and <title> belong in <head>. (Bowers 2007, 37).
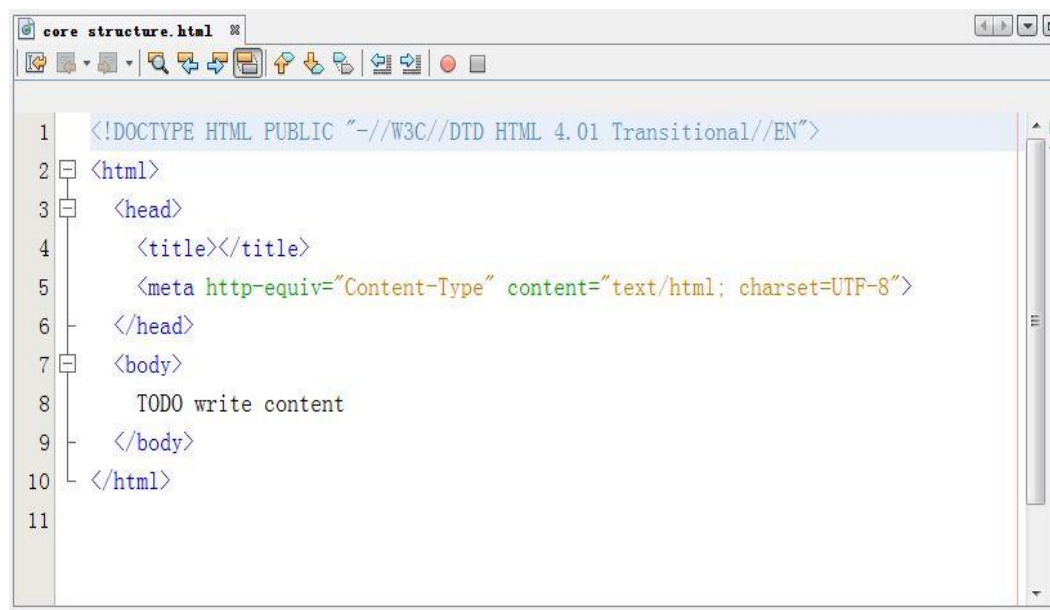

Figure 1. The HTML Core Structure in NetBeans IDE 6.9.1

For block elements, there have three types: structural, such as <ol>, <ul>, <dl>, and <table>; multi-purpose, such as div>, <li>, <dd>, <td>, <th>, <form>, and <noscript>; and terminal, such as <h1>, <p>, <blockquote>, <dt>, <address>, and <caption>. For

inline elements, there also have three major types: semantic, flow, and inline-block. (Bowers 2007, 37 – 51).

In addition, to accomplish my website, the two parts I used most frequently in HTML are Tables and Forms.

2.2 CSS

2.2.1 The History of CSS

Style sheets have existed in one form or another since the beginnings of SGML in the 1970s. Cascading Style Sheets were developed as a means for creating a consistent approach to providing style information for Web documents. (Cascading Style Sheets Wikipedia 2011).

Although style sheets were not a new invention at that time during the beginning of HTML, unfortunately, no standardized method of implementing style sheets was ever outlined, leaving this function up to the various browsers. (York 2005, 13). As HTML grew, it came to encompass a wider variety of stylistic capabilities to meet the demands of Web developers. This evolution gave the designer more control over site appearance but at the cost of HTML becoming more complex to write and maintain. Variations in Web browser implementations made consistent site appearance difficult, and users had less control over how Web content was displayed. (Cascading Style Sheets Wikipedia 2011).

In 1994, after a few days of the World Wide Web Consortium were founded, Håkon Wium Lie published his first draft of Cascading HTML Style Sheet. This draft was a proposal for how HTML documents could be styled using simple declarations. Bert Bos, who was working on a style sheet proposal of his own, responded to Håkon's draft. They dropped HTML from the title, realizing that CSS would be better as a

general style sheet language, applicable to more than one type of document. (York 2005, 13).

Moreover, CSS comes in four different versions, each newer version building on the work of the last. The first version is called CSS level 1 and became a W3C recommendation in 1996. The second version, CSS level 2, became a W3C recommendation in 1998. The third version, CSS level 2.1, is currently a candidate recommendation. (York 2005, 11). The fourth version is called CSS level 3 and its earliest drafts were published in June 1999 (Cascading Style Sheets Wikipedia 2011).

2.2.2 The Definition of CSS

Cascading Style Sheet, referred to as CSS, is a style sheet language used to describe the presentation semantics (the look and formatting) of a document written in a markup language (Cascading Style Sheets Wikipedia 2011). So CSS is a way to control how the browser renders HTML elements (Lemay & Colburn 2006).

Web style sheets are a form of separation of presentation and content for Web design in which the markup of a Webpage contains the page's semantic content and structure, but does not define its visual layout. (Style sheet (web development) Wikipedia 2011).

Cascading means using cascade, it is the cascade that gives CSS developers the most power. There is a hierarchy to be embraced with multiple style sheets dependent on the order and method by which they are applied to the HTML or XHTML. That is the cascade. (Collison 2006, 22).

Most commonly, CSS is combined with the markup languages HTML or XHTML. These markup languages contain the actual text you see in a Web page — the paragraphs, headings, list, and tables — and are the glue of a Web document. They

contain the Web page's data, as well as the CSS document that contains information about what the Web page should look like. (York 2005, 9).

2.2.3 The Features of CSS

HTML has its own limitation that the main aim is to pay close attention to the structure, means HTML is to express the information such as "where headline is", "where table is". For this reason, CSS was introduced to solve the restriction, and it is very useful.

CSS is designed primarily to enable the separation of document content, which is written in HTML or a similar markup language, from document presentation, including elements such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content. (Cascading Style Sheets Wikipedia 2011).

Put simply, CSS handles the look and feel part of a Web page. With CSS, you can control the color of the text, the style of the fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, as well as a variety of other effects. (York 2005, 9).

CSS can also allow the same markup page to be presented in different styles for different rendering methods, for example, on-screen, in print, by voice (when read out by a speech-based browser or screen reader) and on Braille –based tactile devices (Cascading Style Sheets Wikipedia 2011).

2.2.4 The Syntax of CSS

CSS has a simple syntax and uses a number of English keywords to specify the names of various style properties. A style sheet is made up of a list of rules. Each rule or rule-set consists of one or more selectors and a declaration block (Figure 2). (Cascading Style Sheets Wikipedia 2011).
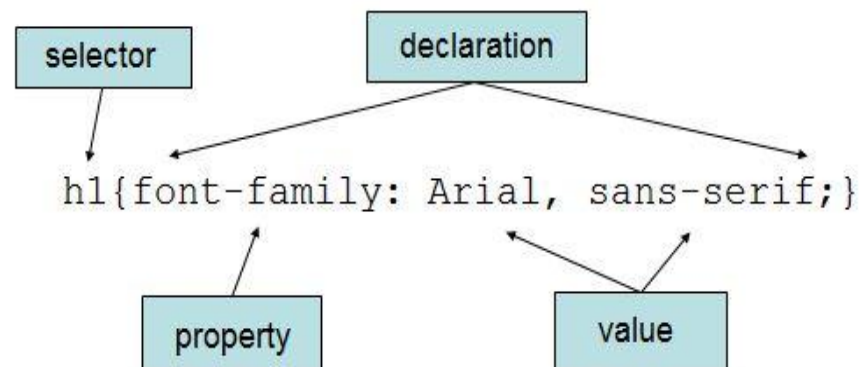
Figure 2. The CSS Rule — (Mynttinen, 2011)

In CSS, selectors are used to declare which of the markup elements a style applies to, a kind of match expression (Cascading Style Sheets Wikipedia 2011). Put simply, the selector tells the browser where and what to format (York 2005, 44). Selectors may apply to all elements of a specific type, or only those elements that match a certain attribute; elements may be matched depending on how they are placed relative to each other in the markup code, or on how they are nested within the Document Object Model. (Cascading Style Sheets Wikipedia 2011).

The declaration block contains one or more declarations in braces. A declaration is always formatted as a property followed by a colon and then a value followed by a semicolon. The colon and semicolon can be followed by zero or more spaces. (Meyer 2007). Like HTML, CSS can use white space and line breaks for purposes of

readability. The interpreter reading the CSS does not care how much white space appears in the style sheet. (York 2005, 44). A value is either a single keyword or a space-separated list of one or more keywords that are permitted for that property. If you use either an incorrect property or value in a declaration, the whole thing will be ignored. (Meyer 2007).

