

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI[®]

CINDI Book Bag System: Design and Implementation

Zhenjia Bradley Zhao

A Major Report

in

Department

of

Computer Science

**Presented in Partial Fulfillment of the Requirements
For the Degree of Master of Computer Science
Concordia University
Montreal, Quebec, Canada**

August 2001

© Zhenjia Bradley Zhao, 2001



**National Library
of Canada**

**Acquisitions and
Bibliographic Services**

**395 Wellington Street
Ottawa ON K1A 0N4
Canada**

**Bibliothèque nationale
du Canada**

**Acquisitions et
services bibliographiques**

**395, rue Wellington
Ottawa ON K1A 0N4
Canada**

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-64091-4

Canada

Abstract

CINDI Book Bag System: Design and Implementation

Zhenjia Bradley Zhao

This report describes the design and implementation issues of a book bag shopping system for the CINDI virtual library.

There are a number of book bag systems, most of which are developed using Cgi/Perl, Java, Active Server Page (ASP) and ColdFusion. In the design and implementation of the CINDI book bag system, we use the PHP as a scripting language, MySQL as a database server, and Apache as a Web server under Debian Linux. This system environment can meet the requirements of the CINDI virtual library. It is a good combination for developing the CINDI book bag system.

For introduction and easy understanding of the book bag system, the shopping bag system of the online Chapters.ca bookstore is reviewed.

Some important issues for book bag system are discussed, like maintaining persistent information on the stateless web, user authentication, security, and the principles of web GUI design. What we applied in the system is also stated.

The components of the book bag system and its implementation are given. Some screens of the GUI interface pages and some related code are also presented.

Finally, the integration of the book bag system in CINDI virtual library system and further works for the system are depicted.

Acknowledgement

I'd like to take this opportunity to express my deepest appreciation to my supervisor, Professor Bipin C. Desai, for not only his constant guidance and encouragement, but also for his technical support. He spent a lot of spare time and helped me to fulfil this project, especially when he is on sabbatical. Thanks for his patience and kindness.

I am grateful to the professors and staffs in Computer Sciences department at Concordia for the wonderful courses and services. I would especially like to say thanks to Ms. Halina Monkiewicz, who was always friendly and prompt in providing assistance. Also I would say thanks to the security staffs both in downtown and in Loyola campus.

Thanks also should reach to my good friends, such as Mr Lei Yaohui, for giving me encouragement, advice and assistance.

Finally, I have to say thanks to my wife, Meimin Liu for her understanding and continuous encouragement and support. I should say sorry to my lovely son and small daughter, for not being able to get enough time to company them and guide them, especially during the weekends and holidays.

Contents

1.	Introduction to CINDI System and Book Bag	1
2.	Requirements for CINDI Book Bag System Development	4
2.1	Book Bag/Shopping Cart Review	4
2.2	Requirements for CINDI Book Bag System	9
3.	CINDI Book Bag Design	11
3.1	Open Source Software – Development Tools and Environment for CINDI Book Bag System	11
3.1.1	PHP – Scripting Language	12
3.1.2	MySQL – Back End Database Server	15
3.1.3	Apache – Web Server	16
3.1.4	Linux – Operating System	17
3.2	Maintaining Persistent Information on the Stateless Web	19
3.2.1	Using Cookie	20
3.2.2	Using Query String	22
3.2.3	Using IP Address	23
3.2.4	Using Combination of Query String with IP Address and Time .	24
3.3	Security and User Authentication	25
3.3.1	Security	25
3.3.2	User Authentication	26
3.4	Web User Interface Design Principles	30
4.	CINDI Book Bag Implementation	32
4.1	Book Bag Components Overview	32
4.2	Database Scheme	32
4.3	The Catalog of the Site	38
4.4	The Book Bag Status	43
4.5	The Checkout Counter	45
4.6	Site Administration	48
4.7	Error Prevention and Message Prompting	58
5.	Integration of Book Bag in the CINDI System	60
6.	Discussion and Further Work	63

6.1	Maintaining State with PHP Session	63
6.2	Using User Id and Password to Authenticate the General Users	64
7.	Conclusion	66
8.	Appendix : Example Code	67
9.	References	87

1. Introduction to CINDI and Book Bag System

The CINDI (Concordia INDEXing and DIScovery system)[4] is a system initiated by Prof. B. C. Desai. CINDI enables any resource provider to catalog his/her own resource and any user to search for hypermedia documents using a search criteria, such as author, title, subject etc. The system offers a bibliographic database that includes information about documents available on the Internet. A standardized index scheme is used to ensure homogeneity of the syntax and semantics of such an index. These index entries are stored in the database system (Semantic Header Database System). The system is based on a set of interconnected Internet nodes, each node has a graphical user interface. The CINDI system is characterized by the following:

- A standardized metadata to describe each information resource – the Semantic- Header.
- A Semantic-Header registration system to register the Semantic-Header into the Semantic Header Database System.
- A search system that allows query entry for information discovery.
- An annotation mechanism.
- A Semantic-Header Database that stores the Semantic-Headers.
- A Catalog Database that stores information about subject classification using a standard cataloguing scheme.

Using graphical user interfaces, the user of the CINDI virtual library can register/update an index entry, make annotations to any index entry, and executes search.

There are many advantages of the CINDI system compared to other similar systems:

- The Semantic-Header allows the indexation of documents accessible by any protocol.
- The Semantic-Header includes annotations of reviewers and other users thus offering the possibility of a more informed decision as to pertinence of the source resource.
- The Semantic-Header syntax offers a way to register standardised keywords chosen by the provider of the resource. These make searching uniform. Other systems often hack terms from title and/or content bringing about unpredictable results.
- The Semantic-Header is designed to be part of each resource. The HTML/SGML based syntax allows its display by appropriate browsers like Internet Explorer and Netscape.
- The distributed nature of the Semantic-Header database provides for scalability.
- In other indexing systems, one of the limitations is the low numbers of indexed documents. This is illustrated by the disappointing results for manual index systems. The difficulty lies in convincing people to register information regarding their resources. CINDI provides an easy-to-use interface to register metadata of resources, thereby assuring the presence of metadata in resources by browsers.
- Since the registration of the Semantic-Header in the database is performed by the provider of the resource, it is economic, accurate and efficient.

Book bag (also called shopping cart) is a software component that can be used in an online digital content sale. Online book bag works like a shopping cart in a supermarket. The only difference is that the users purchases are electronic resource

during online shopping. The basic functionality of the book bag can be explained as follows. If a user visits the online bookstore, he/she can search for a book by title, author, or a list of keywords. The database of the bookstore will provide the user with a list of books, according to the searching criteria. By clicking on the button a book he/she wants to purchase will add it to the book bag. The user may continue to add other books to the bag. The content of the bag can also be displayed. The user may remove a book from the bag. Finally, when the user is ready to checkout the bookstore and would like to place an order, all books inside the bag will be displayed. If the user decides to make a purchase, the order information (the user's name, address, phone number, ordered book titles, credit card number, etc.) will be sent back to the database of the bookstore. After this order is processed, the purchased items will be shipped to the user.

From the above description, we can see that the book bag provides a place where the informations about items that interest a customer are stored temporarily. Thus, the book bag should have the basic capability to add, delete, and retrieve pieces of the information. This information includes the unit price, unit quantity, item name, etc.

CINDI system holds digital documents and other resources provided by authors. Some of them have their own copyrights and prices. When users want any of these resources, they should pay for it. So a book bag component for shopping is needed.

This report describes the design and implementation issues of a book bag shopping system for the CINDI virtual library.

For introduction and easy understanding of the book bag system, the shopping bag system of the online Chapters.ca bookstore is reviewed.

Some important issues for book bag system are discussed, like maintaining persistent information on the stateless web, user authentication, security, and the principles of web GUI design. What we applied in the system is also stated.

The components of the book bag system and its implementation are given. Some screens of the GUI interface pages and some related code are also presented.

Finally, the integration of the Book Bag system in CINDI virtual library system and further works for the system are depicted.

2. Requirements for CINDI Book Bag System Development

2.1 Book Bag/Shopping Cart Review

The explosive growth of information technology has changed forever the way we live, work, and do business. We can also see the same explosive growth in the rapidly growing, worldwide development and application of Internet. A big portion of the huge web sites is commerce-oriented like online supermarkets and bookstores. These online businesses need a system named book bag, shopping bag, or shopping cart to allow the customers to hold their selected items.

A lot of virtual book bag systems have been used by the web sites. They are developed with different kinds of software, and run on various servers and database servers under various platforms, for their different application orientations. For example, Easy Cart [16] uses CGI as the scripting language; Eastland Hybird [17] uses CGI and Java as the developing tool; Eastland ColdFusion [19] uses ColdFusion as the scripting language.

Although different kinds of virtual book bag systems have many common functions, each of them has its own features and functionalities. Basically, the book bag system includes attributes such as item number, quantity, item description, price, etc, as well as command buttons such as add, delete and modify.

At first, let us look at how such a system works using a web site with book bag. We take the popular bookstore site Chapters.ca [27] as an example to explore it in detail.

Shopping with Chapters.ca is made with the Shopping Bag feature. New users can browse the Chapters.ca without any registration or providing any personal information,

until he/she finds an interesting item and has the intention to put it in a wish list or a shopping bag.

When the user finds an interesting item in the various categories of Chapters.ca, he/she can click on the icon or title of the item, and get the detailed comments, author, review, price, and shipment information. Then the user has three choices. He/she can add the item to Shopping Bag if he/she intends to buy it, or he/she can add this item to the Wish List if he/she is just interested in this item but does not decide to buy it or not. The third choice is not to do anything about it.

With Shopping Bag system, the user can put his/her items in the bag, and then proceed to the checkout. With each item listed on Chapters.ca, there is a box called "Quantity" The user can enter the number of items he/she likes to buy here, then click on "Add To Shopping Bag" button. He/she will go to the "Shopping Bag" page. He/she can also reach this point by clicking on the "Shopping Bag" button on the top of any page if the user has already registered.

To add more items to his/her Shopping Bag or Wish List, the user can click on "Continue Shopping" button. To remove an item from the bag, the user can click on "Remove" box to the right of it, then click on "Update." The item will then be removed from his/her Shopping Bag. To change the quantity of any item in the Shopping Bag, the user can enter the desired number in the Quantity ("Qty.") box, then click on "Update". The new quantity and subtotal should then be displayed.

Once the items the user likes to purchase are in his/her Shopping Bag, he/she can click on "Checkout." he/she will then be prompted to enter his/her e-mail address and

password. If he/she doesn't have an account, he/she will be asked to "Create an Account".

On Create a New Account page, the user will be asked to enter an email address and password of his/her choice. These two pieces of information will be used together to identify the user during his/her future visits to Chapters.ca. After he/she submits the information, he/she will be asked to enter some further information about himself in the "Your Settings" section.

In the "Your Settings" section, the user can either enter his/her personal information (if he/she is a new customer) or edit existing information. In the Form sections, the user will be asked to enter his/her personal information, billing information, credit card information, and shipping address and to choose a shipping method.

After the user signs in as a registered customer or creates his/her account as a new customer, a summary of his/her order -- along with his/her billing, credit card and shipping information -- will displayed. Then the user can click on "Purchase" to enter his/her complete order for immediate processing.

The user can click on "Edit" to go to "Your Settings", where he/she can change his/her billing, shipping and credit card information. After he/she has modified the settings, he/she can click on "Submit" to complete the change. The user will return to the Order Summary page, where he/she can click on "Purchase" button to complete his/her order.

Chapters.ca uses cookies (small files stored in the user's computer) in order to keep track of what the user put in his/her Shopping Bag -- but only if the user's browser is configured to accept them. This information is saved from visit to visit. When the user

shops at Chapters.ca, a small file - a "cookie" - is stored in his/her computer. The cookie contains a unique personal identification code and is used to:

- To save his/her Shopping Bag and retrieve it next time when he/she visits the site.
- To personalize his/her experience by alerting him to features and information Chapters.ca thinks would be of interest to the user.

If the user arrives at Chapters.ca with his/her Cookies turned off, he/she won't be identified until he/she registers and signs-in.

Wish List is a feature of Chapters.ca online shopping system where the user can keep track of items that he/she is interested in and let others know what gifts he/she would like to receive. To add items to his/her Wish List, the user can simply browse Chapters.ca as usual, and click on the "Add to Wish List" button on any item page. He/she will be prompted to log in and then will be brought to his/her list. If the user wants to manage an existing list, click on "Your Wish List" link in "Your Settings."

In Wish List page, the items the user is currently interested in will be displayed, along with information about price and quantity. If he/she likes to receive more than one quantity of an item, he/she can do so in this page by changing the quantity field and clicking on "Update" button.

In Wish List page, the user can move items between his/her Wish List and his/her Shopping Bag using "Move to Shopping Bag" and "Move to Wish List" buttons. After the user has moved the items into Shopping Bag, click on "Shopping Bag" button to go to

the normal purchase process. To remove an item completely, select the checkbox in the "Remove" column for that item, then click on "Update" button.

To invite others to see his/her Wish List, the user can click on "Invite Others" button. A form will be displayed, where he/she can fill in up to five email addresses at a time. To allow others who purchase items from his/her List to send items directly to his/her home address, select an address at the bottom of his/her Wish List page. This is optional.

2.2 Requirements for CINDI Book Bag System

For the CINDI Book Bag, there are two kinds of users. One is the end user, who browses and buys items through his/her browser, and another kind is the administrator who manages the CINDI virtual library system.

After analysis of the specialties of the CINDI system, we get the following end user requirements for the CINDI Book Bag:

- Users should be able to use the Book Bag from the major web browsers like Internet Explorer and Netscape.
- Users will be requested to provide their personal information only when it is really needed.
- User should be uniquely identified by a certain mechanism.
- The transaction of the user should be secure. That is, some basic security mechanism will be built into the book bag system, for preventing unauthorized persons to access the information provided by users.
- Users should be able to buy CINDI Library items through the use of a book bag.

- Users will be able to choose and add items to their bag, and decide later whether they will buy the selected items or not.
- Users should be able to change items from their bag, before checking out.
- After the user has checked out and the payment has been processed, all the selected items of the order should be shipped to the user by the ways designed by CINDI virtual library.

The administrator, who manages the CINDI system site, has specific requirements of his/her own:

- Administrator should be able to manage the book bags using his web browser.
- Administrator should be able to add/delete/update the category of the library.
- Administrator should be able to add/delete/update the subject of the library.
- Administrator should be able to add/delete/update the sub-subject of the library.
- Administrator should be able to add/delete/update the item of the library.
- Administrator should be able to change the status of the items purchased by the user, after the user has been given a code to access the purchased item.
- Administrator should be able to review the orders.

3. CINDI Book Bag System Design

3.1 Open Source Software – Development Tools and Environment for CINDI Book Bag System

Open source [10] promotes software reliability and quality by supporting independent peer review and rapid evolution of source code. The basic idea behind open source is very simple. When programmers can read, redistribute, and modify the source code for a piece of software, the software evolves. People improve it, people adapt it, and people fix bugs.

Open source software is an idea whose time has finally come. For twenty years it has been building momentum in the technical cultures that built the Internet and the World Wide Web. Now it's breaking out into the commercial world.

It's hard to know how many people are using open-source software because it is mostly spread via free downloads off the Internet, and nobody has anything like total sales figures.

Linux is believed to have somewhere between 4 and 27 million users, with best estimates towards the upper end of that range. (According to IDG, business Linux usage increased 212% in 1998. Other figures indicate it is roughly doubling yearly.)

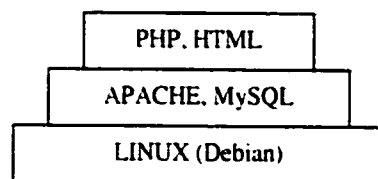
The Netcraft web server survey[23] tallies which web servers are used on the Internet. It consistently shows the open-source Apache web server to have over 50% and

steadily increasing market share, beating out better-hyped proprietary products like Netscape's and Microsoft's server suites.

The Internet Operating System Counter[28] collects data about operating system usage on the Internet in Europe. It consistently shows Linux is the most popular Internet-connected operating system there.

Open source has brought a lot more than Linux[8] to the computing world. It has also given us PHP (Hypertext Preprocessor) [5], APACH[7] and MySQL [6] database. The combination of PHP with MySQL is one of the best solutions for data-driven Web sites application.

The CINDI Book Bag system uses all the open source software as its development tools and environment: from the bottom to the top, Linux – Apache – PHP with the MySQL as DBMS.



3.1.1 PHP – Scripting Language

There are some kinds of web scripting languages, and the popular ones are ASP, PHP, Cold Fusion, and Perl. Each of them has special features of its own, advantages and disadvantages [5].

ASP is not really a language in itself. It is an acronym for Active Server Pages, the actual language used to program ASP with is a script version of Visual Basic. The

biggest drawback of ASP is that it's a proprietary system that is natively used only on Microsoft Internet Information Server (IIS). This limits its availability to Win32 based servers. There are a couple of projects in the works that allow ASP to run in other environments and web servers. ASP is said to be a slower and more cumbersome language than PHP. One of the pros of ASP is that since it uses VBScript, it's relatively easier to pick up the language if you've already known how to program in Visual Basic. ASP support is also enabled by default in the IIS server, making it easy to get up and running.

PHP is commonly said to be faster and more efficient for complex programming tasks and trying out new ideas[5]. PHP is generally referred to as more stable and less resource intensive as well. Cold Fusion has better error handling, database abstraction and date parsing although database abstraction is being addressed in PHP 4. Another thing that is listed as one of Cold Fusion's strengths is its excellent search engine, but it has been mentioned that a search engine is not something that should be included in a web scripting language. PHP runs on almost every platform there is; Cold Fusion is only available on Win32, Solaris, Linux and HP/UX. Cold Fusion has a better IDE and is generally easier to get started with, whereas PHP initially requires more programming knowledge.

The largest advantage of PHP over Perl is that PHP was designed for scripting for the web where Perl was designed to do a lot more and can, because of this, get very complicated. The flexibility / complexity of Perl makes it easier to write code that another author / coder has a hard time reading. PHP has a less confusing and stricter format without losing flexibility. PHP is easier to integrate into existing HTML than Perl.

PHP has pretty much all the 'good' functionality of Perl: constructs, syntax and so on, without making it as complicated as Perl can be. Perl is a very tried and true language. It's been around since the late eighties, but PHP is maturing very quickly.

PHP (recursive acronym for PHP: Hypertext Preprocessor)[5] is an open-source server-side scripting language (freely downloadable from php.net and zend.com) for creating dynamic Web pages for e-commerce and other Web applications. A dynamic Web page is a page that interacts with the user, so that each user visiting the page sees customized information. Dynamic Web applications are prevalent in commercial (e-commerce) sites, where the content displayed is generated from information accessed from a database or other external source.

PHP offers a simple and universal solution for easy-to-program dynamic Web pages. The intuitive interface allows programmers to embed PHP commands right in the HTML page. Its elegant design makes PHP significantly easier to maintain and update than comparable scripts in other languages.

PHP has been in development since 1994. PHP 3, released in June 1998, has gained rapid popularity, and is now used in Web-related applications by some of the prominent organizations such as Mitsubishi, Redhat, MP3-Lycos, Ericsson and NASA. Netcraft survey [23] shows that PHP usage has jumped from 7,500 hosts in June 1998 to 410,000 in March 1999, and further to 6,156,321 domains and 914,146 IP addresses in April 2001.

Unlike other scripting languages for Web page development, PHP offers excellent connectivity to most of the common databases (including Oracle, Sybase, MySQL, ODBC and many others). Perhaps the greatest advantage of PHP, when compared to

other scripting languages such as ASP (Active Server Page, developed by Microsoft) or ColdFusion (developed by Allaire Corporation), is that it is open-source and cross-platform, suitable for today's heterogeneous network environments.

PHP is the natural choice for developers on Linux machines running Apache server software. PHP runs fastest when embedded in Apache. PHP offers excellent code performance.

The PHP-Apache-Linux combination makes the site provide good performance, even though there are large number of simultaneous users and large number of items stored in the site[5].

3.1.2 MySQL – Back End Database Server

MySQL[6] is a compact database server. In addition to supporting standard SQL, it compiles on a number of platforms and has multithreading abilities on Unix servers, which make for great performance. For non-Unix people, MySQL can be run as a service on Windows NT and as a normal process in Windows 95/98 machines.

MySQL is a relational database management system, which stores data in separate tables rather than putting all the data in one big storeroom. This adds speed and flexibility. The tables are linked by defined relations making it possible to combine data from several tables on request.

MySQL is Open Source Software, which means that it's possible to use and modify for anyone. Anybody can download MySQL from the Internet and use it free. Anybody so inclined can study the source code and change it to fit their needs.

MySQL is very fast, reliable and easy to use database server [6]. MySQL was originally developed to handle very large databases much faster than existing solutions and has been successfully used in highly demanding production environments for several years[6]. Though under constant development, MySQL today offers a rich and very useful set of functions. The connectivity, speed and security make MySQL highly suited for accessing databases on the Internet.

MySQL is a client/server system that consists of a multi-threaded SQL server that supports different backends, several different client programs and libraries, administrative tools and a programming interface.

3.1.3 Apache – Web Server

Apache [7] server named as a plug in words “A PAtCHy server” has been shown to be substantially faster, more stable, and more feature-full than many other web servers [7]. Although certain commercial servers have claimed to surpass Apache's speed, we feel that it is better to have a mostly-fast free server than an extremely fast server that costs thousands of dollars. Apache runs on sites that get millions of hits per day, and they have experienced no performance difficulty [7].

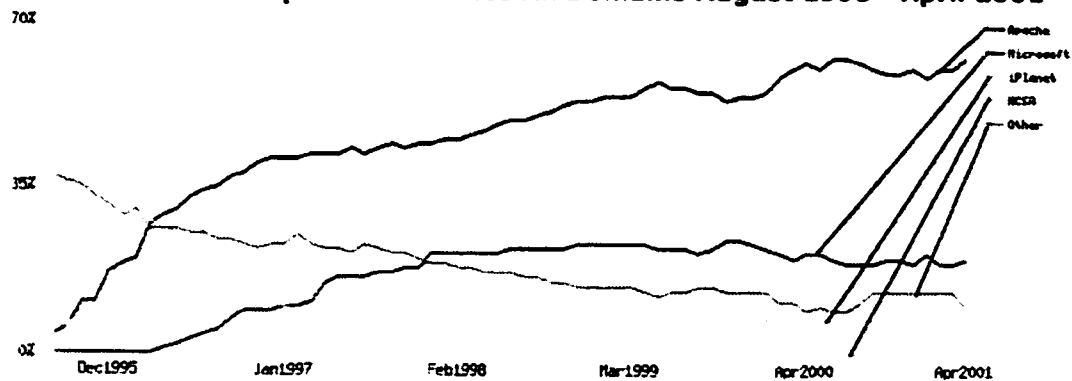
The advantages of the Apache web server:

- It is a powerful, flexible, HTTP/1.1 compliant web server
- It implements the latest protocols, including HTTP/1.1
- It is highly configurable and extensible with third-party modules
- It can be customized by writing 'modules' using the Apache module API

- It provides full source code and comes with an unrestrictive license
- It runs on Windows NT/9x, Netware 5.x, OS/2, and most versions of Unix, as well as several other operating systems
- It is being actively developed

In the April 2001 Netcraft survey[23], Netcraft received responses from 28,669,939 sites.

Market Share for Top Servers Across All Domains August 1995 - April 2001



Top Servers						
Server	March 2001	Percent	April 2001	Percent	Change	
Apache	17238004	60.25	17932251	62.55	2.30	
Microsoft-IIS	5648960	19.74	5916724	20.64	0.90	
Netscape-Enterprise	1750429	6.12	1762872	6.15	0.03	
Zeus	738068	2.58	779209	2.72	0.14	
Rapidsite	395094	1.38	402829	1.41	0.03	
thttpd	365445	1.28	369930	1.29	0.01	
AOLserver	262650	0.92	272815	0.95	0.03	
tigershark	182481	0.64	200620	0.70	0.06	
WebSitePro	141737	0.50	119586	0.42	-0.08	
ConcentricHost-Ashurbanipal	103249	0.36	106443	0.37	0.01	

3.1.4 Linux -- Operating System

Linux [8], the most popular Unix variant for PCs, is the only operating system not from Microsoft that is gaining significant market share. Created by Linus Torvalds and developed collaboratively over the Internet, Linux is the great success story of Open Source software development. Millions of users worldwide use Linux in a wide variety of

environments. Linux is used as a mainstay server operating system by many web sites and Internet service providers (ISP).

Linux is a modern operating system. This means that it runs on 32-bit architecture, uses preemptive multitasking, protects memory, supports multiple users, and has rich support for networking, including TCP/IP networking. Linux runs all the applications a Unix server system should run, including web servers like Apache, and database servers like Oracle, Informix, or more open applications like MySQL.

Linux may be the best example, but it is only one example of the power of open source. Collaborative, networked development is a new model of software development made possible by the Internet. Open source accelerates the development process. It breaks down the barriers between developers and users, and removes obstacles in developer-to-developer communication. In today's Internet economy, these kinds of accelerators have become not just a business advantage, but a business necessity. The proof of the open source model is in the results: Apache holds roughly 60% market share among web servers, and that market share is growing. Linux is the fastest growing server-class operating system [8].

CINDI system is an academic research project, in addition to the advantages of the open source combination of PHP-APACHE-MySQL-LINUX. Funding is also a main factor when we choose the language and environment for the project. Open source free software gives the chance to do some leading-edge academic research with little financial support.

Incidentally, we didn't find any existing Book Bag written with PHP scripting language. Hence it is a challenge to develop a Book Bag system using PHP.

The following software are used in the development:

- PHP version 3.16
- MySQL version 3.22.16b
- APACHE version 1.3.4
- Debian, a distribution of LINUX, which is easily upgradable.

3.2 Maintaining Persistent Information on the Stateless Web

Web is based on a client-server model. In the client-server model, the client opens up a channel of communication with the server and requests a resource. The server receives the request, locates the resource being requested, and sends it to the client, closing the channel of communication. This is an impersonal transaction between the client and server. The server does not keep open its channel of communication with the client. The server is not concerned with whom it has talked to recently, what it has sent recently, or what it thinks a particular client will request next. The server does one job, and it does it well: wait for a request from the client and then process that request. Due to this impersonal communication between a Web browser and a Web server, user information is not persisted from one request to another.

Now, as long as the user is aimlessly surfing from one site to another, this works without a problem. But what if he/she decides to buy a few books from Chapters.ca? In a "stateless environment", it would be very difficult to keep track of all the items he/she has selected for purchase, as the stateless nature of the HTTP protocol would make it impossible to keep track of the items selected. Because the Web client-server model does not make maintaining state inherently easy, we must examine some advanced techniques

to maintain state, something that allows client connections to be tracked and connection-specific data to be maintained. This can be archived using one of following: "cookies", "query string", and "IP address".

3.2.1 Using Cookies

A cookie is a small amount of data stored by the user's browser in compliance with a request from a server or script. A host can request that up to 20 cookies be stored by a user's browser. Each cookie consists of a name, value, and expiry date, as well as host and path information. An individual cookie is limited to 4KB [34].

To maintain state using cookies, we only need to write the cookie to the client's computer: when the user enters the information that we want to store. Cookies can persist on the client's computer for a variable amount of time. When writing a cookie to the client's computer, we can set when the cookie expires. Cookies allow a user's state to be maintained beyond the current visit. This is the expiry time of the cookie, which by default is to the end of the current session.

Cookies are based on a two-stage process[30]. In the first stage, the cookie is stored in the user's computer without their consent or knowledge. During the second stage, the cookie is clandestinely and automatically transferred from the user's machine to a Web server.

Because cookie only contains text it cannot transmit a virus or do damage to the user system. Due to their use in tracking users and some security problems surrounding them, cookies have been a concern to much privacy and security advocates.

"Cookie Worries Are Unfounded", U.S. Government Says[32]. A U.S. government agency has advised Internet users that the use of cookies -- the popular

technique for tracking website visitors -- does not compromise the privacy of users or the security of their computers.

Cookies are widely used by advertisers and website owners to monitor users' online activity. Some common uses for Internet cookies are[33]:

- An anonymous code given to users so the website operator can see how many users return at a later time. These cookies are configured to stay on the user's system for months or years and are called "persistent" cookies.
- A code identifying the user. This usually occurs after a registration. The site could keep a detailed account of pages visited, items purchased, etc., and even combine the information with information from other sources once they know who the user is.
- A list of items the user/buyer purchased. This is an often-used in "shopping cart" website to keep track of the user/buyer order. Often, cookies of this type 'expire' as soon as the user logs out or after a short time. These are called "session" cookies.
- Personal preferences. This can be anonymous or linked to personal information provided during a registration.

The "Internet Cookies Report"[31], conducted by The Energy Department's Computer Incident Advisory Capability (CIAC), stated that cookies are used by Internet shopping sites to keep track of the user and his/her shopping cart. When a user first visits an Internet shopping site, he/she is sent a cookie containing the name (ID number) of a shopping cart. Each time he/she selects an item to purchase, that item is added to the

shopping cart. When he/she is done with his/her shopping, the checkout page lists all the items in the shopping cart tied to that Cookie. Without Cookies, users would have to keep track of all the items they want to buy and type them into the checkout page or buy each item, one at a time.

The user can prevent any cookies from being sent to his/her system using the browser options. The user will need to turn Cookies back on if he/she wants to use any online services that require Cookies. The user can also choose the option to prompt them before accepting them, but at many sites he/she will be required to continually close the Cookie warning dialog box.

The drawback to Cookie methods is that if the user has cookies disabled, the Web site will appear to be stateless, and Cookies may not work for people behind some firewalls or proxies. Although the majority of Web surfers have Cookies enabled, if it is essential that the site maintain state for all the visitors, cookies variables just won't do.

Although there are pitfalls for using Cookie, it is still one of the most frequently used ways to track users on the Internet, as many commercial sites did [26]. In our first approach, when the user makes the selection of the CINDI library items, CINDI Book Bag system tracks users by asking them to log on using the bag name and password and storing them in Cookies for maintaining information. Because the CINDI system decides not to use Cookie in the system, then we discard all the cookie-related scripts and study other ways to maintain information.

3.2.2 Using Query String

The great drawback of the Cookie is its dependence on the client. Not only the user may choose not to allow Cookie to be sent, it must also rely on the browser's

implementation of the standard. Some browsers have documented bugs concerning the way that they deal with Cookie. When the website only want to save state for a single session, we can use a more traditional approach.

When a form is submitted by using GET method, its fields and values are URL encoded and appended to the URL to which the form is sent. They then become available to the server and to the scripts. Assuming a form with two fields, `user_id` and `password`, the query string should end up looking something like the following:

```
http://dumbo.concordia.ca/test5.php?user_id=mattzandstra&password=HyugT456
```

Each name and value is separated by an equals (=) sign, and name/value pair is separated by an ampersand (&). PHP decodes this string and makes each of the pairs available in the `$HTTP_GET_VARS` associative array variable. It also creates a global variable for each name, populating with the corresponding value. So, to access the `user_id` GET variable, we can use either of the following variables:

```
$HTTP_GET_VARS[user_id];
```

```
$user_id;
```

Query string is not limited to using forms to send strings. We can easily build string and pass substantial amounts of information from page to page.

3.2.3 Using IP Address

When the cookie is not working on the user's browser, some systems to use Internet (IP) address as the unique identifier of the user. Unfortunately there are some side effects by using IP address solely. Most users on the Internet are given their addresses out of a pool, so a user might have one address the first time he/she visits the

site and then a different one the next time. Another user may get that same address and come to shop. The second problem involves proxy servers. Proxy servers are used for security, address sharing, or page caching. Because of this, one single IP address is shared by a group of people. IP addresses do have their uses, but they should have some limitations.

3.2.4 Using Combination of Query String with IP Address and Time

The final CINDI Book Bag system uses the combination of the query string and IP address, date, time to maintain the persistent information on the stateless Web. Users will be distinguished and identified by the IP address, date and time when he/she enters the CINDI library. The user identification information is maintained using query string.

Because Book Bag system is used in an online shopping system, it is reasonable to assume that the user makes and finishes his/her shopping during a single session and hence using the same IP address. When the same user comes to another IP address (machine), it will be treated as another shopping cycle. We also assume that there is only one user in shopping on an IP address in a certain second. If there are more than one user using the same IP address in the same second for shopping on the CINDI virtual library, one user can keep his/her own selections and remove others when he/she decide to make an order. This kind of case should very rarely happen. At this point, it leads the user to the check out, and the user becomes a customer (we name it buyer in Book Bag system) of the web-shopping site. Since then the IP address is no longer used for maintaining the information, instead we use the database on our server to store and maintain customer information well as the order.

3.3 Security and User Authentication

3.3.1 Security

Security is an important part of a web site [1][35]. Without strong server authentication, users of a public Web server will not be able to determine if they have connected to an authentic Web server or a bogus version operated by an intruder. In such cases, the user may receive false information and act upon it based on the assumption that it originated from your Web server. The user may also provide sensitive information such as a credit card number.

Encryption can be used to protect information traversing the connection between a Web browser client and a public Web server. Without encryption, anyone with access to network traffic can determine, and possibly alter, the content of sensitive and restricted information, even if the user accessing the information has been carefully authenticated. This may violate the confidentiality and integrity of critical information.

Using secure server software - Secure Socket Layer (SSL) [1], is a great way of improving the security of the web site without having to change one single line of code. What SSL does is to use cryptograph to protect the flow of information between the Web server and the browser. Not only does SSL encrypt all the data flowing over the Internet, it also provides the means for both parties to authenticate each other. Through this way, the user can buy products online without his/her credit card information and other personal information being seen by a third party.

SSL uses an encryption technique called key cryptography, where the server end of the connection sends the client a public key for encrypting information, which only the server can decrypt with the private key it holds. The client uses the public key to encrypt

and send the server its own key, identifying it uniquely to the server and preventing onlookers at points between the two systems from mimicking either server or client.

These characteristics make SSL very well suited for use in the CINDI system, as well as other commercial sites, where sensitive information needs to be exchanged. Like online Amazon bookstore [21] declared on its site: "Safe Technology: Our secure server software (SSL) is the industry standard and among the best software available today for secure commerce transactions. It encrypts all of the personal information including credit card number, name, and address, so that it cannot be read as the information travels over the Internet."

3.3.2 User Authentication

On a web site we sometimes need to uniquely identify a user. Without strong user authentication, we will not be able to restrict access to specific information by authorized users. All information that resides on the public Web server will be accessible by anyone who accesses that server.

Basic HTTP authentication uses a challenge/response scheme to authenticate users attempting to access a password-protected page. The challenge process begins when the user requests a file from the web server. If the file is within a protected area, the server responds by sending out a 401(unauthorised user) string in the header of the response. The browser detects that response and pops up the username/password dialog box; once these fields are filled in, the user clicks OK to send the information back to the server for authentication.

If the username and password pair is valid, the protected file will be displayed to the user. The validation will last for as long as the now-authenticated user is within the protected area. However, if the username and password typed into the dialog box cannot be authenticated, the dialog box will again be up popped, prompting the user to try again. This cycle will be repeated until the proper username and password combination is entered or the user gives up and leaves.

One user authentication method is through the Web server – Apache. We can simply let Apache take care of authenticating the user, which requires minimal effort on the side of the PHP programming [1]. The example code is like this:

```
AuthName    "Secret page"      # The realm
AuthType    Basic
# The password file has been placed on the web
AuthUserFile /home/cindi/website.pw
<Limit GET POST>
require    valid-user
</Limit>
```

We can put these directives in a file called .htaccess in the directory we try to protect the user from accessing. We also need to create a file with the username/password combinations. We can do this with the htpasswd program that comes with Apache. It is better to store the password file outside the Web tree and make sure that the permission on the password file only allows the owner to view and modify. Of course, the Web server must be able to read the password file too.