For the style, CSS can be used at varying ways of specificity. You can put all your styles into a separate file, and link to that file from your Web page. That way, if you want to change the appearance of your site, you can simply edit your CSS file and make changes that span every page that links to your style sheet. Or, if you prefer, you can include styles at the top of your page so that they apply only to that page. You can also include styles inside the tags themselves using the style attribute. (Lemay & Colburn 2006).

So to sum up above, there are three different ways to use style. The first is External style sheet, means all the styles that is placed in a completely separate file that has .css as the filename extension. The external style sheet is included in a document from within HTML by use of the <link> element:

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

*** is the .css file name, it stands for the path to your CSS file.

The following attributes are required to use the <link> element for linking to a CSS document. (Mynttinen, 2011).

The second is Internal style sheet, means a collection of styles that are placed in the <head> section of your Web page markup. You can then use the styles from this style sheet to format the Web controls on that page. (Mynttinen, 2011).

The third is Inline style, means the style that is placed directly inside an HTML tag. The style attribute can contain any CSS property. (Mynttinen, 2011).

2.3 PHP

2.3.1 The History of PHP

Sometime in 1994 when Rasmus Lerdorf put together a bunch of Perl scripts to track down who was looking at his resume, gradually, people started to get interested in the scripts; and they were later released as a package "Personal Home Page" tools, this is the first meaning of PHP. In view of the interest, he wrote a scripting engine and incorporated another tool to parse input from HTML forms: FI, Form Interpreter, thereby creating what was called variously PHP/FI or PHP2. This was done around mid-1995. (Castagnetto et al. 1999).

Over the next two years, PHP/FI gradually grew in popularity. Its major benefit was its simplicity; made it easy to handle all the messy tasks thrown at a Web developer. However, by 1997, programmers began to create more complex applications; PHP/FI was already showing its age. The developer also changed from just one person to a group that worked on a new version. This version, which became PHP 3, solved many difficulties faced by PHP/FI while remaining true to the essential nature of PHP/FI. Besides, PHP 3 was easy-to-use extension API and introduced a very simplistic object- oriented syntax. (Trachtenberg 2004, 2 – 3).

A few months after PHP 3 went final in June of 1998, work started on PHP 4, the problem was speed. PHP 4 was released in May 2000, which provided the capability to create larger and more complex web sites. (Trachtenberg 2004, 3).

In 2008, PHP 5 became the only stable version under development and it is still used now (PHP Wikipedia 2011). PHP 5 has many new capabilities, such as better object-oriented features.

2.3.2 The Definition of PHP

PHP is a general-purpose scripting language (PHP Wikipedia 2011). A scripting language is a command set for controlling some specific piece of hardware, software, or operating system, often with rudimentary and in some cases more advanced programming-like control flow constructs, and is almost always usable from a stored format such as a simple text file, a section of read-only persistent storage in an embedded device, a deck of punched cards, or other mechanism (Scripting language Wikipedia 2011). Figure 3 show the logo of PHP.

Figure 3. The Logo of PHP — (PHP Wikipedia 2011)

PHP originally stood for Personal Home Page but was changed in line with the GNU (Gnu's Not Unix) recursive naming convention and now stands for PHP Hypertext Preprocessor. (Welling & Thomson 2009, 3).

PHP is a server-side embedded language. This means that PHP can perfectly merge with HTML, it works within an HTML document to confer the capacity of generating content on demand. (Castagnetto et al. 1999). PHP code is embedded into the HTML source document and interpreted by a Web server with a PHP processor module, which generates the Web page document. (PHP Wikipedia 2011).

PHP language is flexible and quite tolerant; making it is not very hard to learn even if there is no any programming experience in the past. If you already know another programming language, you will almost certainly find similarities here, that PHP looks like a cross between C, Perl, and Java; and that may cause you more easily to adapt the programming style to PHP. (Newman 2005).

2.3.3 The Features of PHP

HTML and CSS provide the ways to establish structure and show page content. However, increasingly, the Internet and intranets are being used for applications, most of which need to incorporate databases. Both these websites and applications are dynamic, because their contents will change according to the data involved and the actions of the user. This is where PHP comes in. (Castagnetto et al. 1999).

PHP is a programming language that was designed for producing dynamic Web pages. It slots into the Web server and processes instructions contained in a Web page before that page is sent through to the Web browser. Certain elements of the page can therefore be generated on-the-fly so that the page changes each time it is loaded. For example, you can use PHP to display the current date and time at your pages. (Newman 2005).

By running PHP programs on the server, users can achieve many more things than just using HTML, which is interpreted by the browser, rather than executed on the server (Castagnetto et al. 1999).

For instance, users can create very powerful applications that interact with a database and generate content dynamically. So PHP can make it easier to edit contents of a Web page by updating the database instead of the HTML code itself; dealing with the data therein by being able to sort the entries into any order or view a subset of them. Besides, PHP can provide customized pages to display only things that will be of

interest to a particular user, getting feedback from the user and then returning information based upon that feedback. Apart from this, creating pages that rotate through a series of different graphics can also be achieved. (Castagnetto et al. 1999).

PHP began life called PHP/FI, the "FI" part standing for Forms Interpreter. Although the name was shortened a while back, one of PHP's most powerful features is still how easy it becomes to process data submitted in HTML forms. For example, you could enter a search keyword into a form field on a web page, and then background will query a database with this value, and produce a page of matching results. (Newman 2005).

In other side, PHP is an Open Source project, unlike commercial, closed-source products, which means that users have access to the source code and can use, modify, and redistribute it all without charge. Users do not need to wait for the manufacturer to release patches, or worry about the manufacturer deciding to stop supporting the product. (Welling & Thomson 2009, 3 – 6).

Moreover, the last version, PHP 5, pay more attention to object-oriented programming (OOP). OOP is a means to group functions and data together into a prepackaged unit that is called an object. OOP encourage a behavior known as encapsulation; the underlying data structures and database tables are not accessed directly. Instead, users define a set of functions and route all requirements through those functions. It will lower the amount of code and let users safely modify the database schema. Objects allow users to wall off the implementation internals from outside access. This prevents people from relying on code that may change and forces them to use the functions to reach the data. (Trachtenberg 2004, 9 – 10).

2.3.4 The Syntax of PHP

PHP is an embedded language (Figure 4). PHP pages, for the most part, end in a .php extension. This extension signifies to the server that it needs to parse the PHP code before sending the resulting HTML code to the viewer's Web browser. As in a restaurant, patrons just see a plate full of beautiful food served up for them. They don't see where the food comes from, nor how it was prepared. In a similar way, PHP fits right into HTML code and is invisible to the people visiting your site. The only thing visitors see when they view the source of code is the resulting HTML output. This gives creators more security for the PHP code and more flexibility in writing it. (Glass et al. 2004).

```
embedded.php

1    <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
2    <html>
3        <head>
4            <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
5            <title></title>
6        </head>
7        <body>
8            <?php
9            // put your code here
10           ?>
11       </body>
12   </html>
13
```
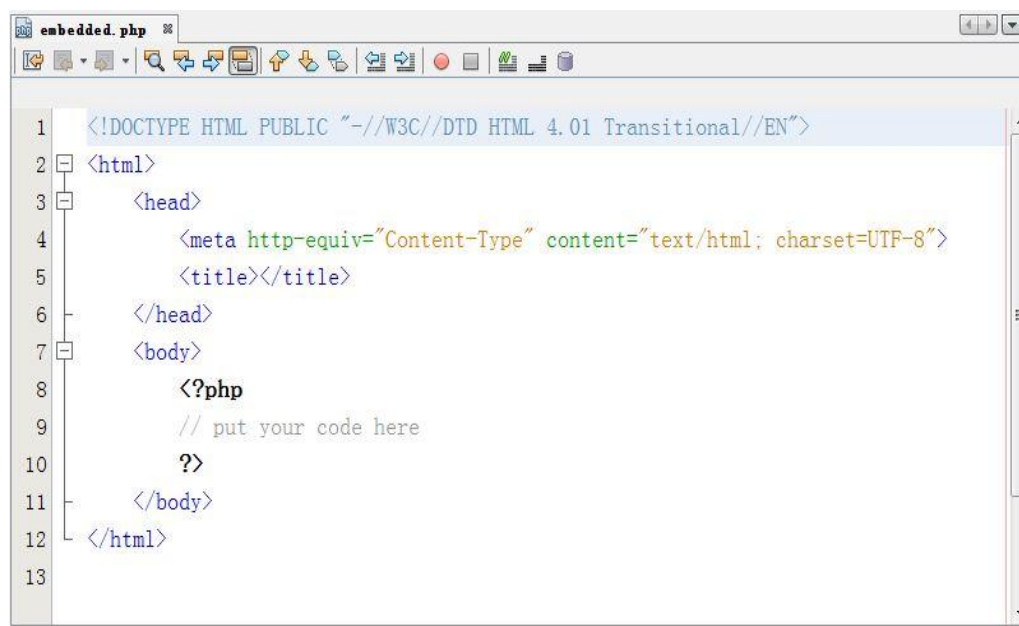
Figure 4. PHP Code embedded within HTML Code

HTML can also be written inside the PHP section; it allows creators to format text while keeping blocks of code together. Besides, PHP can also be written as an independent program, with no HTML at all; it is helpful for storing the connection variables, redirecting the visitors to another page of same site, or performing other functions. (Glass et al. 2004).