After that, when a user wants to access a file in the protected directory, the Web server asks the browser for a username and a password. The browser pops up a small input box where the user can type in his/her username and password. Then the server will authenticate the user information against the password file.

Another method is user authentication with PHP. A simple PHP script can mimic the HTTP authentication challenge/response system by sending the appropriate HTTP headers that cause the automatic display of the username/password login page. PHP assigns the username and password entered to the global variables `$PHP_AUTH_USER` and `$PHP_AUTH_PW`, respectively. We use these variables to validate input against a stored list of usernames/passwords, such as those found in a flat file, `.htpasswd` file, or database[25].

There are some ways to validate username/passwords:

a) Validate Username/Passwords Using Hard-Coded Values

Let us use hard-coded values "user" for the username and "open" for the password as an example. The code will echo a "Success!" message if both of these values are correctly entered in the dialog box. Otherwise, a "Failure!" message will be displayed.

b) Validate Username/Passwords Using a Flat File

Instead of using hard-coded values within the script, we can use a flat (text) file to authenticate the users. This file would consist of a list of usernames and passwords, such as: `joy:ff44hg`, `jason:fdh567`, `mei:h65uhjF`, `ted:g667yGff`, `michael:f45VVg`. By creating a loop, we can test the user and password pairs one by one, comparing the username read from the file to `$PHP_AUTH_USER`, and its corresponding password to `$PHP_AUTH_PW`. Determining authentication requires us to make a number of comparisons. We initially set a Boolean variable to false and iterate through the series of comparisons until a positive match can be found. If a match is found, the variable is set to true and the user is authenticated.

Otherwise, the variable *remains* false, and the browser is prompted to display an authentication dialog box.

c) Validate Username/Passwords Using a .htpasswd File

In the .htaccess-based authentication [34], the usernames and encrypted versions of user passwords are kept in a text file usually called .htpasswd. In general, passwords are encrypted using standard UNIX DES encryption, and stored in a format similar to the following: joy: WvzodahMR9Usk, mei:h65uhjF,

ted:g667yGff. The actual plaintext password for user joy is not "WvzodahMR9Usk", as shown here. Rather, his/her plaintext password is (let us say) "abba001", and "zodahMR9Usk" is the DES-encrypted version of that string.

d) Validate Username/Passwords Using a Database

Database user validation is to compare an authentication dialog box username/password pair with a list of username/password pairs residing in a database table. It uses PHP's MySQL connection functions. To exact a match for both a username and a password, the SQL statement could be:

```
SELECT *  
FROM users  
WHERE username=$PHP_AUTH_USER` and  
password=$PHP_AUTH_PW`
```

Then a positive result will be returned when the values for \$PHP_AUTH_USER and \$PHP_AUTH_PW both match exactly with their username and password field counterparts, as entered in the users table.

In the CINDI Book Bag system, we use hard-coded values at the beginning and then we change to use database to authenticate the administration user. We don't authenticate users for the Catalog pages that are public accessible.

3.4 Web User Interface Design Principles

User interface (UI) of a system is the component by which a system is judged. An interface, which is difficult to use, will result in a high level of user errors. It will cause the software system to be discarded, irrespective of its functionality.

The following principles for user interface design are followed in designing the CINDI Book Bag interfaces/pages:

- **Simplicity.** Use simple and natural dialog, present exactly the information the user needs in a natural and logical order, remove or hide irrelevant or rarely needed information.
- **User Familiarity.** Use words and concepts from the users' world, avoid the use of system technical terms, view interactions from the user's perspective, use meaningful icons and abbreviations.
- **Consistency.** Keep the consistency of effects, input, language and graphics.
- **User Guidance.** Inform the user about: where they are, where they have been, and where they can go. Inform the user what the system or application is doing, how it is interpreting the user's input.
- **Recoverability.** Include mechanisms to allow users to recover from their errors.
- **Prompting.** Deal with errors in a positive and helpful manner, provide meaningful error messages.
- **Prevention.** Prevent errors, try to make errors impossible, provide reasonable checks on input data.

With these principles in mind when we make the system design and implementation, users will be provided with a simple, user-friendly, easy-to-understand and easy-to-manipulate Web user interfaces/pages.

4. CINDI Book Bag Implementation

4.1 Book Bag Components Overview

There are some basic components to a Book Bag system. The first is the catalog. As we expect, a catalog lets the user view items, get information, compare prices, etc. It is where the user visits first and often. Some authors don't put the catalog as a part of the Book Bag system. Here, for easy and complete description, we simulate a catalog/subjects/items database and treat it as a component of the Book Bag system.

Next, we have the actual Book Bag (temporarily holding items). The bag works just like it sounds. It collects all the things that the user wants to buy. When the user sees something he/she likes, he/she can add the items to his/her Book Bag. If he/she change his/her minds, items can be removed.

When the user is ready to buy, he/she heads to the checkout counter. Here he/she can review all the items in his/her bag, make changes, and hand over his/her personal information including payment information.

Finally when the payment is successfully processed, the items the user bought will be given to him/her by sending special codes to him/her.

4.2 Database Scheme

We name the back end database as Bag, which mainly contains the following 10 tables:

- Buyers
- BagItems
- Category

- Subjects
- Subsubjects
- Items
- Orders
- Receipts
- Users
- Administrators

Here we have two tables to describe the people who use the CINDI virtual library.

The “User” table is used for keeping the surfer ID on the stateless Web. It automatically stores the user (surfer) ID and his/her surfing date and time. In the CINDI Book Bag system, we assume that the user will checkout during the session he/she put his/her selection in the book bag. The “Buyer” table is used for storing the information of the user who has become a customer.

The Buyers table contains information like name, email, phone, address, and payment information of the customer (each time the customer places an order, he/she is treated as a distinguishing buyer):

Column Name	Description
Name	Name of the user
Email_address	Customer's email address
DayPhone	Daytime phone number of the customer
PayMethod	Payment method (Credit card or check)
Card_type	Customer's credit card type, master/visa
Card_no	Customer's credit card number
Expiry_date	Expiry date of the credit card
Address_line1	Address line 1
City	City name
State	State name
Zip	Postal/Zip code
Country	Country name
EvePhone	Evening phone number of the customer
OrderTotal	Total amount of the customer's order
Date	Order date
OrderNumber	Foreign key from Orders table
UserID	Foreign key from Users table
BuyerID	Primary key

The BagItems table contains the description of the user's bag:

Column Name	Description
UserID	Foreign key from Users table
ItemID	Foreign key from Items table
ItemQuantity	Item quantity
Date	Bag item selection date
BagItemID	Primary key

The Category table contains the information about the first layer of the hierarchy in the Catalog of the CINDI virtual library:

Column Name	Description
Category	Category name
CategoryID	Primary key

The Subjects table contains the information about the middle layer of the hierarchy in the Catalog of the CINDI virtual library:

Column Name	Description
Subject	Subject name
Category	Category ID of the subject, foreign key from Category
Subject ID	Primary key

The Subsubjects table contains the information about the middle layer of the hierarchy in the Catalog of the CINDI virtual library:

Column Name	Description
Subsubject	Subsubject name
Subject	Subject ID of the subject, foreign key from Subject
Subsubject ID	Primary key

The Items table contains the detailed information about the root layer of the hierarchy in the Catalog of the CINDI virtual library:

Column Name	Description
Authors	Author names of the item
Title	Title of the item
Description	Short introduction of the item
Price	Price of the item
Subsubject	Subsubject ID of the item, foreign key from Subsubject table
ItemID	Primary key

The Orders table contains the order information:

Column Name	Description
BuyerID	Foreign key from Buyers table
UserID	Foreign key from Users table
Date	Order date
OrderTotal	Total amount of the order
OrderNumber	Primary key

The Receipts table contains the detailed information about the receipt of the buyer/customer's order:

Column Name	Description
BuyerID	Foreign key from Buyers table
UserID	Foreign key from Users table
Date	Order date
OrderNumber	Foreign key from Orders table
ItemID	Foreign key from Items table
TotalCost	Total amount of the order

The Users table contains the temporary information about the user of the CINDI virtual library:

Column Name	Description
User	User's ID
Date	Shopping date and time of the user

The Administrators table contains the information about the administrators of the CINDI virtual library:

Column Name	Description
Name	Administrator name
AdminID	Administrator login ID
Password	Administrator password
DayPhone	Daytime phone number of the Administrator
EvePhone	Evening phone number of the Administrator
Email	Email of the Administrator

The following code shows the SQL commands under MySQL for the creation of the database and tables:

Code: create_db_objects.sql

```
Create database Baq;

connect Baq;

CREATE TABLE Buyers (
Name VARCHAR(100) NOT NULL,
Address1 VARCHAR(100) NOT NULL,
Address2 VARCHAR(100) NOT NULL,
City VARCHAR(100) NOT NULL,
```

```

State CHAR(2) NOT NULL,
Zip VARCHAR(20) NOT NULL,
Country VARCHAR(20) NOT NULL,
Email VARCHAR(100) NOT NULL,
DayPhone VARCHAR(100) NOT NULL,
EvePhone VARCHAR(100) NOT NULL,
Contact INT(1) NOT NULL,
PayMethod VARCHAR(100) NOT NULL,
CCType VARCHAR(100) NOT NULL,
CCNum VARCHAR(100) NOT NULL,
CCExpire VARCHAR(100) NOT NULL,
OrderTotal DECIMAL(8,2) NOT NULL,
Date VARCHAR(100) NOT NULL,
OrderNumber BIGINT NOT NULL,
UserID VARCHAR(100) NOT NULL,
BuyerID BIGINT NOT NULL AUTO_INCREMENT,
PRIMARY KEY (BuyerID)
);

CREATE TABLE BagItems (
UserID VARCHAR(100) NOT NULL,
ItemID VARCHAR(25) NOT NULL,
ItemQuantity INT(4) NOT NULL,
Date VARCHAR(100) NOT NULL,
BagItemsID BIGINT NOT NULL AUTO_INCREMENT,
PRIMARY KEY (BagItemsID)
);

CREATE TABLE Category (
Category VARCHAR(100) NOT NULL,
CategoryID BIGINT NOT NULL AUTO_INCREMENT,
PRIMARY KEY (CategoryID)
);

CREATE TABLE Subjects (
Subject VARCHAR(100) NOT NULL,
Category BIGINT NOT NULL,
SubjectID BIGINT NOT NULL AUTO_INCREMENT,
PRIMARY KEY (SubjectID)
);

CREATE TABLE Subsubjects (
Subsubject VARCHAR(100) NOT NULL,
Subject BIGINT NOT NULL,
SubsubjectID BIGINT NOT NULL AUTO_INCREMENT,
PRIMARY KEY (SubsubjectID)
);

CREATE TABLE Items (
Authors VARCHAR(100) NOT NULL,
Title VARCHAR(100) NOT NULL,
Description MEDIUMTEXT NOT NULL,
Price DECIMAL(7,2) NOT NULL,
Subsubject BIGINT NOT NULL,
ItemID BIGINT NOT NULL AUTO_INCREMENT,
PRIMARY KEY (ItemID)
);

```

```

CREATE TABLE Orders (
  BuyerID BIGINT NOT NULL,
  UserID VARCHAR(100) NOT NULL,
  Date VARCHAR(100) NOT NULL,
  OrderTotal DECIMAL(8,2) NOT NULL,
  OrderNumber BIGINT NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (OrderNumber)
);

CREATE TABLE Receipts (
  BuyerID BIGINT NOT NULL,
  UserID VARCHAR(100) NOT NULL,
  OrderNumber BIGINT NOT NULL,
  Date VARCHAR(100) NOT NULL,
  ItemID BIGINT NOT NULL,
  ItemQuantity INT NOT NULL,
  TotalCost DECIMAL(8,2) NOT NULL
);

CREATE TABLE Users (
  User VARCHAR(100) NOT NULL,
  Date INT(40) NOT NULL
);

CREATE TABLE Administrators (
  Name VARCHAR(100) NOT NULL,
  AdminID VARCHAR(100) NOT NULL,
  Password VARCHAR(100) NOT NULL,
  DayPhone VARCHAR(100) NOT NULL,
  EvePhone VARCHAR(100) NOT NULL,
  Email VARCHAR(100) NOT NULL
);

```

4.3 The Catalog for the Site

A catalog is where users will spend the bulk of their time looking for something they want, and should help them find it as quickly as possible.

The first idea that comes is categorization. If we divide the items into categories, the user can quickly find the items of interest.

In CINDI Book Bag system, we simulate a hierarchy of the catalog for the CINDI virtual library. The top level of the hierarchy is Category, which is the top level of subject

classification based on the Library of Congress subject headings. The second level of the hierarchy is Subject, which is the child classification level for the general level Category. The third level of the hierarchy is Subsubject, which is the lower subject level for the general level Category and Subject level. The bottom level of the hierarchy is Items, which is the last level of the Catalog, and includes a short description about the items.

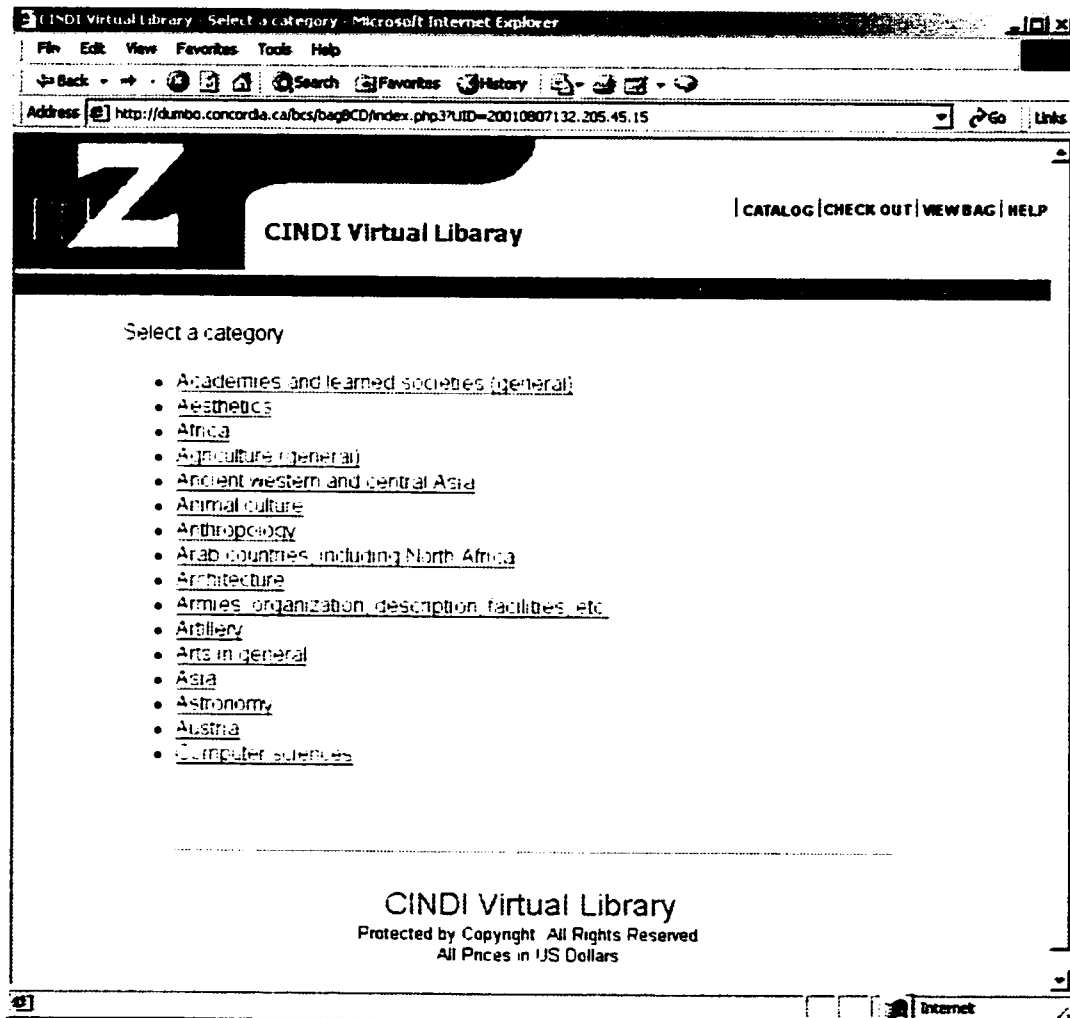


Figure 1: CINDI library category – top level of the catalog hierarchy

In Figure 1 to 4, a catalog system for CINDI system is simulated to show the complete CINDI Book Bag. Section 5 of this report – Integration of Book Bag and

CINDI – will describe how to merge the components of CINDI Book Bag system into CINDI virtual library system. As a main user interface/page of CINDI Library, this Category page is shown when the user visits this site. From here the user can find items by browsing the Category, Subject, Subsubject, and Item hierarchy. The user is able to browse all the materials in the virtual library through the hierarchy of its Semantic Header. If the user finds some interesting items, he/she can further view the short description about the item.

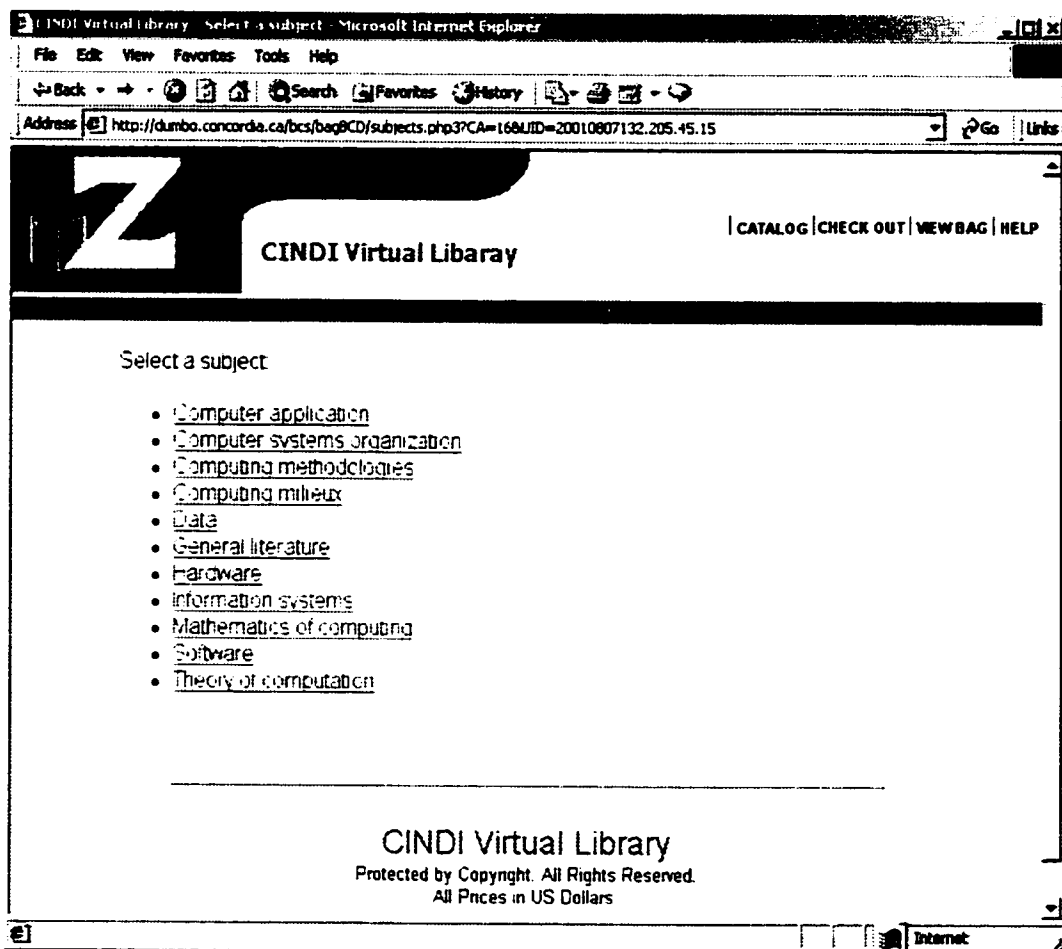


Figure2: CINDI library subjects – second level of the catalog hierarchy

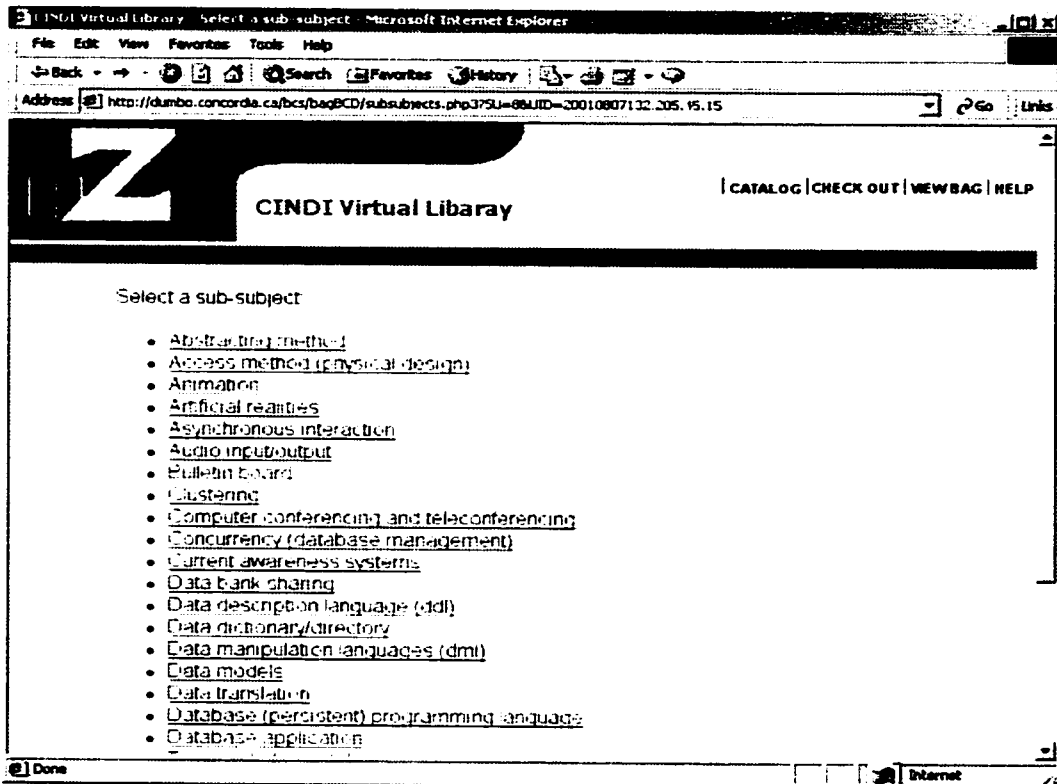


Figure 3: CINDI library sub-subjects – third level of the catalog hierarchy

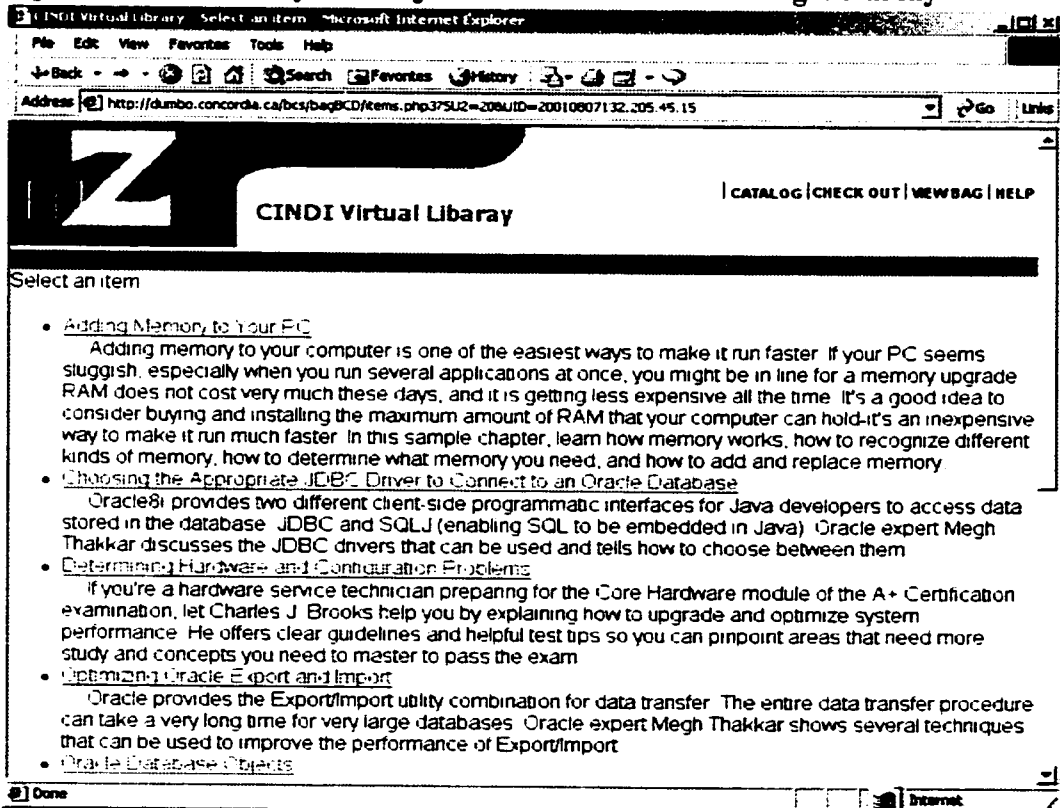


Figure 4: CINDI library sub-subjects – last level of the catalog hierarchy

The user can click on the item he/she is interested in and get detailed information about this item as shown in **Figure 5**.

If the user wants to buy an item, he/she can add the item to his/her bag by clicking on "Add to My Bag" button, as shown in **Figure 6**. Then a Book Bag for this user is shown which includes the item the user selected and its detailed information. The user can then continue shopping by clicking on the corresponding button. All the selected items will be listed together and the total cost is calculated.

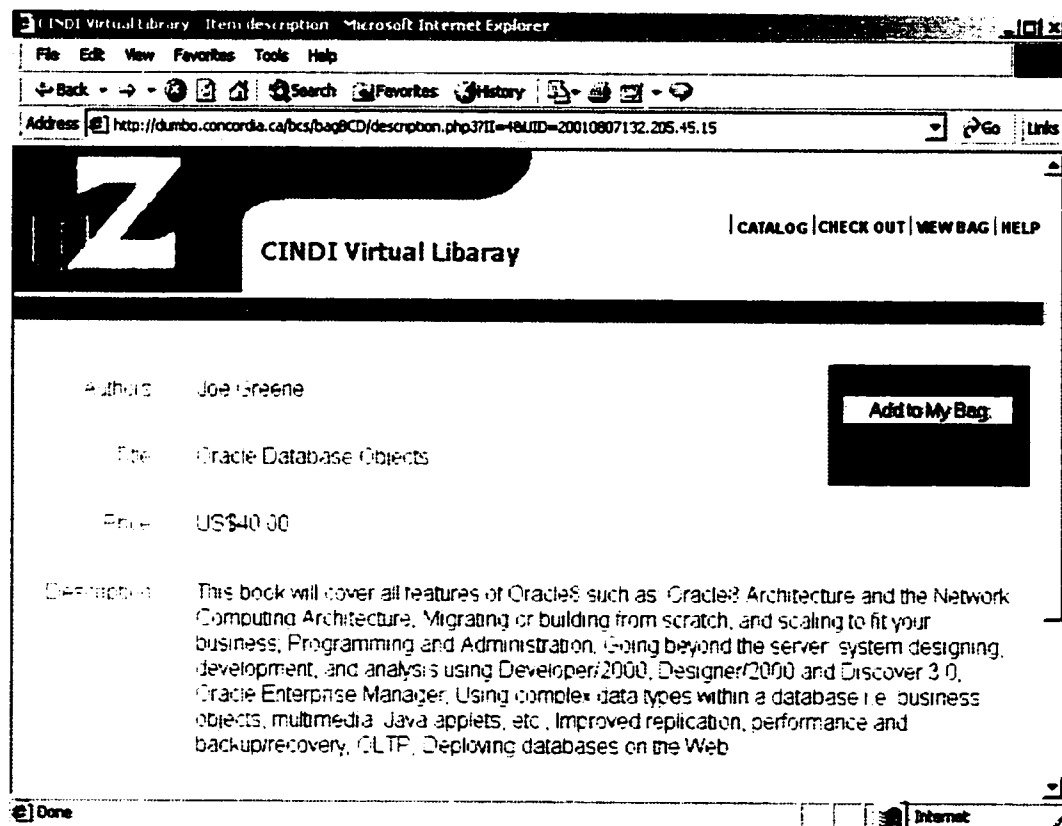


Figure 5: Description of an item the user selected

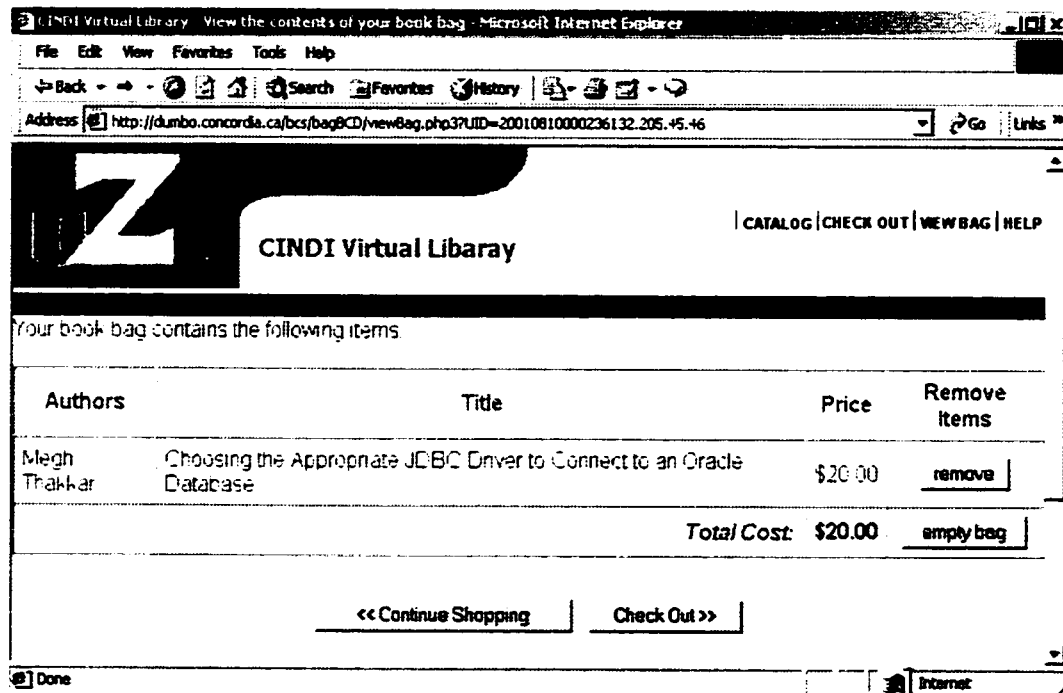


Figure 6: Detailed Book Bag information of the user

4.4 The Book Bag Status

Suppose a user has put some items into his/her Book Bag as shown in **Figure 7**, which depicts the status of the user's bag. It allows the user to view all items he/she selected. The user can also make corrections if he/she wants. The user can remove the items he/she selected one by one if he/she doesn't want to buy, or he/she can empty the Bag by clicking on "empty bag" button if he/she wants to buy none of the selected items.

In **Figure 7**, there is a button called "Check Out". This button is used when the user finishes verifying all the items in the list and decides to place an order as shown in

Figure 8.

CINDI Virtual Library | CATALOG | CHECK OUT | VIEW BAG | HELP

Your book bag contains the following items

Authors	Title	Price	Remove Items
Megh Thakkar	Choosing the Appropriate JDBC Driver to Connect to an Oracle Database	\$20.00	remove
Megh Thakkar	Optimizing Oracle Export and Import	\$30.00	remove
Joe Greene	Oracle Database Objects	\$40.00	remove
Total Cost:		\$90.00	empty bag

[<< Continue Shopping](#) [Check Out >>](#)

Figure 7: The status of the user's book bag

CINDI Virtual Library | CATALOG | CHECK OUT | VIEW BAG | HELP

Your book bag contains the following items

Authors	Title	Quantity	Cost	Total	Remove Items
Megh Thakkar	Choosing the Appropriate JDBC Driver to Connect to an Oracle Database	1	\$20.00	\$20.00	remove
Megh Thakkar	Optimizing Oracle Export and Import	1	\$30.00	\$30.00	remove
Joe Greene	Oracle Database Objects	1	\$40.00	\$40.00	remove
Total Cost:				\$90.00	empty bag

Are you sure that you want to check out?

[Yes](#) [No](#)

Figure 8: Confirmation of the check out decision by the user

4.5 The Checkout Counter

In **Figure 8**, the user can still change his/her mind if he/she will buy the item or not, and then make the confirmation for checking out. If the user chooses “No”, the previous screen will be displayed. If the user chooses “Yes”, next page will be shown (**Figure 9**) and the user will be asked to provide personal and payment information like: name, email, phone, payment method, and some optional information. If the user chooses to pay by credit card, he/she should also provide his/her credit card information as mandatory required information.

The screenshot shows a web browser window with the address <http://dumbo.concordia.ca/bcs/bag@CD/UserInfo.php3>. The page content is as follows:

Please fill out the information below

(*) - mandatory information

Your name(*)	Do you need order confirmation by email?
<input type="text"/>	<input checked="" type="radio"/> No
Your e-mail address(*)	<input type="radio"/> Yes
<input type="text"/>	Your address
Payment method(*)	<input type="text"/>
<input checked="" type="radio"/> Credit card	Your city and state
<input type="radio"/> Check or Money order	<input type="text"/>
If you pay by credit card	Your postal/zip code and country
Credit Card type(*)	<input type="text"/>
<input type="text"/>	Your daytime (work) phone number
Credit Card number(*)	<input type="text"/>
<input type="text"/>	Your evening (home) phone number
Credit Card expiration date(*)	<input type="text"/>
<input type="text"/>	

Figure 9-1: The user provides information for check out

After the user filled all the needed information for check out, and click the "Submit" button, a receipt of the order is shown to the user (**Figure 10**).

This is your receipt. Please print this receipt for your records.

Order Number: 12

Make check with order number payable to and Send check to
CINDI Virtual Library
 1455 de Maisonneuve West
 Montreal PQ, H3G 1M8

Phone (514)848-0000
 Web: <http://dumbo.concordia.ca/bcs/bagBCD>
 E-mail: info@dumbo.concordia.com
 Order Date: 2001-09-06

Order For:
 B. Zhao

You have ordered the following items...

Authors	Title	Quantity	Cost	Subtotal
Megh Thakkar	Optimizing Oracle Export and Import	1	\$30.00	\$30.00
Megh Thakkar	Choosing the Appropriate JDBC Driver to Connect to an Oracle Database	1	\$20.00	\$20.00
Joe Greene	Oracle Database Objects	1	\$40.00	\$40.00
Total:			\$90.00	\$90.00

You have elected to pay via Check or Money Order.
 We will ship the products you ordered upon receipt of your payment.

Please include the order number with the payment.

If you have questions about this order, please contact your librarian at:
info@dumbo.concordia.com
 or
 (514)848-0000

Please refer to order number 12

If you need access to this document again, you can bookmark the following web page:
<http://dumbo.concordia.ca/bcs/bagBCD/bagBCD/Receipt.php?Order=2001090612&SPM=Check&CT=SCo>

Thank you for ordering from CINDI Virtual Library!