The PHP interpreter only executes PHP code within its delimiters. Anything outside its delimiters is not processed by PHP, although non-PHP text is still subject to control structures described within PHP code. The most common delimiters are <?php to open and ?> to close PHP sections (Figure 4). Moreover, generally speaking, PHP lines end with a semicolon (Glass et al. 2004).

Variables in PHP code are prefixed with a dollar symbol and a type does not need to be specified in advance. Unlike function and class names, variable names are case sensitive. (PHP Wikipedia 2011). It can contain only letters, numbers, and the underscore character, and each must begin with a letter or underscore (Newman 2005). Both double-quoted and heredoc strings allow the ability to embed a variable's value into the string (PHP Wikipedia 2011).

The several important functions to PHP code in my mind are to achieve flow control, submit the form, and connect the database. I will explain them in detail with my code in the later chapter.

2.4 MySQL

2.4.1 The History of MySQL

MySQL was developed by a consulting firm in Sweden called TcX. They were in need of a database system that was extremely fast and flexible; but they could not find anything on the market that could do what they wanted. So, they created MySQL, which is loosely based on another database management system called mSQL. (Maslakowski 2000, 9).

The derivation of the name MySQL is not clear. The base directory and a large number of libraries and tools have had the prefix "my" for almost ten years. However,

co-founder Monty Widenius's daughter is also named My. Which of the two gave its name to MySQL is still a mystery. (MySQL AB 2005, 5).

The logo of the MySQL is a dolphin named "Sakila" (Figure 5). The name was submitted by Ambrose Twebaze, an Open Source Software developer from Swaziland, Africa; and chosen by the founders of MySQL AB. (MySQL AB 2005, 5).

Figure 5. The Logo of MySQL — (MySQL Wikipedia 2011)

2.4.2 The Definition of MySQL

MySQL, pronounced My-Ess-Que-Ell, is a fast, strong, relational database management system (RDBMS). A database is a series of structured files on a computer that are organized in a highly efficient manner. These files can store tons of information that can be manipulated and called on when needed. (Maslakowski 2000, 9). The simplest way to think of it is to imagine a database as a filing cabinet. The filing cabinet is simply a physical location to store data, regardless of what that data is or how it is organized. (Forta 2005).

And comes to a relational database, simply defined, it is a database that is made up of tables and columns that relate to one another. These relationships are based on a key value that is contained in a column. (Maslakowski 2000, 11). A relational database

system allows users to separate information into tables or "areas of pertinent information." While in the non-relational, all the information is stored in one big area, which makes it much more difficult and troublesome to sort and extract only the data you want. (Glass et al. 2004).

MySQL is more than just a database; it is a system that manages databases (Maslakowski 2000, 10). The database is the container created and manipulated via the DBMS. Users never access a database directly anyway; they always use the DBMS to accesses the database for them. (Forta 2005). So MySQL controls who can use database and how they are manipulated. It logs actions and runs continuously in the background. (Maslakowski 2000, 10).

MySQL is not a form of this language but a database system that uses SQL, the Structured Query Language, to manipulate, to create, and to show data (Maslakowski 2000, 9). SQL is the most standard language for accessing RDBMSs (Welling & Thomson 2009, 243). It is designed to do one thing and does it well provide users with a simple and efficient way to read and write data from a database. (Forta 2005). MySQL is a program that manages databases, much like Microsoft's Excel manages spreadsheets. SQL is a programming language that is used by MySQL to accomplish tasks within a database, just as Excel uses Visual Basic for Applications (VBA) to handle tasks with spreadsheets and workbooks. (Maslakowski 2000, 9).

2.4.3 The Features of MySQL

A database enables users to efficiently store, search, sort, and retrieve data (Welling & Thomson 2009, 3). It contains a number of tables; each table is made up of a series of columns. Data is stored in rows, and the place where each row intersects a column is known as a field. And a DBMS can contain many databases. Users connect to the database server and issue requests. The database server queries its databases and returns the requests to the issuers. (Maslakowski 2000, 9 – 10).

DBMSs fall into two categories: shared file based and client-server; MySQL is client-server software. Client-server applications are split into two distinct parts. The server portion is a piece of software called database server that is responsible for all data access and manipulation. Only the server software interacts with the data files. All requests for data, data additions and deletions, and data updates are funneled through the server software. These requests or changes come from client software. The client is the piece of software with which the user interacts. Client software submits that request over the network to the server software. Then the server software processes the request; filter, discard, and array data as necessary; and sends the results back to client software. (Forta 2005).

MySQL also is a multi-threaded, Enterprise-level database management system. The Enterprise is the area in the business world where many large systems interact with one another to accomplish a common goal. The computer applications that exist at this level of business tend to have certain characteristics. The first characteristic of an Enterprise-level application is that the same application can be used by more than one person at the same time. Another characteristic is that they provide some sort of security mechanism. The final characteristic is that applications at this level have to be very flexible. (Maslakowski 2000, 9 – 10).

The MySQL server controls access to data to ensure that multiple users can work with it concurrently, to provide fast access to it, and to ensure that only authorized users can obtain access (Welling & Thomson 2009, 3). So it is suitable for Enterprise.

As with PHP, users can obtain and modify the source code for MySQL. This point is not important to most users most of the time, but it provides users with excellent peace of mind, ensuring future continuity and gives users options in an emergency. (Welling & Thomson 2009, 8).

2.4.4 The Syntax of SQL

MySQL use SQL as a language to operate data; unlike other languages, SQL is made up of very few words. It is a language designed specifically for communicating with databases. (Forta 2005). SQL is a nonprocedural language: users specify what information they require, rather than how to get it. It is essentially free-format, which means that parts of statements do not have to be typed at particular locations on the screen. (Mynttinen, 2011)

An SQL statement consists of reserved words and user-defined words. Reserved words are a fixed part of the SQL language and have a fixed meaning. They must be spelled exactly as required and cannot be split across lines. User-defined words are made up by the user and represent the names of various database objects such as tables and columns. (Mynttinen, 2011). A table is a structured file that can store data of a specific type. Every table in a database has a unique name that identifies it. And tables are made up of columns. A column contains a particular piece of information within a table. (Forta 2005).

Most components of an SQL statement are case-insensitive, but the literal character data must be typed exactly as it appears in the database table (Mynttinen, 2011). Typically, keep the MySQL commands in all caps, although this is not necessary; the purpose of this is to help keep the MySQL syntax separate from the variables and table or database names (Glass et al. 2004).

The command structure consists of English words, including such as CREATE that can create new databases and tables; ALTER that can modify existing tables; SELECT that can choose the data you want; DELETE that can remove the data from the table; UPDATE tablename SET that can make you modify data already in a table; INSERT INTO tablename VALUES that can add values into the table; DROP that can delete an entire table or database. (Glass et al. 2004).

When comes to the connection of PHP and MySQL, there are MySQL commands within PHP code almost as seamlessly as you do with HTML. This is important for accessing database through PHP, mainly including: (Glass et al. 2004).

Mysql_connect ("hostname", "user", "pass"): Connects to the MySQL server.

Mysql_select_db("database name"): Makes the selected database the active one.

Mysql_query("query"): Used to send any type of MySQL command to the server.

Mysql_fetch_array("results variable from query"): Used to return several rows of the entire results of a database query.

Mysql_error(): Shows the error message that has been returned directly from the MySQL server.

3   INTRODUCE PROGRAMMING ENVIRONMENTS

In my project, I will use NetBeans IDE as my programming tool to compile codes; and WampServer as the environment to run codes.

3.1 NetBeans

NetBeans refers to both an open platform framework for Java desktop applications, and an open-source integrated development environment (IDE) for developing with Java, JavaScript, PHP, Python, Ruby, Groovy, C, C++, Scala, Clojure, and others (NetBeans Wikipedia 2011a). It is an open source software development tools created by Sun Microsystems (NetBeans Wikipedia 2011b).

NetBeans is an extensible development platform; it can extend the functionality through extension plugins. NetBeans Platform is a reusable framework for simplifying the development of other desktop applications. (NetBeans Wikipedia 2011b). It allows applications to be developed from a set of modular software components called modules. Applications based on the NetBeans platform, including NetBeans IDE, can be extended by third party developers. (NetBeans Wikipedia 2011a).