[CINDI Virtual Library Home](#)

Figure 10: The receipt of the order for the buyer

The receipt page is a summary of the information about the order the buyer just placed. It includes the order number, the information of the CINDI virtual library, the name of the buyer, the full list of the items the buyer purchased, the payment method and its information. This page also provides a choice for the buyer to bookmark the receipt page in order to access the document later on.

At the bottom of the receipt, a link to "CINDI Virtual Library Home" is provided. Usually a customer would say bye-bye after he/she checks out, but in case some of them want to make other purchase, they can use the link provided.

An excellent feature that makes e-commerce sites more usable and more comfortable is to allow the user to access his/her shopping Book Bag and to check out whenever he/she wants. We provide this feature not only after the user adds an item to his/her Bag (**Figure 7, 8**), but also in all the pages of the CINDI library he/she surfs. If the user has selected one or more items, he/she can click on "View Bag" or "Check Out" button to see his/her bag status or go to the check out stand. If the user doesn't make any selection, he/she is given a message showing "the bag is empty".

After the payment of the order is processed, all the items in the order will be shipped to the buyer/customer by CINDI virtual library.

4.6 Site Administration

As a user of the CINDI library site, the administrator needs to log-in when he/she wants to access it with his/her UserID and password as shown in **Figure 11**. After the administrator has successfully logged in, the main page of the administration will be displayed as shown in **Figure 12**. Through this page the administrator can review the order records of all the buyers. He/she can manipulate the catalog hierarchy of the CINDI library: add a category, a subject, a subsubject, and an item; delete a category, a subject, a subsubject, and an item; update an item.

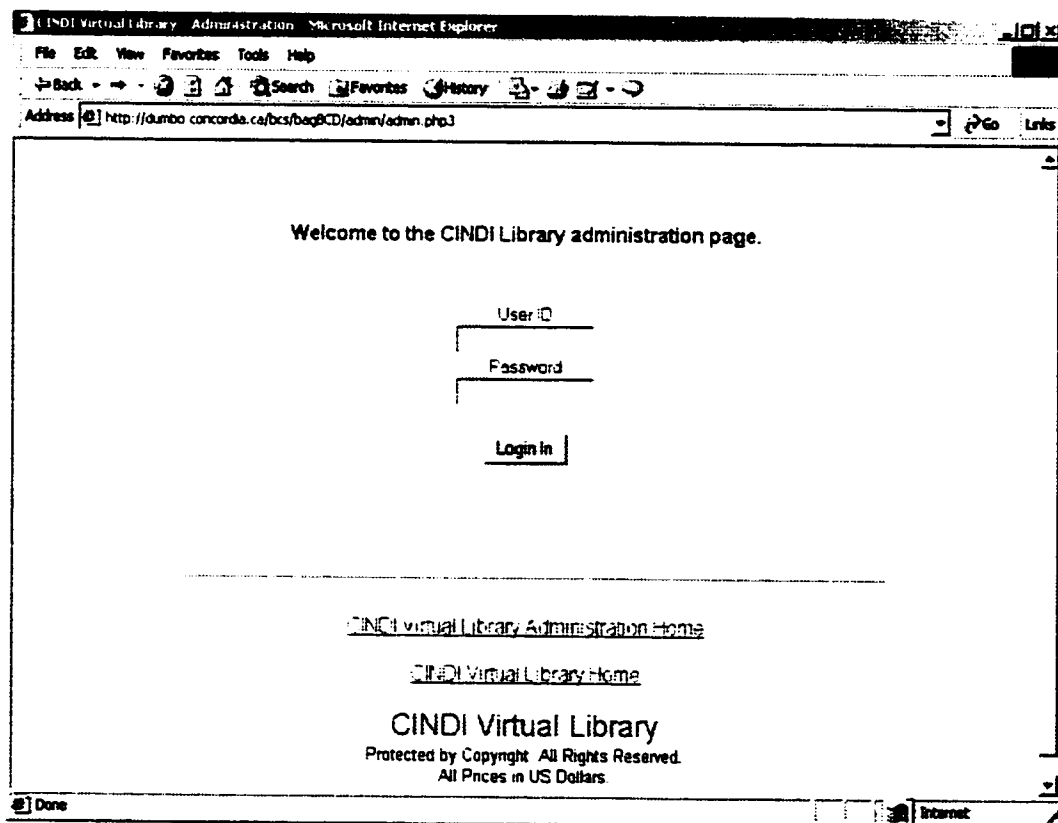


Figure 11: The login page for the administrator of the CINDI library

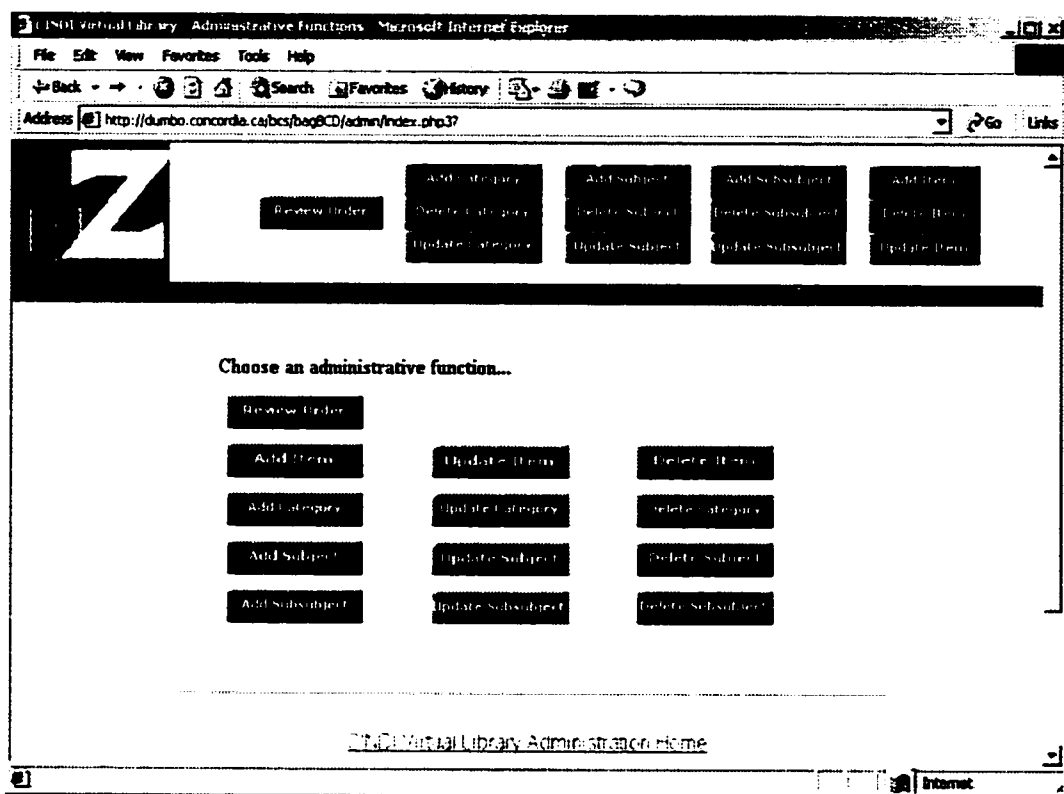


Figure 12: The main page of the administration

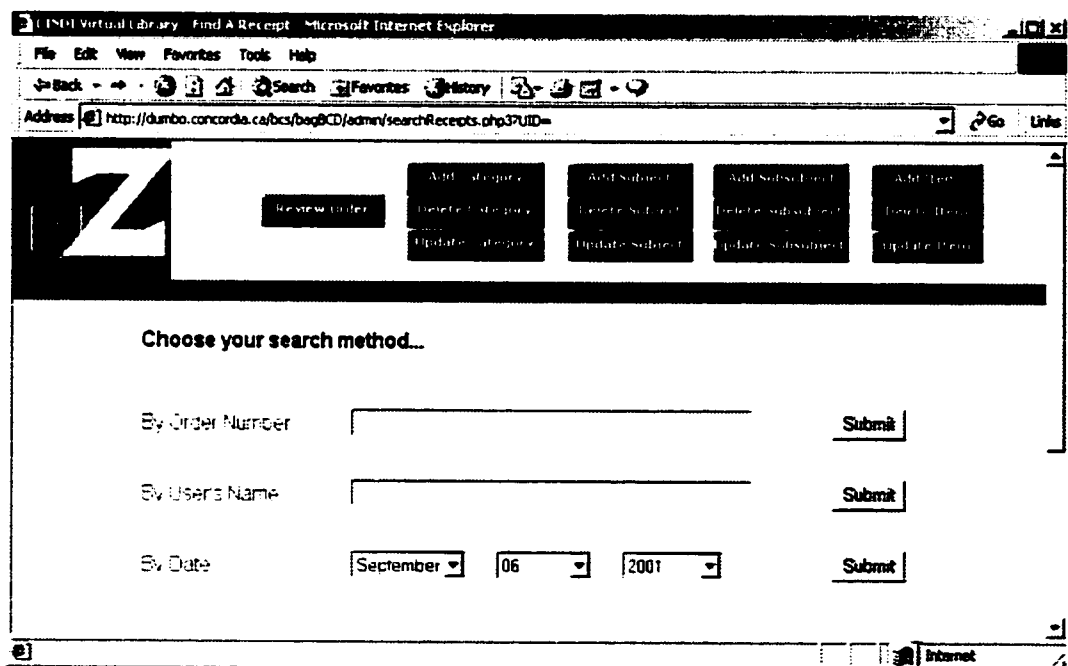


Figure 13: The order/receipt review page

When “Review Order” button is clicked, another screen is shown as in **Figure 13**. From here the administrator can retrieve order(s) by filling the order number, or the user name, or the date. A response page of search will be popped up as shown in **Figure 14** (search by order number or user name), or **Figure 15** (search by order date). Then the administrator can select the proper order to view the details of the order. Here “search by date” option has many available combinations of order search. The administrator can review all orders made on certain day by selecting a date to get all the orders information for that date. The administrator can review all orders in certain month by selecting a month and year, and selecting the date as “Any Day”. Similarly, he/she can review all orders information in certain year, and all orders in CINDI virtual library.

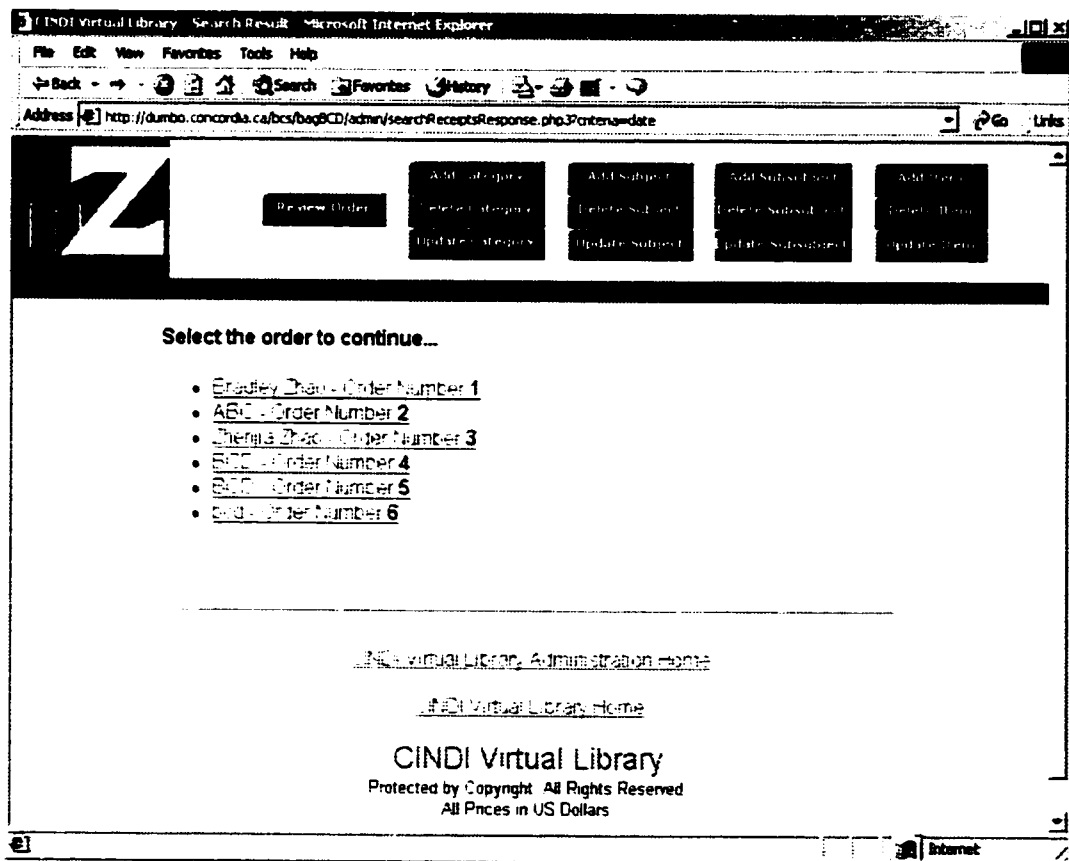


Figure 14: Response page of searching by date of the order

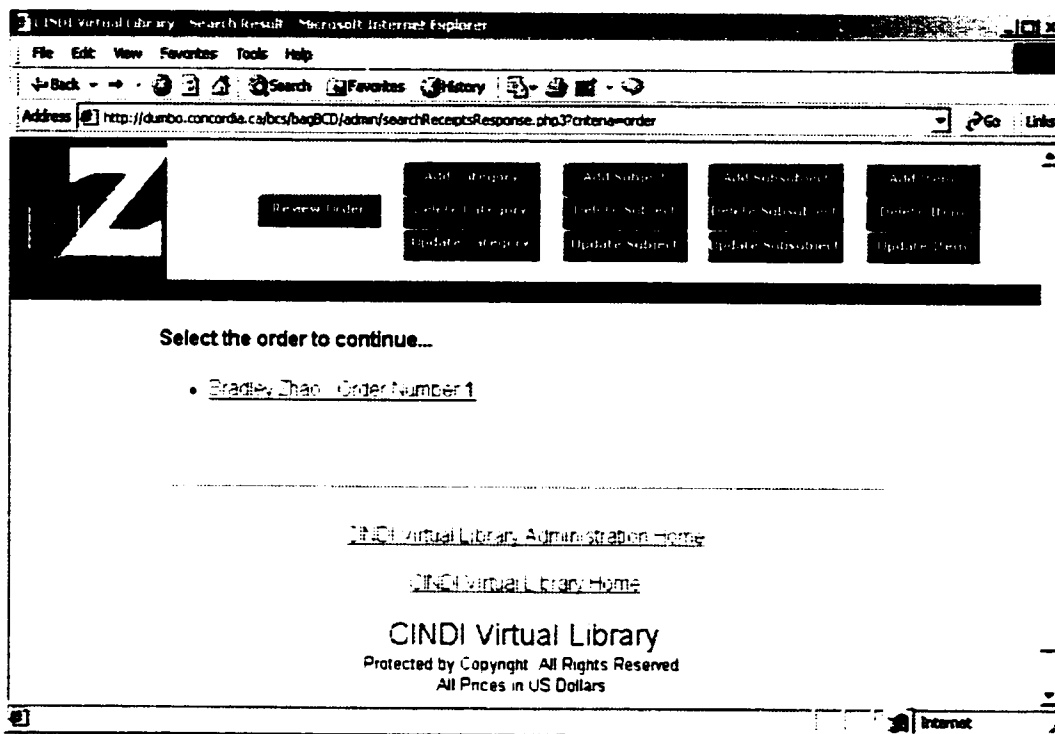


Figure 15: Response page of searching by the order number or user name

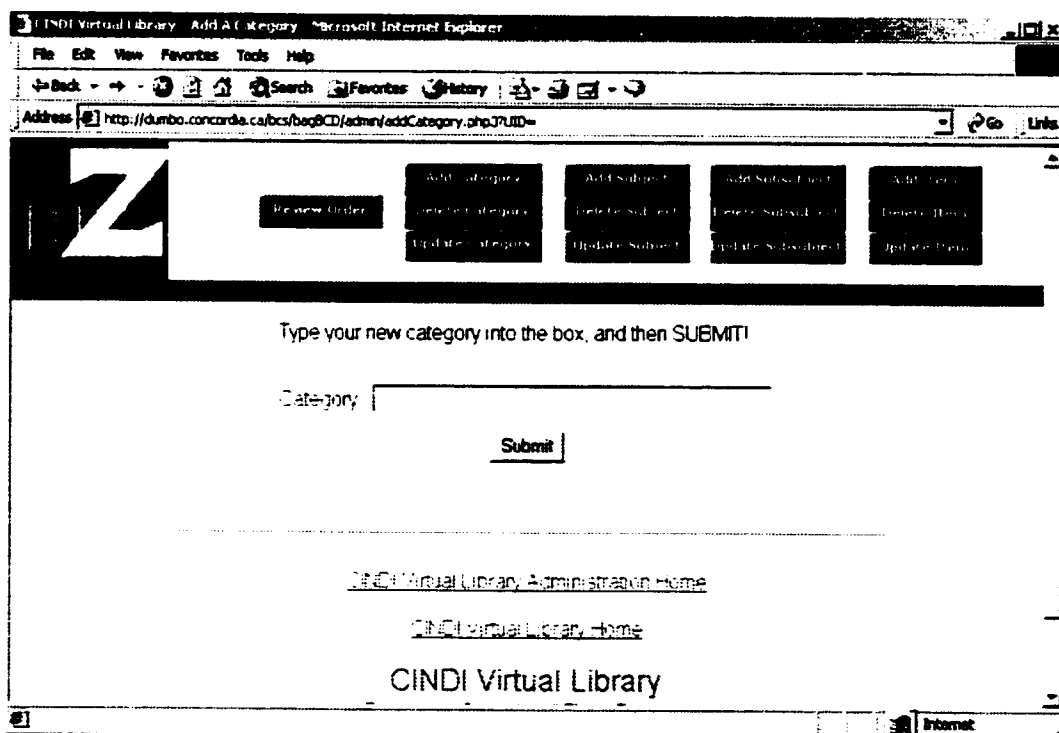


Figure 16: Add a category to the library

The administrator can add a category by clicking on “Add Category” button on the top of the administration page. The administrator can do it by typing the category name and submitting it as shown in **Figure 16**. The administrator can also delete a category by clicking on button “Delete Category”, a screen will then be popped up as shown in **Figure 17**. The administrator should make sure that the category he/she wants to delete has no any child hierarchy exists, otherwise he/she will be asked to cancel this action.