NetBeans IDE is much more than a Java IDE; it is designed as a modular developer tool for a wide range of development tasks. The base IDE includes an advanced multi-language editor, debugger and profiler integration, file versioning control, and unique developer collaboration features. (Base IDE 2011). All the functions of the IDE are provided by modules; each module provides a well-defined function (NetBeans Wikipedia 2011a).

The latest release version of NetBeans IDE is NetBeans IDE 7.0; and the version I used is NetBeans IDE 6.9.1. Figure 6 shows the platform I used. Besides, there are several kinds of NetBeans IDE Download Bundles that can be chosen by users (Figure

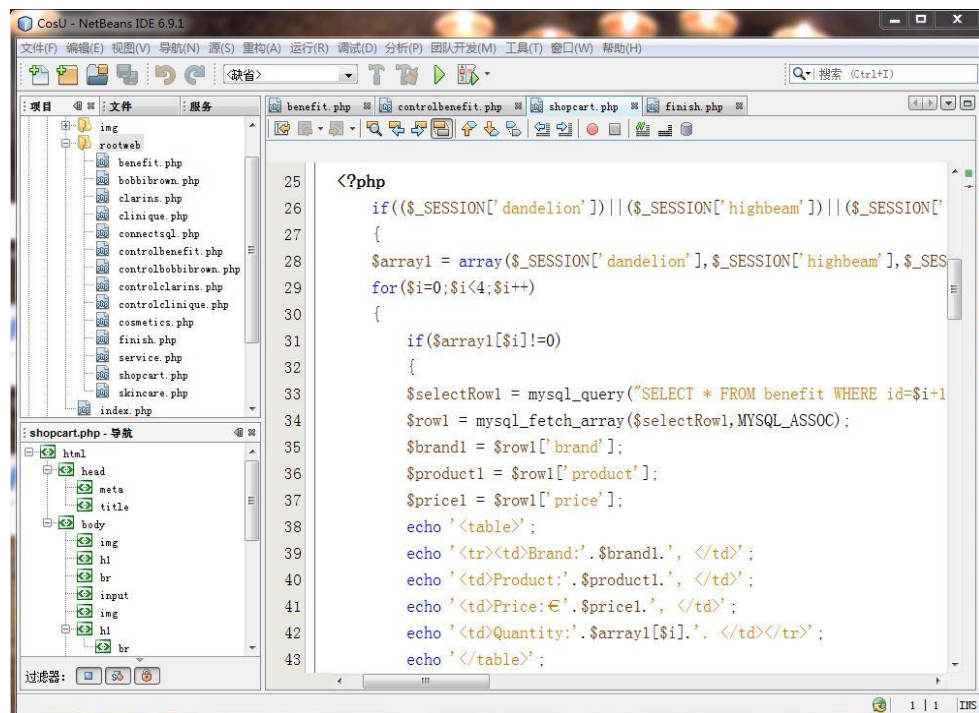7). And I chose the NetBeans IDE Bundle for PHP, which is suitable for completing my project.

Figure 7. The Platform of NetBeans IDE 6.9.1

Figure 8. NetBeans IDE Download Bundles — (NetBeans IDE 7.0 Download 2011)

3.2 WampServer

WAMPs are packages of independently-created programs installed on computers that use a Microsoft Windows operating system. WAMP is an acronym formed from the initials of the operating system Microsoft Windows and the principal components of the package: Apache, MySQL and one of PHP, Perl or Python. (WAMP Wikipedia 2011).

WampServer is a Windows web development environment. Its installer installs automatically, and usage is very intuitive. It also has a tray icon to manage server and its settings (Figure 9). (WampServer Presentation 2011). I chose WampServer as my PHP running environment, and saved my PHP files in a "www" directory under C:\wamp. When running the PHP files, I will open the browser and enter the address http://localhost (Figure 10), then click my project. Besides, WampServer also comes with phpMyAdmin which provides a graphical user interface for the MySQL database manager (WAMP Wikipedia 2011). The phpMyAdmin helps me easily create and modify tables in my database. I will illustrate it in detail in the later chapter.
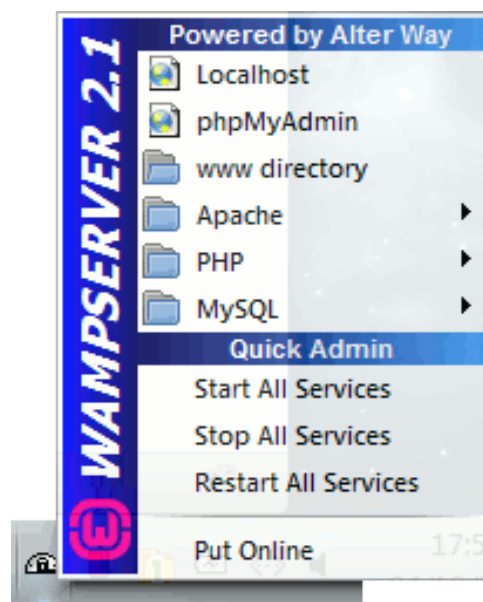

Figure 9. The Tray Icon of WampServer — (WampServer Presentation 2011)

Figure 10. The Homepage of WampServer — (WampServer Presentation 2011)

4 IMPLEMENT MY PROJECT USING SESSION CONTROL

The topic of my final thesis is "Session control using PHP and MySQL". So I chose to build an online shopping website, and it supports the function of shopping cart, which is achieved by using session control. Session control can be used for achieving many functions; the reason why I chose the function of shopping cart is that it is more intuitive and suitable for express session control.

4.1 Session Control

Almost all people know that HTTP is a stateless protocol, which means that the protocol has no built-in way of maintaining state between two transactions (Welling & Thomson 2009, 509). The Web server considers each visit of each client as an independent visit; it does not maintain any previous access information, even if the visit happened one second ago. The process to access a website is continuous and regular for customers; but for Web server, it is composed of many unrelated continuous requests. (Xue et al. Translation 2001).

For my project, if there is no session control, the situation like this will happen: The customer has already ordered a commodity, and then seeing another one he also wants; however, at the time, the Web server cannot know both two commodities, it just knows the later one.

The idea of session control is to be able to track a user during a single session on a website (Welling & Thomson 2009, 509). The key lead to success of session control is how to distinguish between different clients. That is session ID. Each time when a client browser sends a new request to the server, it will generate a session ID to declare the identity. (Xue et al. Translation 2001). Sessions in PHP are driven by this unique session ID, which is a cryptographically random number. The session ID is generated by PHP and stored on the client side for the lifetime of a session. It can be

either stored on a user's computer in a cookie or passed along through URLs. (Welling & Thomson 2009, 509). When receiving the session ID, server will check whether it exists. If it does not exist, then simply create it and build some new data according to it. If it already exists, just extract the data that associated with it. (Xue et al. Translation 2001).

People who have built websites have such experience that variables in one page cannot be used in another page. Session can solve the problem. Variables, which are registered in session, can be used as global variables. That means these session variables can be used in all of the programs on this website. (Xue et al. Translation 2001).

The basic steps of implementing session control are firstly start a session, and then register session variables, afterwards use session variables, finally deregister variables and destroy the session. These steps do not necessarily all happen in the same script, and some of them happen in multiple scripts. (Welling & Thomson 2009, 512). I will use my codes to explain it in details.

4.2 The Design of My Project

My project is to build an online shopping website, which sells cosmetics. The reason why I choose to sell cosmetics is that I feel cosmetics are necessities for most of women; selling cosmetics is very popular, useful, and has lots of profits. Besides, I use cosmetics in my daily life, so I am familiar with some cosmetics to a certain degree.

My website is divided into several levels. The Homepage is about the introduction of website, including my trademark, the discount products and the best sellers. The next level pages of homepage are three main parts in my website. There are Cosmetics, Skincare, and Service.

One thing I want to say is about the Customer Service. Online shopping is different from shopping in a real Department Store. Customers can really see, touch, and try the product they want to buy in the shopping mall; however, it cannot be realized on the internet. Customers just can see the pictures and description texts of the commodities, so there will be more likely to have service problems than shopping in a real market. Moreover, if the commodity has some problems, the clientele can consult with the salesperson face to face; it is easier to solve the problems.

Then comes to the online shopping, in my mind, the Customer Service link is more considerable. So in my website, I set up a feature specially to send email to the service mail box. If the products have any problems, clienteles can simply send an email to describe the problem, then the service staff will help to solve the problem. It will make the process more effective and convenient. In view of the importance of Customer Service link, although it has some troubles now, which is the email cannot send out because I do not have the SMTP Server to support this function, I still choose to retain it.

Back to another two pages, Cosmetics and Skincare, these two pages can directly jump to each other by links. It can make clienteles more convenient to browse different kinds of products. And there are links of two brands separately. Each brand has several commodities.

When the clientele selects a product, the link will bring him or her to the next level page in confirming or canceling the previous operation. It will display the product information in this page. It is the second chance to the customer to think over whether to purchase this product or not to. Most online shopping websites do not have this page, the reason why I choose to set it is that I think sellers should give customers chances in determining the consumption to reduce unnecessary consumptions. Besides, it can avoid the disoperation of customers. From the technical level, it will facilitate

my work of compiling codes; and it can avoid some wastage of data resources in transmitting.

After the confirmation page, if the clientele clicks the Confirm button, the link will bring him or her to the shopping cart. If not, the clientele will back to the Cosmetics or Skincare page.

In the shopping cart page, customers can choose to continue buying or settle accounts. Customers do not need to worry about the commodities, which they already chosen, if they select to continue buying; those commodities will still in their shopping cart when they choose new products and come to this page again. That is the fascination of session control.

However, session has its lifetime. Unlike cookie, the values of session variables will be cleared after closing the browser. So if the clientele does not settle accounts and close the browser, the commodities in shopping cart will disappear.

When the clientele finish shopping and click the Settle Accounts button, ostensibly, it will display "Dear Customer: Your transaction has been successfully processed." on the next level page. Actually, the background will update database according to the new quantity of products. That is the whole process of designing my website.

4.3 Explain Significant Codes

There are lots of codes in whole program; I am going to explain parts of them, which are considerable to the entire project.

4.3.1 The Creation of Database

I need to build up database before achieving combination of PHP and MySQL. There are fixed statements and format to create database. However, I selected another method to build up database. I use the WampServer as the running environment, it comes with phpMyAdmin, which provides a visual interface to create database (Figure 11).
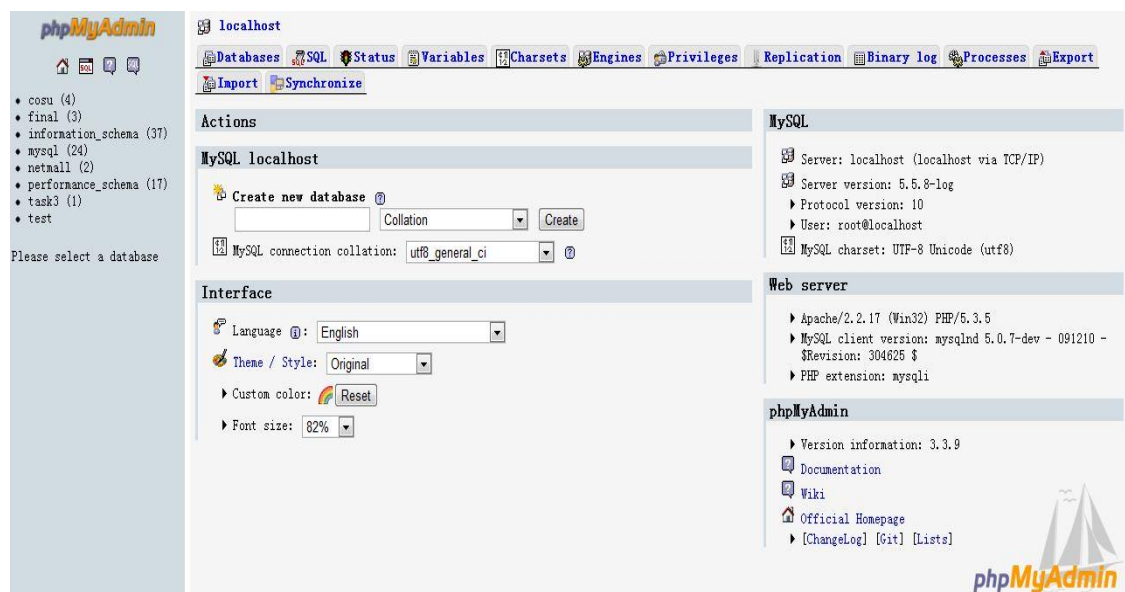

Figure 11. The Interface of phpMyAdmin

Here are the basic steps of creating my database by using phpMyAdmin. Initially, I need to input the name of database, and then click the Create button (Figure 12). The second point I want to make is input the name of table and the number of fields, then click the Go button (Figure 13). This brings me to the third step, fill in the names and types of each field, then click the Save button (Figure 14). Finally, I need to click the Insert button; it will lead me to fill in the values of each field, then click the Go button (Figure 15). After these four steps, one row in a table is completed. According to the basic steps, after several rows are inserted, Figure 16 shows the view of a complete table.

Figure 12. Create Database

Figure 13. Create Table

Figure 14. Fill in the Names and Types of Each Field

Figure 15. Fill In the Values of Each Field

Figure 16. The View of a Complete Table

There are totally four tables in my database, stand for four different brands respectively. The reason why I choose to create four tables is that it will make me more convenient to find and use the data in PHP.

4.3.2 The Significant Codes

Initially, I am going to illustrate the process of starting my project by using the NetBeans IDE 6.9.1. Firstly, select new project and choose PHP (Figure 17), then click the Next Step. Secondly, fill in the Project Name and set the Source Folder address to C:\wamp\www (Figure 18), then click the Next Step until Complete. After done the previous steps, it is the turn to create new files. I choose to create PHP Web page, because it can compile HTML codes and PHP codes in the same page. After all of these are completed, I am able to begin my work of programming.
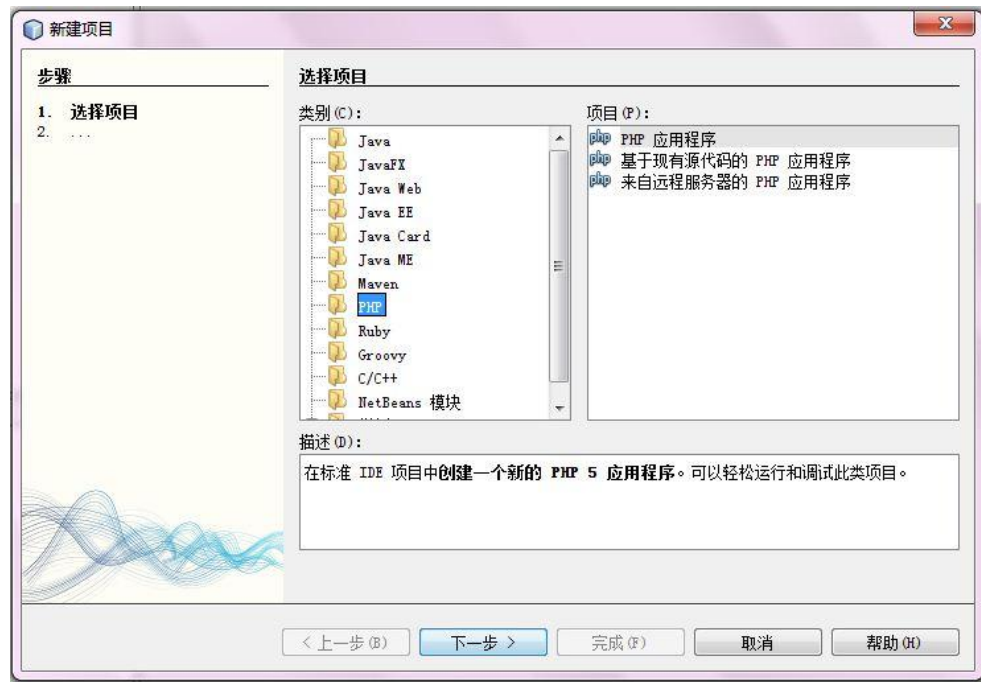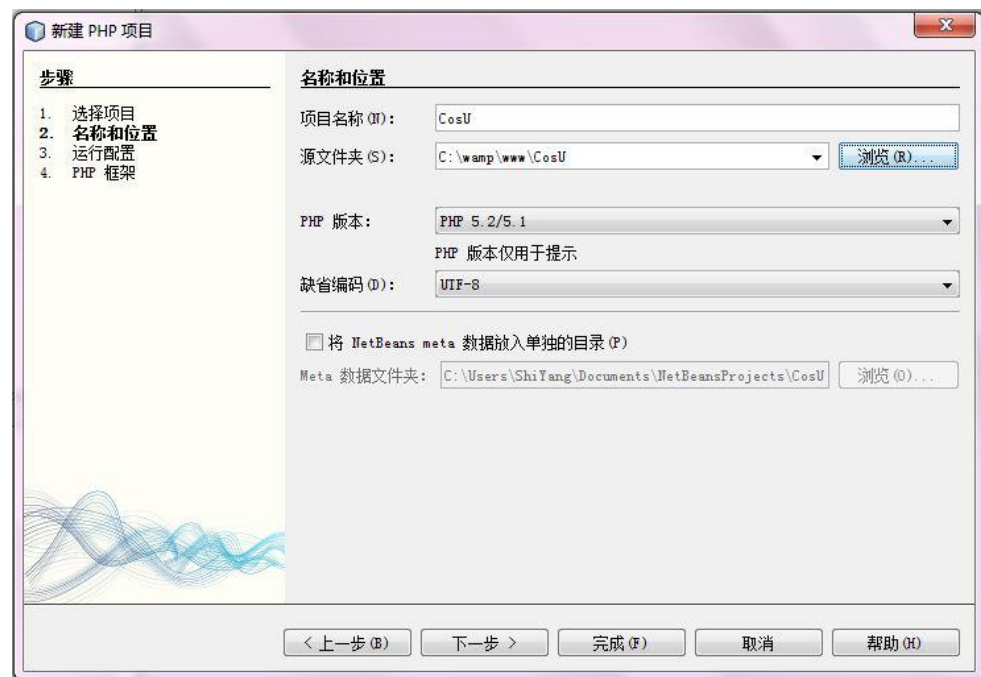
Figure 17. New Project