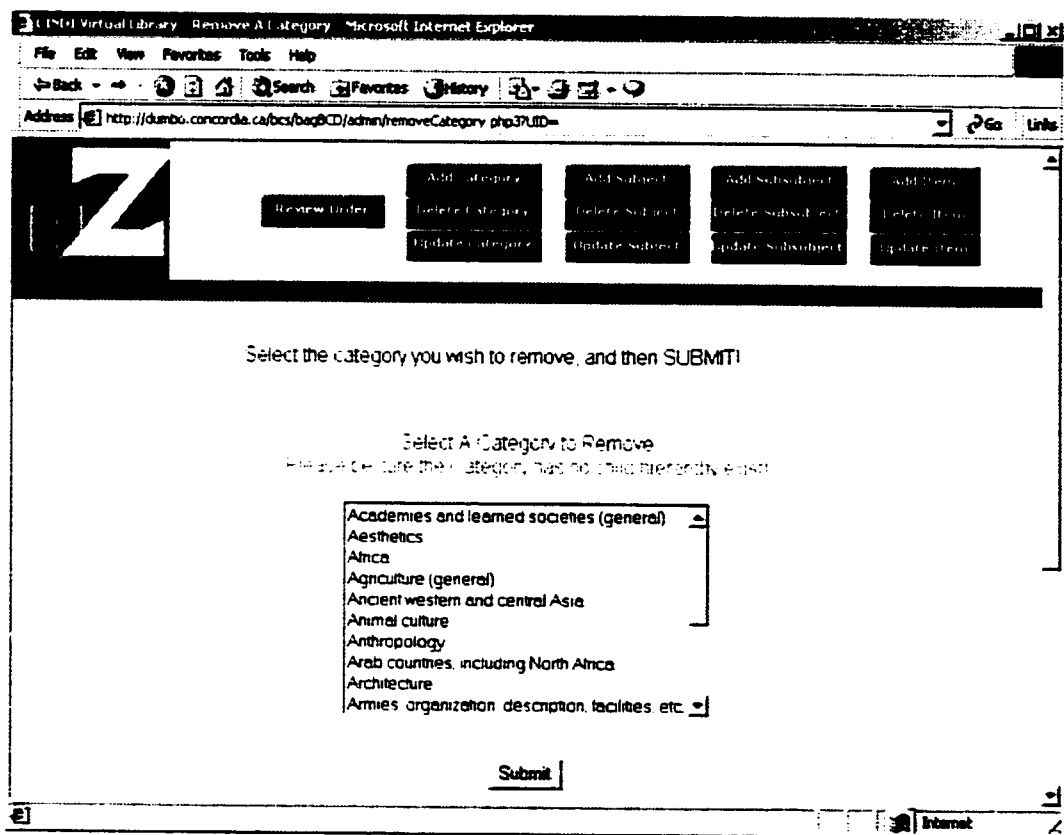


Figure 17: Delete a category from the library

If the administrator needs to update a category, he/she can click on “Update Category” button. First he/she should select a category to be updated, then the category he/she selected will be displayed for updating information. The updated information will be saved into database when the administrator clicks on “Submit” button.

The administrator can add a subject to the library through the “Add Subject” button. The “add a subject screen” is popped up as shown in **Figure 18**. The administrator should select a category of the subject, then enter the subject and submit it.

Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Search Favorites History

Address <http://dumbo.concordia.ca/bcs/bag8CD/admin/addSubject.php?PUID=> Go

Z

Review Order

Add Category Delete Category Update Category

Add Subject Delete Subject Update Subject

Add Subsubject Delete Subsubject Update Subsubject

Add Item Delete Item Update Item

Select the category and Type your new subject into the box, and then SUBMIT!

Select A Category

Academies and learned societies (general)

Aesthetics

Africa

Agriculture (general)

Ancient western and central Asia

Animal culture

Anthropology

Arab countries including North Africa

Architecture

Arms organization description facilities, etc.

Type A Subject

Submit

Internet

Figure 18: Add a subject to the library

Similarly, the administrator can delete a subject, update a subject, add a sub-subject, delete a sub-subject, and update a sub-subject.

The administrator can add an item to the CINDI library by clicking the “Add Item” button at the top of the administration page as shown in **Figure 19**. The administrator needs to select the category, the subject, and the sub-subject the item belongs to, and enter the information of the item, and then submit it.

UNOT Virtual Library - Add An Item - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Search Favorites History

Address <http://dumbo.concordia.ca/focs/bag8CD/admin/addItem.php?UID=> Go Links

Z

Review Order

Add Category Delete Category Update Category

Add Subject Delete Subject Update Subject

Add Subsubject Delete Subsubject Update Subsubject

Add Item Delete Item Update Item

Type new item into the boxes, and then SUBMIT!

Authors

Title

Item Cost (US\$)

Select A Category
Computer sciences

Select A Subject
Information systems

Select A Subsubject
Database application

Item Description

Submit

Figure 19: Add an item to the library

Figures 20, 21, and 22 are showing the deletion of an item from the library. First, the Title and/or the Authors name of the item should be entered (**Figure 20**). Then the details of the item will be popped up and the administrator can see if this is the right item that he/she wants to delete (**Figure 21**). And finally, the administrator will be asked to confirm the deletion (**Figure 22**).

UNO Virtual Library - Remove An Item - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Search Favorites History

Address http://dumbo.concordia.ca/bcs/bagBCD/admin/removeItem.php3UID=

Z Review Order Add Category Delete Category Update Category Add Subject Delete Subject Update Subject Add Subsubject Delete Subsubject Update Subsubject Add Item Delete Item Update Item

Fill the Title and/or Authors of the item to remove, and then SUBMIT!

Title

Authors

Figure 20: Delete an item from the library

UNO Virtual Library - Remove An Item - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Search Favorites History

Address http://dumbo.concordia.ca/bcs/bagBCD/admin/removeItem1.php3

Z Review Order Add Category Delete Category Update Category Add Subject Delete Subject Update Subject Add Subsubject Delete Subsubject Update Subsubject Add Item Delete Item Update Item

Is this item that you want to delete? If Yes, SUBMIT! If No, Back.

Authors

Title

Cost

ItemDescription

Figure 21: Showing the detail of the item

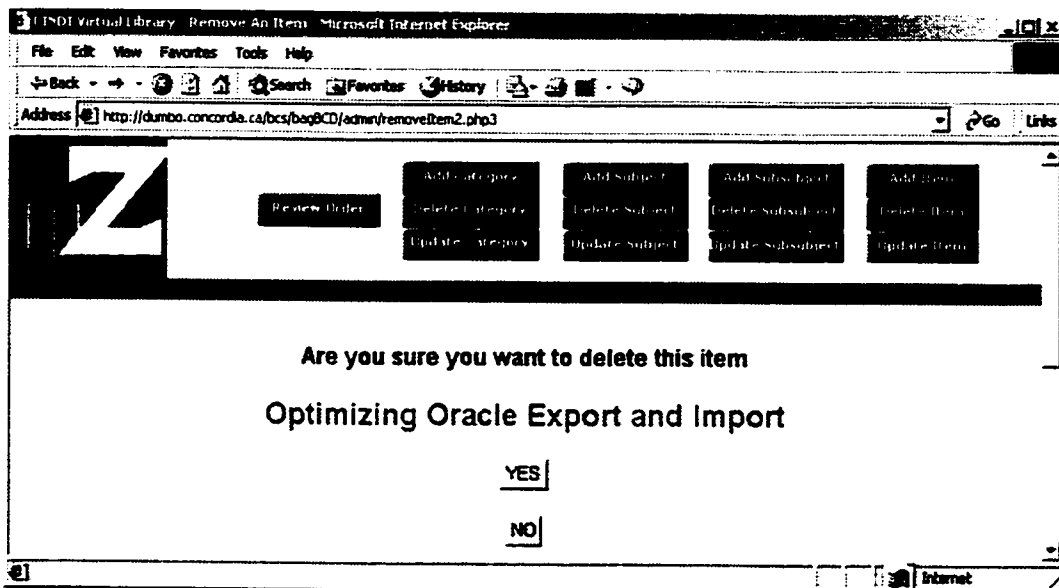


Figure 22: Asking for the confirmation of the deletion

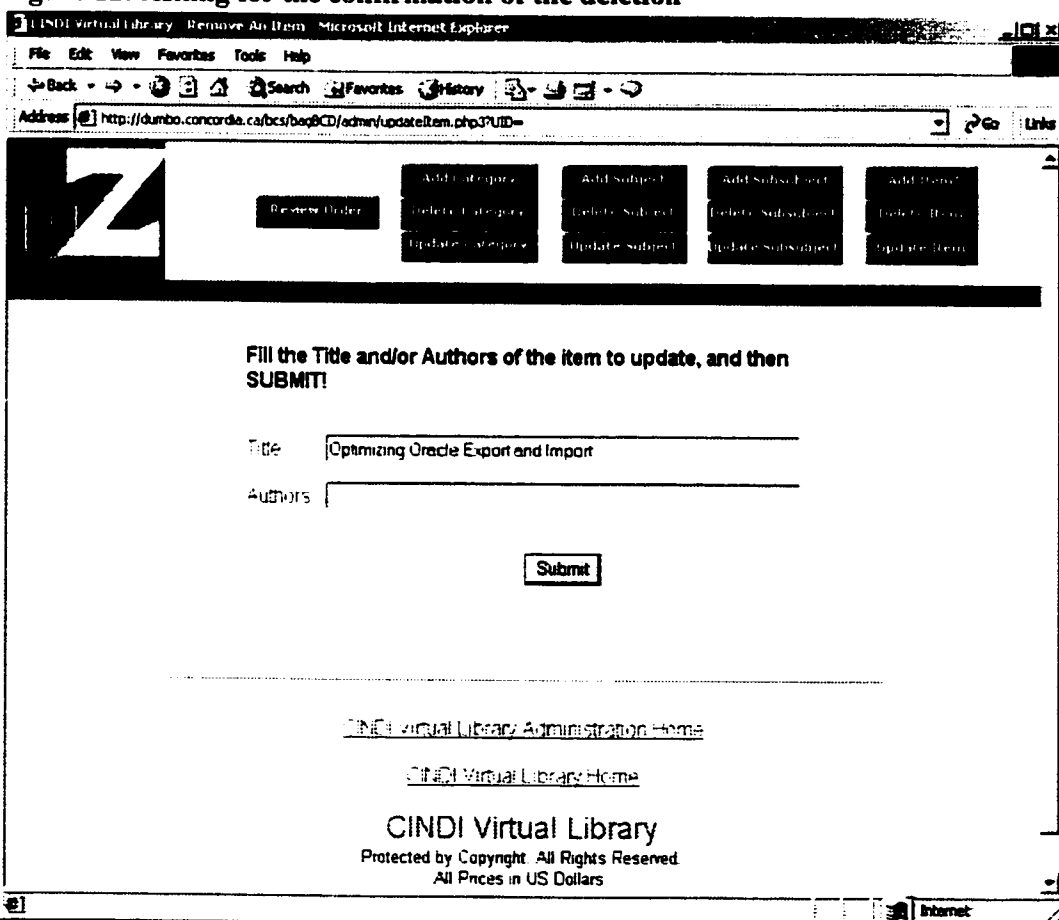


Figure 23: First Screen to update an item in the library

The items in the library can be updated by clicking the "Update Item" button. Similar to "Delete Item" function, it first pops up a screen for filling the key information of the item: Title and/or Authors as shown in **Figure 23**. Then the detail information about the item will be displayed as shown in **Figure 24**. The administrator can make the change and then submit it.

Virtual Library - Update An Item - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites History

Address http://dumbo.concordia.ca/bcs/bag9CD/admin/updateItem2.php3

Go Links

Z

Re new Order

Add Category Delete Category Update Category

Add Subject Delete Subject Update Subject

Add Subsubject Delete Subsubject Update Subsubject

Add Item Delete Item Update Item

Type the new data into the boxes, and then SUBMIT!

Authors
Megh Thakkar

Title
Optimizing Oracle Export and Import

Cost
30.00

Select A Category
Academies and learned societies (general)

Select A Subject
Computer application

Select A Sub-subject
Database application

ItemDescription
Oracle provides the Export/Import utility combination for data transfer. The entire data transfer procedure can take a very long time for very large databases. Oracle expert Megh Thakkar shows several techniques that can be used to improve the performance of Export/Import.

Submit

Internet

Figure 24: Change the item information for updating

4.7 Error Prevention and Message Prompting

According to the principles of user interface design (refer to section 3.4), we have already applied some of the rule during our design and implementation of the CINDI virtual library book bag system. These are simplicity, user familiarity, recoverability and consistency, which have been shown in the description of the system design and implementation. Now, some other rules of interface design like message prompting and error prevention will be further shown by the example figures as shown in **Figure 25, 26, and 27.**

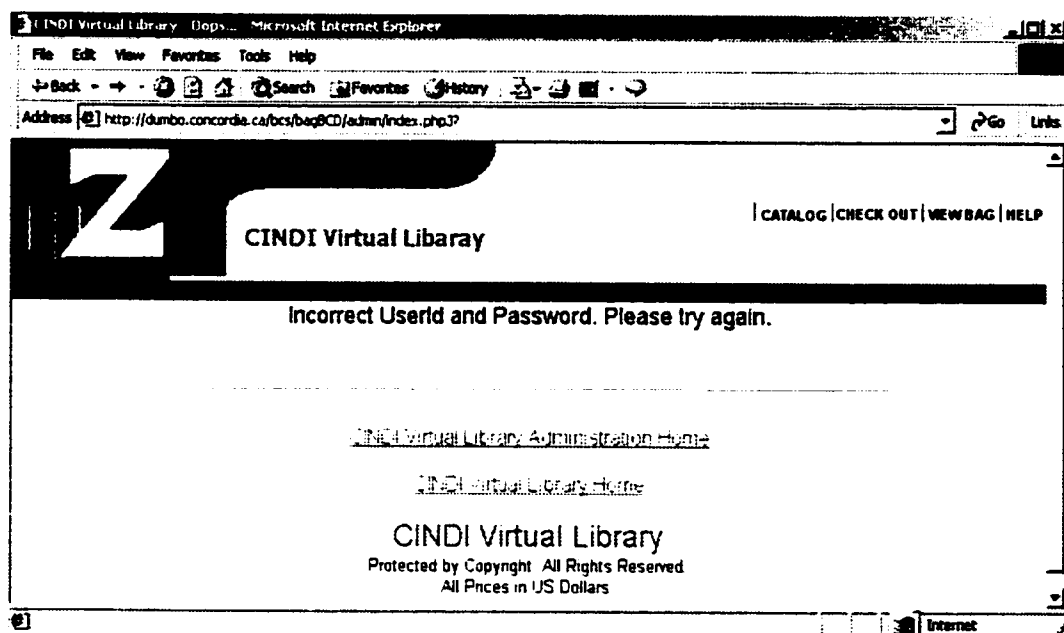


Figure 25: Administrator login failure response

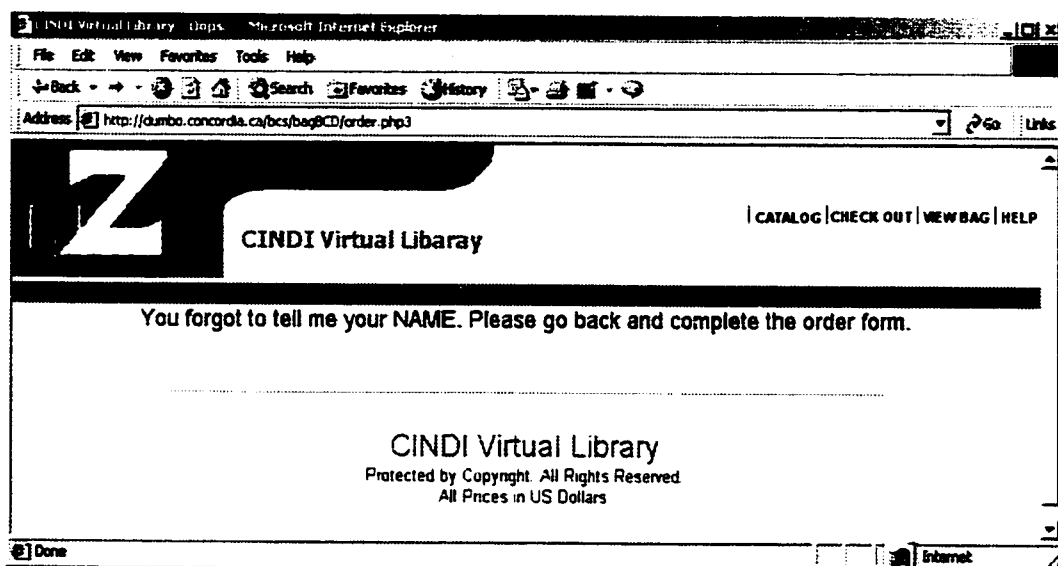


Figure 26: Order page response when the required user information is not provided

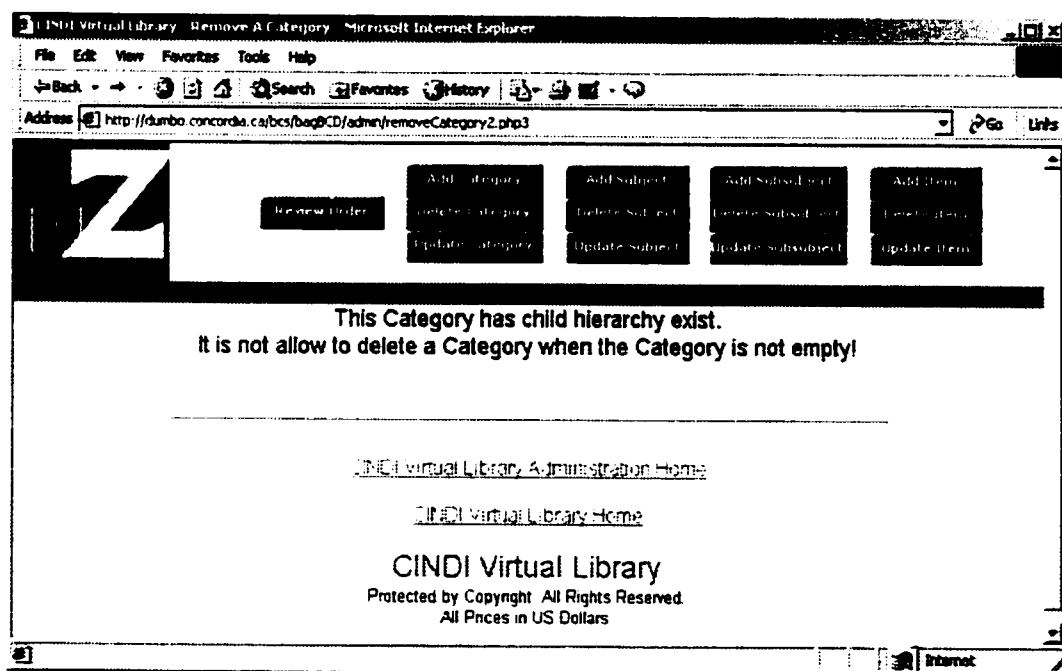


Figure 27: "Delete Category" warning page

5. Integration of Book Bag System in the CINDI System

When the Book Bag System of CINDI library is ready to be displayed, an important issue arises: integrating the Book Bag system into CINDI virtual library system.