Figure 18. Fill in Project Name and Source Folder

I use many tables in my program to make the page layout become tidy. It is powerful and easy to use. I do not consider it is necessary to explain some codes specially.

There is one point I want to explain that about HTML forms and interactivity. Here is a certain part of codes in my program.

```
<tr>
    <td align="center" colspan="4">
        <input type="button" value="Confirm & Go to Shopping Cart"
        onclick="window.location.href='shopcart.php'">
    </td>
</tr>
<tr>
    <form action="cosmetics.php" method="post">
        <td align="center" colspan="4">
            <input type="submit" name="cancelbenefit" value="Cancle & Back
            to Cosmetisc">
        </td>
    </form>
</tr>
```

There are two rows in a table. The first row is a button of jumping to "shopcart.php" without any interactivity; the second row is about form and interactivity. In my codes, I use the processing method "post" in form. The processing method refers to what happens to the form after the user enters all the data and presses the Submit button (Mynttinen, 2011). It still has another method "get", the different between them is that the "post" method can hide what has been send in URL, while the "get" method cannot. So I choose "post" method, it is more secure. The action is necessary, it tells the browser to deal with the variable $_POST['cancelbenefit'] in "cosmetics.php". There are several kinds of forms, such as input text, radio, checkbox, button, and select; most of them were used in my program.

Then comes to PHP, the first point I am going to expound is about extracting data in database using PHP statements. Here are the codes.

require ('connectsql.php');

*$selectRow1 = mysql_query("SELECT * FROM benefit WHERE id='1'");*

$row1 = mysql_fetch_array($selectRow1,MYSQL_ASSOC);

$quantity1 = $row1['quantity'];

I give a name called Jelly Beans to this problem. It takes me amount of time to solve the problem. I need to connect database every time when I want to use data, therefore, I put the two statements in the file "connectsql.php", and call it when needed. Back to the Jelly Beans problem, at the beginning, I want to use the simplest method to extract a data; just use one command to exactly locate and extract the data. But it cannot implement. After several attempts, I found that it needs three statements to extract a data. Finally I have two ways to achieve it

I adopted the first method. In my mind, I use jelly beans instead of data to understanding these statements. I want to pick up a jelly beans from packet. The first command is to take a "row" of jelly beans from the packet. The second command is to put these jelly beans in a row of boxes separately. The last statement is to pick up one box with the jelly beans that wanted.

The other method is firstly take one jelly beans from the packet, then put it in a box, at last, pick up the box with the jelly beans that wanted. The statements are:

$selectRow1 = mysql_query("SELECT quantity FROM benefit WHERE id='1'");

$row1 = mysql_fetch_assoc($selectRow1);

$quantity1 = $row1['quantity'];

In addition, I use select form and *for* loop to show the data that is just selected. It is a number, and I choose to display the whole sequence from zero.

```
<select name="dandelion">
<?php
for($i=0; $i<=$quantity1; $i++)
{
    echo '<option value="'.$i.'">'.$i.'</option>';
}
?>
</select>
```

The next point I want to explain is about updating the quantity of products. After clienteles buying commodities, the quantity of products will have new values. It is necessary to update them. Here are the codes.

```
<?php
session_start();
require ('connectsql.php');

$array1 = array ($_POST['dandelion'], $_POST['highbeam'], $_POST['smokin'],
        $_POST['lip']);
$array2    =    array    ($_SESSION['quantity1'],    $_SESSION['quantity2'],
            $_SESSION['quantity3'], $_SESSION['quantity4']);
$array3 = array();

for ($i=0;$i<4;$i++)
{
    $array3[$i] = $array2[$i]-$array1[$i];
}
```

```
$_SESSION['last1']=$array3[0];

$_SESSION['last2']=$array3[1];

$_SESSION['last3']=$array3[2];

$_SESSION['last4']=$array3[3];


for($i=0;$i<4;$i++)

{

    if($array1[$i]!=0)

    {

        $selectRow = mysql_query("SELECT * FROM benefit WHERE id=$i+1");

        $row = mysql_fetch_array($selectRow,MYSQL_ASSOC);

        $brand = $row['brand'];

        $product = $row['product'];

        $price = $row['price'];

        echo '<tr><td><b>Brand: </b>'.$brand.', </td>';

        echo '<td><b>Product: </b>'.$product.', </td>';

        echo '<td><b>Price: </b>€'.$price.', </td>';

        echo     '<td><b>Quantity:    </b>'.$array1[$i].'.    </td></tr><tr><td

              colspan="4"> </td></tr>';

    }

}


mysql_close($connectsql);

?>
```

The first half parts of these codes are about calculating the new values of products'
quantity. When the clientele clicks the Buy Now icon, all numbers in select forms will
send to the next level file. *$array1* put all the numbers of buying into an array.
$array2 put the total quantity of each product into an array. *$array3* is an empty array.
After this, I do the subtraction to get the new values of each product by using *for* loop,

and then put these values into *$array3*. Until now, I have the new quantity of each commodity in *$array3*, and I can use them separately. The rest parts of these codes are about displaying the product information by using tables in PHP. Besides, each time before I use session variables, I should start the session firstly. In addition to these, there still have some codes about updating data.

```php
<?php
session_start();
require("connectsql.php");

if(($_SESSION['dandelion'])||($_SESSION['highbeam'])||($_SESSION['smokin'])||($_SESSION['lip']))
{
mysql_query("UPDATE benefit SET quantity='$_SESSION[last1]' WHERE id=1");
mysql_query("UPDATE benefit SET quantity='$_SESSION[last2]' WHERE id=2");
mysql_query("UPDATE benefit SET quantity='$_SESSION[last3]' WHERE id=3");
mysql_query("UPDATE benefit SET quantity='$_SESSION[last4]' WHERE id=4");
}

mysql_close($connectsql);
session_unset();
session_destroy();
?>
```

These codes are to update database using the new values just calculated. It mainly uses SQL update command combined with PHP. Moreover, when the session is finish, I firstly unset all the session variables, and then destroy the session. After this, both the session variables and the session are cleared.

The shopping cart page is the most important page to reflect session control. It also takes me much time to compile the codes successfully. At the beginning, it always has troubles; the error shows that the session variables, which are stand for the quantity of buying, do not create. Then I know the problem is the location of creating session variables is not suitable. After several attempts, finally I decided to create those variables in the homepage, and set the value of all of them to zero. Besides, if a value need to use in several pages, I assign all of them to session variables.

```php
<?php
session_start();
$_SESSION['dandelion']=0;
$_SESSION['highbeam']=0;
$_SESSION['smokin']=0;
$_SESSION['lip']=0;
$_SESSION['blushcompact']=0;
$_SESSION['mascara']=0;
$_SESSION['foundation']=0;
$_SESSION['exfoliate']=0;
$_SESSION['moisturize']=0;
$_SESSION['eyerepair']=0;
$_SESSION['oil']=0;
?>
```

Although the shopping cart can successfully execute, it brings a new problem to me. If a clientele has already bought a commodity, he or she cannot back to homepage; otherwise, information about this commodity will disappear. I did not solve this problem until now, so I choose a compromise solution, which is I do not set the link to homepage during the process of buying. Apart from homepage, all of other pages can jump to each other. So I do not fell that it will have much effect. Here is part of codes to implement shopping cart.

```php
<?php
if(($_SESSION['dandelion'])||($_SESSION['highbeam'])||($_SESSION['smokin'])||($
_SESSION['lip']))
{
$array1    =    array($_SESSION['dandelion'],    $_SESSION['highbeam'],
              $_SESSION['smokin'], $_SESSION['lip']);


for($i=0;$i<4;$i++)
{
    if($array1[$i]!=0)
    {
    $selectRow1 = mysql_query("SELECT * FROM benefit WHERE id=$i+1");
    $row1 = mysql_fetch_array($selectRow1,MYSQL_ASSOC);
    $brand1 = $row1['brand'];
    $product1 = $row1['product'];
    $price1 = $row1['price'];
    echo '<tr><td><b>Brand: </b>'.$brand1.', </td>';
    echo '<td><b>Product: </b>'.$product1.', </td>';
    echo '<td><b>Price: </b>€'.$price1.', </td>';
    echo '<td><b>Quantity: </b>'.$array1[$i].'. </td></tr><tr><td colspan="4">
            </td></tr>';

    }
}


}
?>
```

The most considerable codes among these codes are the first two lines. It is to judge

whether the customer had bought the product or not, in order to determining whether

to display the product information or not to. The following codes are to display the selected information using *for* loop. Besides, I designed two different kinds of page styles according to whether the shopping cart is empty or is not.

There is also a problem that has troubled me for an amount of time. It is about the occasion of assignment or reset to variables. I have the confirmation page between the commodity page and the shopping cart page; customers can confirm or cancel the products they just have chosen to buy. So it will involve a problem that when shall the assignment be done. I tried quite a lot of methods, at least, I determined that directly assign values to the session variables before clientele click Confirm or Cancel button, and then control the reset by using conditions in the next level of HTML form. If the customer clicks the Confirm button, it will bring him or her to the shopping cart. If the customer clicks the Cancel button, it will bring him or her to Cosmetics or Skincare page, and implement the operation of reset in the background. Here are parts of the assignment codes and reset codes.

Assignment:

```php
<?php
session_start();
$_SESSION['dandelion']=$_POST['dandelion'];
$_SESSION['highbeam']=$_POST['highbeam'];
$_SESSION['smokin']=$_POST['smokin'];
$_SESSION['lip']=$_POST['lip'];
?>

<table border="0" align="center">
<tr>
    <td align="center" colspan="4">
        <input type="button" value="Confirm & Go to Shopping Cart"
            onclick="window.location.href='shopcart.php'">
```

```
                </td>
        </tr>
        <tr>
                <form action="cosmetics.php" method="post">
                    <td align="center" colspan="4">
                        <input type="submit" name="cancelbenefit" value="Cancel & Back
                            to Cosmetics">
                    </td>
                </form>
        </tr>
    </table>
```

Reset:

```
<?php
session_start();
if(isset($_POST['cancelbenefit']))
{
    $_SESSION['dandelion']=0;
    $_SESSION['highbeam']=0;
    $_SESSION['smokin']=0;
    $_SESSION['lip']=0;
}
?>
```

4.3.3 The Decoration of Web Page

Friendly page, in my mind, needs to be achieved by two main points. One is the page layout should be clear, the other is the page design needs be beautiful. Besides, there supposed to be links among each page, so that visitors can browse conveniently.

Since I have a little challenge with index page session control variables issue, it is forbidden in linking back to homepage during purchasing process. I have already mentioned it in previous section. To make visitors easy to browse my website, apart from homepage, visitors can jump among all other pages.

To the page layout issue, I used HTML tables frequently to make the distribution more orderly. And most of tables are not separate, but nested. In addition, I set a Back to Top button on the page, which uses long tables to show the information. The button can help visitors quickly back to the top of the page. Here is the code to achieve it.

At the location of heading:

```
<table border="0" align="center">
<tr>
    <a name="top"> <td width = "171px" height = "119px"> <img src=
      "./img/Logo.png" alt="cosu" width="90%"> </td> </a>
    <td style = "font-family: Giddyup Std; font-size: 60pt"> <b> -Welcome to
      CosU- </b></td>
</tr>
</table>
```

At the end of code:

```
<center>
    <input type = "button" value = "Back to Top" onclick =
    "window.location.href='#top'">
</center>
```

When it comes to embellish a Web page, people who have ever built website will immediately think of CSS. It is a powerful programming language in beautifying Web pages.

There are three kinds of styles to implement CSS, inline, internal, and external style. Each of them has their own advantages and disadvantages. I chose to use the inline and internal style. I do not want to make too many files, so I chose these two styles, which are not need to be supported by creating individual CSS file.

Internal style provides an overall design to the involved tag in the whole file. Here is an example in my program.

<head>

    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

    <title>Welcome to CosU---Your Best Cosmetics & Skincare Choice</title>

    <link rel="icon" href="./img/Logo.png" type="image/x-icon" />

    <style type="text/css">

        a:link { text-decoration: none; color: blue}

        a:active { text-decoration: none; color: blue}

        a:hover { text-decoration: none; color: blue}

        a:visited { text-decoration: none; color: blue}

    </style>

</head>

This is to set the style of all the links on index page. The default link style is blue words with underline; and if the link has been visited, its word color will change to purple. In my link style, I remove the underline, and set the color still in blue after the link is clicked. Between *<style type="text/css">* and *</style>*, *a:link* is for all unvisited links; *a:active* is for the active link; *a:hover* is for the links on mouse over; *a:visited* is for all visited links. Set the *text-decoration* to *none* will achieve the first point.

Moreover, I designed a trademark for my online shopping website. I put it not only in the heading, but also in the background and title (Figure 19). The code to make the picture in title is *<link rel="icon" href="./img/Logo.png" type="image/x-icon" />*.
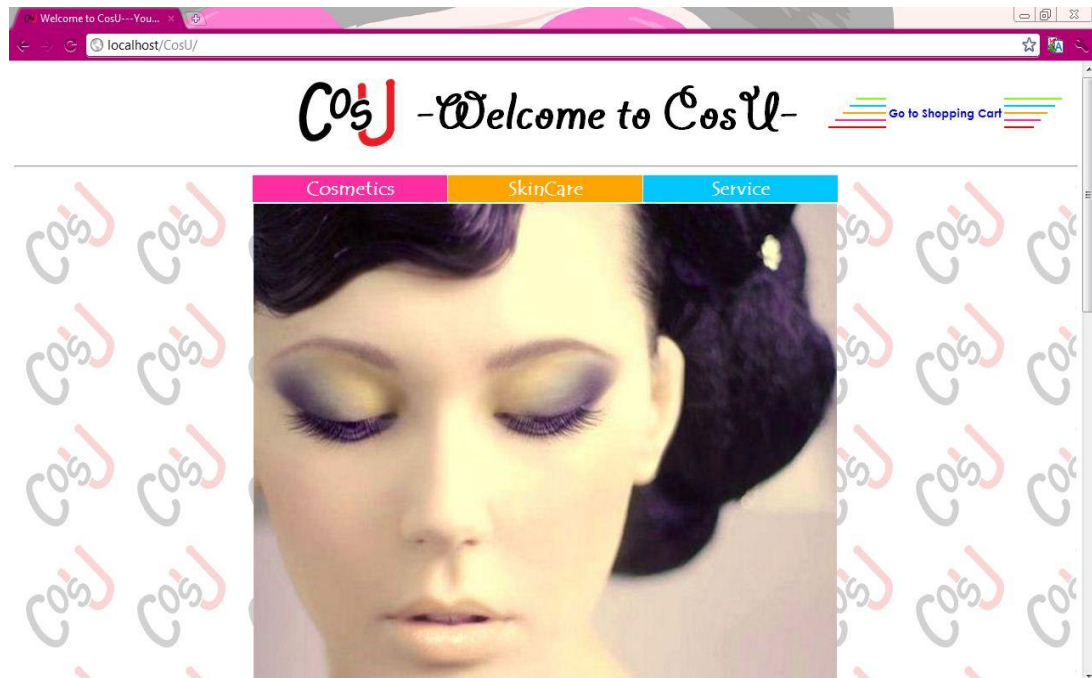

Figure 19. Using Picture in Title

Inline style can design different appearance to the same type of tags. PHP statement is executed line by line from top to bottom. And the priority of inline style is higher than internal style. So the background will execute the commands with inline style.

I modified the picture by using Photoshop, and apply it as the page background. Because I just want the background under the main table, I put the code inside the table tag.

<table border = "0" width = "100%" style = "background – image : url ('. /img/
    background.png ')">

...