1) Book bag system files list

In the CINDI book bag system root directory, there are several files:

- addBag.php3
- bag.php3
- checkout.php3
- description.php3
- emptyBag.php3
- index.php3
- items.php3
- order.php3
- receipt.php3
- subjects.php3
- subsubjects.php3
- userInfo.php3
- viewBag.php3

There are two subdirectories under the root directory, called "admin" and "images". As the names imply, the "images" directory holds some images used in the user system. The "admin" directory is for administration functions. Under the "admin" directory there is another subdirectory named "images" which contains some images used in the administration functions. The /admin directory has following files:

- addCategory.php3
- addCategoryResponse.php3
- addItem.php3
- addItemResponse.php3
- addSubject.php3
- addSubjectResponse.php3
- addSubsubject.php3
- addSubsubjectResponse.php3
- admin.php3
- bag.php3

- index.php3
- receipt.php3
- removeCategory.php3
- removeCategory2.php3
- removeCategoryResponse.php3
- removeSubject.php3
- removeSubject2.php3
- removeSubjectResponse.php3
- removeSubsubject.php3
- removeSubsubject2.php3
- removeSubsubjectResponse.php3
- removeItem.php3
- removeItem1.php3
- removeItem2.php3
- removeItemResponse.php3
- searchReceipts.php3
- searchReceiptsResponse.php3
- updateItem.php3
- updateItem2.php3
- updateItemResponse.php3
- updateCategory.php3
- updateCategory2.php3
- updateCategoryResponse.php3
- updateSubject.php3
- updateSubject2.php3
- updateSubjectResponse.php3
- updateSubsubject.php3
- updateSubsubject2.php3
- updateSubsubjectResponse.php3

2) Set up MySQL database

In the Book Bag system, we have 10 tables created in the "Bag" database. Please refer to section 4.2 of Database Scheme. The database creation script is included.

3) Modify the bag.php3 file in both root directory and /admin directory

Replace the variables in the first section of both bag.php3 files to reflect information about how to contact CINDI. Replace the variables in the second section of bag.php3 files with the proper database Host Name, Username, Password and Database Name. Replace the variables in the third section with absolute (Linux) path and relative

(to web users) path to the scripts. Do not end either of these with a forward slash (/).

however it is required to prefix the \$Relative variable with a forward slash.

4) Set up the catalog

Go to the online /admin directory and add the categories/subjects/subsubjects/items, according to the CINDI virtual library Semantic Headers. Once this has been done, the system can be tested and displayed.

6. Discussion and Further Work

6.1 Maintaining State with PHP Session

For maintaining state and user authentication, a common approach is the use of a "session" to store specific bits of information when a client visits a Web site; this session data is preserved for the duration of the visit, and is usually destroyed on its conclusion. A session can thus be considered a basket of information that contains a host of variable-value pairs; these variable-value pairs exist for the duration of the visit, and can be accessed at any point during it. This approach provides an elegant solution to the "stateless" nature of the protocol, and is used in many of today's larger sites to track and maintain information for personal and commercial transactions.

Session management is a mechanism to maintain state about a series of requests from the same user across some period of time. That is, the term "session" refers to the time that a user is at a particular web site. Session management refers to the way that associates data with a user during a visit to a Web page. Every session created is associated with a unique identification string which is sent to the client, while a temporary entry with the same unique identification number is created on the server, either in a flat file or in a database. It now becomes possible to register any number of "session variables" - these are ordinary variables, which can be used to store textual or numeric information, and can be read/written throughout the session.

PHP3 supports session through PHPLIB, a set of powerful PHP classes which adds seamless session management to PHP3-based sites. PHP4 includes support for session creation and maintenance. Unlike PHP 3, PHP 4.0 has built-in capabilities to

handle session management. PHP 4 sessions support the following methods of passing the session ID:

- Cookies which are the default way to pass the session ID between pages;
- Using GET/POST method to pass the ID;
- Hidden in the URL, either done manually or by automatic URL rewriting. Automatic URL rewriting is one of the very cool new features of PHP 4, allowing us to add the session ID to all the links within the page. To enable it, we need to configure PHP with --enable-trans-id and recompile it. Then the session ID will be added to all relative links within your PHP-parsed pages. While this is a handy feature, it should be used with caution on high-performance sites. PHP has to look at each individual page, analyze it to see whether it contains relative links, and eventually add the ID to the links. This obviously introduces a performance penalty. Cookies, on the other hand, are set only once, avoiding the overhead of URL rewriting.

Session gives us freedom, flexibility and functionality that are associated with any good server-side scripting language. However, PHP4 session has a few limitations: First of all, we cannot store objects in the sessions, which would be absolutely fantastic. Just imagine storing a complete user object in the session! Second, storing data in session variables is not very efficient because PHP4 is using files to store session information.

6.2 Using User Id and Password to Authenticate the General Users

In section 3.3.2 of this paper, we discussed the user authentication issue. We applied these concepts in the CINDI virtual library Web site administration part. The special user of this site – administrator is required to provide his/her user ID and password before he/she gets into the administration site. In our system, we will not ask

the general user - Web surfer to keep a user ID and password. The user will be required to provide his/her personal and payment information only when he/she decides to place an order. The advantage of this approach is that the user doesn't have to keep a user ID and password for his/her shopping in the CINDI library. The user will feel free when he/she browses the library. The disadvantage is that the user is required to provide his/her personal and payment information each time when he/she places an order in this library.

Another popular method now is using user ID and password to authenticate the general user like we applied in the administration part. In the administration part, because all the information in this part has to be protected, the administrator is required to provide his/her user ID and password at the very beginning. In the general user part, the contents of the library should be freely accessible by any user. Therefore the user authentication can be applied when the user wants to add item to his/her Bag. Here we can ask the user to make a registration if he/she is a new user, or ask the user to login with his/her user ID and password if he/she has already registered. A common way of registration is divided into two phases. In the first phase, the registration is asked when a user wants to add an item into his/her Bag just with user ID and password. In the second phase, the registration is asked when the user places an order first time at CINDI library. This time the user is asked to provide detailed personal information as well as the payment information. By this way, the user ID and password can also be used to maintain persistent information instead of using the IP address and date and time.

In summary, the user is asked to register only when he/she is new for adding an item to the Bag and for placing an order. Later, the user will only be asked to provide his/her user ID and password for authentication purpose.

7. Conclusion

The implementation of the CINDI Book Bag system shows that PHP-MySQL-Apache-Linux is a good combination for developing an interactive data-driven web site. PHP is full of functionality and is an easy-to-use scripting language. Developed by using PHP scripting language, the CINDI book bag system can have most of the functionalities compared to the other shopping carts/book bags, in term of the requirements of the CINDI virtual library. Using IP/Date/Time/Query String is a good and workable way to maintain persistent information on the stateless Web.

The CINDI Book Bag system written by using PHP can be easily integrated into CINDI virtual library system. Because CINDI Book Bag system is just a component of the CINDI virtual library, and is developed based on some simulated data, we will have more work to do when we come to the real CINDI virtual library system.

8. Appendix: Example Code

1) bag.php3 code (in the root directory):

```
<?
/* provide your information to the following variables */
$Company = "CINDI Virtual Library";
$Address1 = "1455 de Maisonneuve West";
$City     = "Montreal";
$State    = "PQ";
$Zip      = "H3G 1M8";
$Phone    = "(514)848-0000";
$Web      = "http://dumbo.concordia.ca/bcs/bagBCD";
$Email    = "info@dumbo.concordia.com"; // "bcdesai@cs.concordia.ca";

function Brand() {
global
$Company, $Address1, $Address2, $City, $State, $Zip, $Phone, $Web, $Email;
}

/* Replace these variables with information for connecting to the
database server */
$DBHost="localhost";           // $DBHost="your.database.host" - Database
Server machine
$DBUser="bradley";             // $DBUser="database.username" - Database
login
$DBPass="zbradley";            // $DBPass="database.password" - Database
password
$DBName="Bag";                 // $DBName="database.name" - Database
containing the tables
function DBInfo() {
global $DBHost, $DBUser, $DBPass, $DBName;
}

/* Replace these variables with the absolute and relative paths to your
Book Bag scripts */
$WebRoot="/bcs/bagBCD";        // absolute Path, where application is
installed
$Relative="/bcs/bagBCD";        // relative Path, where
application is installed
$WebHost="http://dumbo.concordia.ca/"; // HTTP Host
function Root() {
global $WebRoot, $Relative, $WebHost;
}

function redFont($font, $text) {
echo "<FONT FACE=\"$font\" COLOR=\"red\">$text</FONT>";
}

function blueFont($font, $text) {
echo "<FONT FACE=\"$font\" COLOR=\"blue\">$text</FONT>";
}

function colorFont($color, $font, $text) {
```

```

echo "<FONT FACE=\"Sfont\" COLOR=\"Scolor\">Stext</FONT>";
}

function fontFace($font,$text) {
echo "<FONT FACE=\"Sfont\">Stext</FONT>";
}

function fontSize($size,$color,$font,$text) {
echo "<FONT FACE=\"Sfont\" COLOR=\"Scolor\"
SIZE=\"Ssize\">Stext</FONT>";
}

function commonHeader($Company,$title) {
echo "<HTML><TITLE>$Company - $title</TITLE>";
echo "<body leftmargin=\"0\" topmargin=\"0\" rightmargin=\"0\"
marginwidth=\"0\" marginheight=\"0\" bgcolor=\"#ffffff\">";

//cindi pages header
echo "<table WIDTH=\"100%\" BORDER=\"0\" CELSPACING=\"0\"
CELLPADDING=\"0\">";
echo "<tr height=\"106\"><td valign=\"top\"><img border=\"0\"
src=\"images/cindi_logo.gif\" width=\"126\" height=\"102\">";
//echo "</td><td valign=\"top\">";
echo "<img border=\"0\" src=\"images/ban.gif\" width=\"266\"
height=\"102\"></td>";
echo "<td align=\"right\">|";
echo "<a href=\"index.php3\">";
echo "<img align=middle border=\"0\" src=\"images/zcatalog.gif\"
width=\"60\" height=\"22\"></a>|";
echo "<a href=\"checkout.php3?UID=$UID\">";
echo "<img align=\"middle\" border=\"0\" src=\"images/zcheckout.gif\"
width=\"70\" height=\"22\"></a>|";
echo "<a href=\"viewBag.php3?UID=$UID\">";
echo "<img align=\"middle\" border=\"0\" src=\"images/zviewbag.gif\"
width=\"60\" height=\"22\"></a>|";
echo "<a href=\"help.php3?UID=$UID\">";
echo "<img border=\"0\" align=\"middle\" src=\"images/zhhelp.gif\"
width=\"35\" height=\"22\"></a>";
echo "</td>";
echo "</tr>";
echo "<tr><td colspan=\"3\" height=\"15\" bgcolor=\"#0078A8\"
width=\"100%\"><hr color=\"#0078A8\"></td></tr>";
echo "</table>";
}

function receiptHeader($Company,$title) {
echo "<HTML><TITLE>$Company Receipt - $title</TITLE><BODY
BGCOLOR=\"#FFFFFF\">";
}

function commonFooter($Relative,$SUID) {
echo "<center><br><br><hr width=\"70%\"><br>";
echo "<font face=\"Arial\" size=\"5\" color=\"red\">CINDI Virtual
Library</font><br>";
fontSize("2","black","Arial","Protected by Copyright. All Rights
Reserved.");
echo "<br>";
}

```

```

fontSize("2","black","Arial","All Prices in US Dollars.");
echo "<br><br></center></BODY></HTML>";
}

function adminHeader($Company,$title) {
echo "<HTML><TITLE>$Company - $title</TITLE>";
echo "<body background=\"images/bg.jpg\" leftmargin=\"0\"
topmargin=\"0\" rightmargin=\"0\" marginwidth=\"0\" marginheight=\"0\"
bgcolor=\"#ffffff\">";

//cindi administration pages header
echo "<table WIDTH=\"100%\" BORDER=\"0\" CELSPACING=\"0\"
CELLPADDING=\"0\">";
echo "<tr><td width=\"126\" valign=\"top\"><img border=\"0\"
align=\"top\"src=\"images/cindi_logo.gif\" width=\"126\"
height=\"106\">";
echo "</td>";
echo "<td align=\"center\">";
echo "<a href='searchReceipts.php3'>";
echo "<img align=\"middle\" border=\"0\" src=\"images/getreceipt.gif\"
width=\"100\" height=\"25\"></a>";
echo "&nbsp;<a href='addCategory.php3'>";
echo "<img align=\"middle\" border=\"0\" src=\"images/addcat.gif\"
width=\"100\" height=\"25\"></a>";
echo "&nbsp;<a href='addItem.php3'>";
echo "<img align=\"middle\" border=\"0\" src=\"images/additem.gif\"
width=\"100\" height=\"25\"></a>";
echo "&nbsp;<a href='updateItem.php3?UID=SUID\">";
echo "<img align=\"middle\" border=\"0\" src=\"images/upitem.gif\"
width=\"100\" height=\"25\"></a>";
echo "&nbsp;<a href='removeCategory.php3?UID=SUID\">";
echo "<img align=\"middle\" border=\"0\" src=\"images/movecat.gif\"
width=\"100\" height=\"25\"></a>";
echo "&nbsp;<a href='removeItem.php3?UID=SUID\">";
echo "<img border=\"0\" align=\"middle\" src=\"images/moveitem.gif\"
width=\"100\" height=\"25\">";
echo "</td>";
echo "</tr>";
echo "<tr><td colspan=\"3\" height=\"15\" bgcolor=\"#0078A8\"
width=\"100%\"> <hr color=\"#0078A8\"></td></tr>";
echo "</table>";
}

function adminFooter($Relative) {
echo "<center><br><br><hr width=\"70%\"><br>";
blueFont("Arial","<a href=\"$Relative/index.php3\">CINDI Virtual
Library Catalog</a><br><br>");
echo "<font face=\"Arial\" size=\"5\" color=\"red\">CINDI Virtual
Library</font><br>";
fontSize("2","black","Arial","Protected by Copyright. All Rights
Reserved.");
echo "<br>";
fontSize("2","black","Arial","All Prices in US Dollars.");
echo "<br><br></center></BODY></HTML>";
}
?>

```

2) index.php3 code (in root directory)

```
<?
require("bag.php3");
DBinfo();
Brand();
Root();
mysql_connect("SDBHost","SDBUser","SDBPass");

/* delete the user and his/her BagItem from the table Users and
BagItem,
   if it has 1 day old */
Sold=date("z")-1;
mysql("SDBName","DELETE FROM Users WHERE Date < Sold");

$result=mysql("SDBName","SELECT BagItemsID,Date FROM BagItems");
while ($row=mysql_fetch_row($result)) {
   SCII=$row[0];
    SCDa=$row[1];
    $pieces=explode(":",SCDa);
    SDCHK=$pieces[1];
    if (SDCHK < Sold) {
        mysql("SDBName","DELETE FROM BagItems WHERE BagItemsID = 'SCII'");
    }
}

if ($SUID != "") {
    $result=mysql("SDBName","SELECT * FROM Users WHERE User='SUID'");
    $num=mysql_num_rows($result);
    if ($num == "0") {
        $sdt=date("YmdHis");
        $SUID="$sdt$REMOTE_ADDR";
        $sdate=date("z");
        mysql("SDBName","INSERT INTO Users VALUES ('SUID','sdate')");
        Header("Location: $Relative/index.php3?UID=$SUID");
    }
}

if ($SUID == "") {
    $sdt=date("YmdHis");
    $SUID="$sdt$REMOTE_ADDR";
    $sdate=date("z");
    mysql("SDBName","INSERT INTO Users VALUES ('SUID','sdate')");
    //Header("Location: $php3_SELF?UID=$SUID");
    Header("Location: $Relative/index.php3?UID=$SUID");
}

commonHeader("$Company","Select a category");

$result=mysql("SDBName","SELECT * FROM Category ORDER BY Category");
echo "<blockquote>";
echo "<blockquote>";
fontFace("Arial","Select a category:<br><br>");
echo "<ul>";
while ($row = mysql_fetch_row($result)) {
```

```

$Cat=$row[0];
$CatID=$row[1];
fontFace("Arial", "<li><a
href='SRelative/subjects.php3?CA=$CatID'>$Cat</a></li>");
    }
echo "</ul>";
echo "</blockquote>";
echo "</blockquote>";

commonFooter($Relative,$SUID);
?>