</table>

There are some bright spot easily be overlooked. As can be shown in Figure 19, the design of the link Go to Shopping Cart. Here is the code to implement it.

```
<td align="right" width="100px">
    <hr width="40%" color="rgb(128,255,0)" align="right">
    <hr width="50%" color="rgb(0,200,255)" align="right">
    <hr width="60%" color="rgb(255,165,0)" align="right">
    <hr width="70%" color="rgb(255,45,160)" align="right">
    <hr width="80%" color="rgb(255,0,0)" align="right">
</td>
<td align = "center" width = "160px"> <a href = "./rootweb/shopcart.php" style =
"font-family : Century Gothic"> <b> Go to Shopping Cart </b> </a>
</td>
<td align="right" width="100px">
    <hr width="80%" color="rgb(128,255,0)" align="left">
    <hr width="70%" color="rgb(0,200,255)" align="left">
    <hr width="60%" color="rgb(255,165,0)" align="left">
    <hr width="50%" color="rgb(255,45,160)" align="left">
    <hr width="40%" color="rgb(255,0,0)" align="left">
</td>
```

I selected five bright colors, which is indicated by three numbers of rgb. The left lines are all right-aligned; on the contrary, the right lines are all left-aligned. The code *width="40%"* is to set the line length. The percentage is specify the length of *<hr>*.

Besides, I retained the HTML table borders. During the process of embellishing pages in later period, I originally set all the table borders to zero, but I found that the page layout would become a bit messy. So I finally keep the borders and change them into dotted lines (Figure 20).

Figure 20. The Border with Dotted Lines

Here is the code to implement the dotted lines and to change the border color.

```
<style type="text/css">
    th {border-style: dotted}
    td {border-style: dotted}
</style>

<table border="0" width="100%" style="border: 2px dotted">
...
</table>

<table border="2" align="center" style="border-color: rgb(255,45,160)">
...
</table>
```

5 CONCLUSION

Although I spent several weeks in completing the whole project, before starting the programming job, it took me a great deal of time to prepare. Firstly, I reviewed the contents have learned from the course Internet Application Development. Then I focused on learning about session control. With a few small practices, I have a basic understanding of using sessions. After this, I started my work of programming.

During the whole process of accomplishing the project, I met quite a lot of difficulties, but I never thought of giving up. By constantly modifying the code and test results; I finally solved the problems one by one. At that moment, the kind of excitement cannot be expressed in words.

Of course, there are some problems for which I did not find solutions. In this case, I chose to achieve the results by other means. For instance, when I compiled the code in showing the product quantity by HTML select form, I originally set the numbers increasing from one, and displayed the words "Out of Stock" when the product quantity is zero. I can successfully show this information by using *if* statement. However, I need to compute the new product quantity utilizing the data submitted by form, the texts cannot be calculated. At last, I chose to set the numbers increasing from zero. It can not only be calculated, but it also indicates that there are no commodities.

By completing the project, I gained a lot. This is the first time I accomplished such a big project individually in a relatively short period of time. It is a great challenge for me. It exercises my learning ability to a large extent. I need to learn new knowledge by myself, and apply the theory to the practice. In my mind, if you want to learn program, you must have practice. Theory cannot change into codes; you must compile codes little by little, constantly modify and test results.

By writing the final thesis, I learned the theoretical knowledge systematically. Besides, to a certain degree, it improved my English skills. The next period of time, I will continue my studies in the United Kingdom. So it is a significant experience for my future learning.

Although I have already completed the project, as many reasons, I still have some ideas that I did not achieve. For example, I want to build a login system. A login system, including registration and login two functions, is a basic part of online shopping website. It requires skillfully using HTML form, combining with PHP and MySQL. Inserting information into database and matching information with database are also needed. In addition, to session control, it is a very important application. The login system can identify users, so that to provide different page content to different users. It will make the website independent to individual visit, and it will implement personalized service to those Registered Users.

This is not the end; I will continue to improve my project as far as possible. I will try my best to achieve the ideas, which are still bothering me, in my spare time.

BIBLIOGRAPHY

Books

Bowers, Michael 2007. Pro CSS and HTML Design Patterns. The United States of America.

Castagnetto, Jesus – Rawat, Harish – Schumann, Sascha – Scollo, Chris – Veliath, Deepak 1999. Professional PHP Programming. The United States of America. Wrox Press Ltd.

Collison, Simon 2006. Beginning CSS Web Development: From Novice to Professional. The United States of America.

Glass, Michael K. – Scouarnec, Yann Le – Naramore, Elizabeth – Mailer, Gary - Stolz, Jeremy – Gerner, Jason 2004. Beginning PHP, Apache, MySQL Web Development. Canada.

Graham, Ian S. 1998. HTML 4.0 Sourcebook. Canada. Robert Ipsen.

Mack, E. Stephen – Platt, Janan 1997. HTML 4.0. Gary Masters.

Maslakowski, Mark 2000. Sams Teach Yourself MySQL in 21 Days. The United States of America.

MySQL AB 2005. MySQL Language Reference. The official guide to the MySQL language and APIs. The United States of America.

Trachtenberg, Adam 2004. Upgrading to PHP 5. The United States of America.

Welling, Luke – Thomson, Laura 2009. PHP and MySQL Web Development. Fourth Edition. The United States of America. Addison-Wesley.

Xue, Zhong Sheng – Shi, Zheng – Yang, Deng Feng – Zheng, Feng et al. Translation 2001. Castagnetto, Jesus et al. Original 1999. Professional PHP Programming. Chinese translation. China Machine Press.

York, Richard 2005. Beginning CSS: Cascading Style Sheets for Web Design. Canada. Wiley.

Unpublished sources

Mynttinen, Timo 2011. Internet Application Development. Course Slides. Mikkeli University of Applied Sciences.

Electronic sources

Base IDE. WWW-document. http://netbeans.org/features/ide/index.html. No update information available. Referred 4.5.2011.

Cascading_Style_Sheets. WWW-document. http://en.wikipedia.org/wiki/Cascading_Style_Sheets. Updated 24.4.2011. Referred 26.4.2011.

Forta, Ben 2005. MySQL Crash Course. The United States of America. Sams Publishing. WWW-document. http://www.ithov.com/book/data/mysql/23038.shtml. Updated 6.10.2007. Referred 29.4.2011.

HTML. WWW-document. http://en.wikipedia.org/wiki/HTML. Updated 25.4.2011. Referred 26.4.2011.

Lemay, Laura – Colburn, Rafe 2006. Sams Teach Yourself: Web Publishing with HTML and CSS in One Hour a Day. The United States of America. Sams Publishing. WWW-document. http://www.ithov.com/book/website/css/36833.shtml. Updated 10.3.2010. Referred 26.4.2011.

Meyer, Eric A. 2007. CSS: The Definitive Guide. Third Edition. The United States of America. O'Reilly. WWW-document. http://www.ithov.com/book/website/css/31681.shtml. Updated 24.11.2008. Referred 26.4.2011.

Musciano, Chuck – Kennedy, Bill 1997. HTML: The Definitive Guide. Second Edition. The United States of America. O'Reilly. WWW-document. http://www.ithov.com/book/website/basehtml/37721.shtml. Updated 3.5.2010. Referred 26.4.2011.

MySQL. WWW-document. http://en.wikipedia.org/wiki/MySQL. Updated 26.4.2011. Referred 29.4.2011.

NetBeans. WWW-document. http://en.wikipedia.org/wiki/NetBeans. Updated 29.4.2011. Referred 4.5.2011a.

NetBeans. WWW-document. http://zh.wikipedia.org/wiki/NetBeans. Updated 29.4.2011. Referred 4.5.2011b.

NetBeans IDE 7.0 Download. WWW-document. http://netbeans.org/downloads/index.html. No update information available. Referred 4.5.2011.

Newman, Chris 2005. SAMS Teach Yourself PHP in 10 Minutes. The United States of America. Sams Publishing. WWW-document. http://www.ithov.com/book/website/php/22681.shtml. Updated 19.9.2007. Referred 28.4.2011.

PHP. WWW-document. http://en.wikipedia.org/wiki/PHP. Updated 27.4.2011. Referred 28.4.2011.

Scripting language. WWW-document. http://en.wikipedia.org/wiki/Scripting_language. Updated 21.4.2011. Referred 28.4.2011.

Style sheet (web development). WWW-document. http://en.wikipedia.org/wiki/Style_sheet_(web_development). Updated 26.4.2011. Referred 26.4.2011.

WAMP. WWW-document. http://en.wikipedia.org/wiki/WAMP. Updated 3.5.2011. Referred 4.5.2011.

WampServer Presentation. WWW-document. http://www.wampserver.com/en/presentation.php. No update information available. Referred 4.5.2011.