```

3) userInfo.php3 code (in root directory)

```

<?
$! = 0;
$# = 0;
require("bag.php3");
DBinfo();
Brand();
Root();
mysql_connect("$DBHost", "$DBUser", "$DBPass");

if ($SUID != "") {
$result=mysql("$DBName", "SELECT * FROM Users WHERE User='$SUID'");
$num=mysql_num_rows($result);
if ($num == "0") {
$dt=date("YmdHis");
$SUID= "$dt$REMOTE_ADDR";
$dt=date("z");
mysql("$DBName", "INSERT INTO Users VALUES ('$SUID', '$dt')");
Header("Location: Sphp3_SELF?UID=$SUID");
}
}

if ($SUID == "") {
$dt=date("YmdHis");
$SUID= "$dt$REMOTE_ADDR";
$dt=date("z");
mysql("$DBName", "INSERT INTO Users VALUES ('$SUID', '$dt')");
Header("Location: Sphp3_SELF?UID=$SUID");
}

commonHeader("$Company", "May I have your information please?");

blueFont("Arial", "Your book bag contains the following items:<br><br>");

echo "<center><table align=\"center\" border='1' cellpadding='5'
cellspacing='0'><tr>";
echo "<th>";
redFont("Arial", "Authors");
echo "</th>";
echo "<th>";
redFont("Arial", "Title");
echo "</th>";

```

```

echo "<th>";
redFont("Arial","Quantity");
echo "</th>";
echo "<th>";
redFont("Arial","Cost");
echo "</th>";
echo "<th>";
redFont("Arial","Total");
echo "</th>";
echo "<th colspan='2'>";
redFont("Arial","Remove Items");
echo "</th></tr>";

$result=mysql("$dbName","SELECT * FROM BagItems WHERE UserID='SUID'");
while ($row = mysql_fetch_row($result)) {
    $CUI=$row[0];
    $CII=$row[1];
    $CIQ=$row[2];
    $CDa=$row[3];
    $CCI=$row[4];
        $result2=mysql("$dbName","SELECT * FROM Items WHERE ItemID='SCII'");
        while ($row2=mysql_fetch_row($result2)) {
            $IIS=$row2[0];
            $IIN=$row2[1];
            $IID=$row2[2];
            $IIC=$row2[3];
            $ICa=$row2[4];
            $III=$row2[5];

$To[$i]=(($CIQ * $IIC)+($CIQ * $ISC));
echo "<tr><td>";
blueFont("Arial","$IIS");
echo "</td><td>";
blueFont("Arial","$IIN");
echo "</td><td>";
blueFont("Arial","$CIQ");
echo "</td><td>";
$IIC=number_format($IIC,"2",".", "thousands_sep");
blueFont("Arial","$$IIC");
echo "</td>";

echo "<td>";
$Tot=number_format($To[$i],"2",".", "thousands_sep");
blueFont("Arial","<b>$STot</b>");

echo "<td align='center'>";
echo "<FORM ACTION=\"$Relative/removeItem.php3\" METHOD=\"$POST\">";
echo "<input type=\"hidden\" name=\"BagItemsID\" value=\"$CCI\">";
echo "<input type=\"hidden\" name=\"UID\" value=\"$SUID\">";
echo "<input type=\"SUBMIT\" NAME=\"SUBMIT\" value=\"remove\">";
echo "</td></tr>";
echo "</form>";
Si--;
        }
}

```



```

while ($j < $i) {
    $total=$total-$to[$j];
    $j--;
}

echo "<tr><td colspan='2'>&nbsp;</td>";
echo "<td colspan='2'>";
redFont("Arial","<center><b><i>Total Cost: </i></b></center>");
echo "<td>";
$total=number_format($total,"2",".", "thousands_sep");
if ($SES == "1") {
    blueFont("Arial","<b><center>$total</b>");
    blueFont("Arial\" size=\"-2\", \" Plus Shipping</center>");
} elseif ($SES != "1") {
    blueFont("Arial","<b><center> $total </center></b>"); }

echo "<td colspan=\"2\"><center>";
echo "<FORM ACTION=\"$Relative/emptyBag.php3\" METHOD=\"POST\">";
echo "<input type=\"hidden\" NAME=\"UID\" value=\"SUID\">";
echo "<input type=\"SUBMIT\" NAME=\"SUBMIT\" value=\"empty bag\">";
echo "</center></td></tr></table></center>";
echo "</form>";

echo "<p>";

echo "<center>";
fontSize("-1","blue","Arial","<hr size=\"3\" color=\"blue\"
width=\"60%\"><b>Please fill out the information below</b><p>");
fontSize("-0","black","Arial","(*) - mandatory information<p>");

echo "<form action=\"$Relative/order.php3\" method=\"post\"><table
width='60%' border=\"0\"><tr><td>";

fontSize("-1","blue","Arial","Your name(*)<br>");
echo "<input type=\"text\" name=\"Name\" size=\"40\"><p>";

fontSize("-1","blue","Arial","Your e-mail address(*)");
//fontSize("-1","black","Arial","<br>(Confirmation will be sent to this
email)<br>");
echo "<br><input type=\"text\" name=\"BuyEmail\" size=\"40\"><p>";

blueFont( "Arial\" size=\"-1", "Payment method(*)<br>");
echo "<INPUT TYPE=\"radio\" NAME=\"PayMethod\" VALUE=\"Credit\"
CHECKED>";
redFont( "Arial\" size=\"-1", "Credit Card<br>");
echo "<INPUT TYPE=\"radio\" NAME=\"PayMethod\" VALUE=\"Check\">";
redFont( "Arial\" size=\"-1", "Check or Money Order<br>");
echo "<p>";

fontSize("-0","black","Arial","If you pay by credit card:<br>");
//echo "<blockquote>";
fontSize("-1","blue","Arial", "Credit Card type(*)<br>");
echo "<input type=\"text\" name=\"CCType\" size=\"40\"><p>";

```

```

fontSize("-1","blue","Arial", "Credit Card number(*)<br>");
echo "<input type=\"text\" name=\"CCNum\" size=\"40\"><p>";

fontSize("-1","blue","Arial", "Credit Card expiration date(*)<br>");
echo "<input type=\"text\" name=\"CCExpire\" size=\"40\"><p>";
//echo "</blockquote>";

echo "</td><td>";

blueFont( "Arial\" size=\"-1", "Do you need order confirmation by
email?<br>");
echo "<INPUT TYPE=\"radio\" NAME=\"Emailconf\" VALUE=\"No\" CHECKED>";
redFont( "Arial\" size=\"-1", "No<br>");
echo "<INPUT TYPE=\"radio\" NAME=\"Emailconf\" VALUE=\"Yes\">";
redFont( "Arial\" size=\"-1", "Yes<br>");
echo "<p>";

fontSize("-1","blue","Arial","Your address<br>");
echo "<input type=\"text\" name=\"BuyAddress1\" size=\"40\"><p>";
//echo "<input type=\"text\" name=\"BuyAddress2\" size=\"40\"><p>";

fontSize("-1","blue","Arial","Your city and state<br>");
echo "<input type=\"text\" name=\"BuyCity\" size=\"25\">";
echo "&nbsp;&nbsp;&nbsp;<input type=\"text\" name=\"BuyState\" size=\"2\"
MAXLENGTH=\"2\">";
echo "<p>";

fontSize("-1","blue","Arial","Your postal/zip code and country<br>");
echo "<input type=\"text\" name=\"BuyZip\" size=\"7\" maxlength=\"12\">";
echo "&nbsp;&nbsp;&nbsp;<input type=\"text\" name=\"BuyCountry\" size=\"2\"
MAXLENGTH=\"2\"><p>";

fontSize("-1","blue","Arial","Your daytime (work) phone number<br>");
echo "<input type=\"text\" name=\"DayPhone\" size=\"40\"><p>";

fontSize("-1","blue","Arial","Your evening (home) phone number<br>");
echo "<input type=\"text\" name=\"EvePhone\" size=\"40\"><p>";

echo "</td></tr><tr><td align=\"center\" colspan=\"2\">";
$Date=date("Y:z:Y-m-d");
$Total=ereg_replace(",",".",$Total);
echo "<input type=\"hidden\" name=\"OrderTotal\" value=\"$Total\">
<input type=\"hidden\" name=\"Date\" value=\"$Date\">
<input type=\"hidden\" name=\"UserID\" value=\"$UID\">
<input type=\"hidden\" name=\"BuyerID\" value=\"\">
<input type=\"hidden\" name=\"OrderNumber\" value=\"\">
<input type=\"submit\" value=\"Submit\"></form>";

echo "</td></tr></table>";
echo "</center>";

commonFooter($Relative,$SUID);
?>

```

4) Order.php3 code (in root directory)

```
<?
require("bag.php3");
DBInfo();
Root();
Brand();
mysql_connect("$DBHost", "$DBUser", "$DBPass");

if ($SName == "") {
Brand();
commonHeader("$SCompany", "Oops...");
fontSize("-1", "blue", "Arial", "<center>You forgot to tell me your NAME.
Please go back and complete the order form.</center>");
commonFooter($Relative, $UserID);
exit;
}

if ($BuyEmail == "") {
Brand();
commonHeader("$SCompany", "Oops...");
fontSize("-1", "blue", "Arial", "<center>You forgot to tell me your E-MAIL
ADDRESS. Email will be used for shipping the items to you!</center>");
fontSize("-1", "blue", "Arial", "<center><br>Please go back and complete
the order form.</center>");
commonFooter($Relative, $UserID);
exit;
}

if ($PayMethod == "Credit" AND $CCType == "") {
Brand();
commonHeader("$SCompany", "Oops...");
fontSize("-1", "blue", "Arial", "<center>You forgot to tell me your CREDIT
CARD TYPE. Please go back and complete the order form.</center>");
commonFooter($Relative, $UserID);
exit;
}

if ($PayMethod == "Credit" AND $CCNum == "") {
Brand();
commonHeader("$SCompany", "Oops...");
fontSize("-1", "blue", "Arial", "<center>You forgot to tell me your CREDIT
CARD NUMBER. Please go back and complete the order form.</center>");
commonFooter($Relative, $UserID);
exit;
}

if ($PayMethod == "Credit" AND $CCExpire == "") {
Brand();
commonHeader("$SCompany", "Oops...");
fontSize("-1", "blue", "Arial", "<center>You forgot to tell me your CREDIT
CARD EXPIRATION DATE. Please go back and complete the order
form.</center>");
commonFooter($Relative, $UserID);
}
```

```

exit;
}

mysql("SDBName","INSERT INTO Buyers VALUES (
'SName','SBuyAddress1','SBuyAddress2',
'SBuyCity','SBuyState','SBuyZip','SBuyCountry',
'SBuyEmail','SDayPhone','SEvePhone',
'SContact','SPayMethod','SCCType',
'SCCNum','SCCExpire','SOrderTotal',
'SDate','SOrderNumber','SUserID',
'SBuyerID')");

Sresult=mysql("SDBName","SELECT BuyerID FROM Buyers WHERE
UserID='SUserID' AND Date='SDate' AND Name='SName'");
while ($row=mysql_fetch_row($result)) {
SBID=$row[0];
}

mysql("SDBName","INSERT INTO Orders VALUES (
'SBID','SUserID','SDate',
'SOrderTotal','SOrderNumber')");

Sresult=mysql("SDBName","SELECT OrderNumber FROM Orders WHERE
UserID='SUserID' AND Date='SDate' AND BuyerID='SBID'");
while ($row=mysql_fetch_row($result)) {
SONu=$row[0];
}

mysql("SDBName","UPDATE Buyers SET OrderNumber='SONu' WHERE
BuyerID='SBID'");

Sresult=mysql("SDBName","SELECT * FROM BagItems WHERE
UserID='SUserID'");
while ($row = mysql_fetch_row($result)) {
SCUI=$row[0];
SCII=$row[1];
SCIQ=$row[2];
SCDa=$row[3];
SCCI=$row[4];
    Sresult2=mysql("SDBName","SELECT * FROM Items WHERE
ItemID='SCII'");
    while ($row2=mysql_fetch_row($result2)) {
        SIIS=$row2[0];
        SIIN=$row2[1];
        SIID=$row2[2];
        SIIC=$row2[3];
        SICa=$row2[4];
        SIII=$row2[5];

        $To[$i]=((SCIQ * SIIC));
        mysql("SDBName","INSERT INTO Receipts VALUES (
'SBID','SUserID','SONu','SDate','SIII','SCIQ','$To[$i]')");
        $i++;
    }
}

Header("Location:

```

```
$Relative/receipt.php3?BI=$BID&ON=$ONu&PM=$PayMethod&CT=$CCType&Co=$Contact&UID=$UserID");
```

```
mysql("$SDBName","DELETE FROM BagItems WHERE UserID='$UserID'");  
mysql("$SDBName","DELETE FROM Users WHERE User='$UserID'");
```

```
$result=mysql( "$SDBName", "SELECT * FROM Buyers WHERE BuyerID='$BID'");  
while ($row=mysql_fetch_row($result)) {  
  $BNA=$row[0];  
  $BA1=$row[1];  
  $BA2=$row[2];  
  $BCi=$row[3];  
  $BSt=$row[4];  
  $BZi=$row[5];  
  $BEm=$row[6];  
  $BDP=$row[7];  
  $BEP=$row[8];  
  $BCo=$row[9];  
  $BPM=$row[10];  
  $BCT=$row[11];  
  $BCN=$row[12];  
  $BCE=$row[13];  
  $BOT=$row[14];  
  $BDa=$row[15];  
  $BON=$row[16];  
  $BUI=$row[17];  
  $BBI=$row[17];  
}
```

```
if ($BCo == "1") {  
  $BCo="Yes";  
} else {  
  $BCo="No"; }
```

```
$pieces=explode(":", $BDa);
```

```
$part1="$BNA\n$BA1\n$BA2\n$BCi, $BSt $BZi\n$BEm\n\nDaytime Phone:  
$BDP\nEvening Phone: $BEP\n\nPay Via: $BPM\nCC Type: $BCT\nSCC Number:  
$BCN\nCC Expires: $BCE\n\nDate: $pieces[2]\n\n";
```

```
$result=mysql( "$SDBName","SELECT * FROM Receipts WHERE BuyerID='$BID'  
AND OrderNumber='$ONu'");  
while ($row=mysql_fetch_row($result)) {  
  $RBI=$row[0];  
  $RUI=$row[1];  
  $RON=$row[2];  
  $RDa=$row[3];  
  $RII=$row[4];  
  $RIS=$row[5];  
  $RIN=$row[6];  
  $RIC=$row[7];  
  $RIQ=$row[8];  
  $RTC=$row[9];
```

```
$total=$total-$RTC;
```

```

Spart2=Spart2."Authors: SRIS\nTitle: SRIN\nCost: $$RIC\nQuantity:
SRIQ\nTotal Cost: $$RTC\n\n";
}

Spart3="TOTAL: $$total\n\n";

$footer="This order was automatically generated from your web
site.\n\n";

if ($Emailconf=="Yes"){
mail("SBEm", "CINDI Library Web Site Order", "$part1$part2$part3$footer",
"From: SEmail");
}
?>

```

5) viewBag.php3 code (in the root directory)

```

<?
$1=0;
$2=0;
require("bag.php3");
DBinfo();
Brand();
Root();
mysql_connect("$DBHost", "$DBUser", "$DBPass");

if ($SUID != "") {
$result=mysql( "$DBName", "SELECT * FROM Users WHERE User='SUID'");
$num=mysql_num_rows($result);
if ($num == "0") {
$dt=date( "YmdHis");
SUID= "$dt$REMOTE_ADDR";
$dt=date( "z");
mysql( "$DBName", "INSERT INTO Users VALUES ('SUID', '$dt')");
//Header( "Location: $php3_SELF?UID=SUID");
Header("Location: $Relative/viewBag.php3?UID=SUID");
}
}

if ($SUID == "") {
$dt=date( "YmdHis");
SUID= "$dt$REMOTE_ADDR";
$dt=date( "z");
mysql( "$DBName", "INSERT INTO Users VALUES ('SUID', '$dt')");
//Header( "Location: $php3_SELF?UID=SUID");
Header("Location: $Relative/viewBag.php3?UID=SUID");
}

$result=mysql("$DBName", "SELECT * FROM BagItems WHERE UserID='SUID'");
$num=mysql_num_rows($result);
if ($num == "0") {
commonHeader("$Company", "View the contents of your book bag");
blueFont("Arial", "Your book bag is empty <br><br>");
} else {

```

```

commonHeader("$Company","View the contents of your book bag");
blueFont("Arial","Your book bag contains the following
items:<br><br>");

echo "<center><table border='1' cellpadding='5' cellspacing='0'><tr>";
echo "<th>";
redFont("Arial","Authors");
echo "</th>";
echo "<th>";
redFont("Arial","Title");
echo "</th>";
echo "<th>";
redFont("Arial","Price");
echo "</th>";
echo "<th colspan='1'>";
redFont("Arial","Remove Items");
echo "</th></tr>";

$result=mysql("$SDBName","SELECT * FROM BagItems WHERE UserID='SUID'");
while ($row = mysql_fetch_row($result)) {
    $CUI=$row[0];
    $CII=$row[1];
    $CIQ=$row[2];
    $CDa=$row[3];
    $CCI=$row[4];
        $result2=mysql("$SDBName","SELECT * FROM Items WHERE
ItemID='SCII'");
        while ($row2=mysql_fetch_row($result2)) {
            $IIS=$row2[0];
            $IIN=$row2[1];
            $IID=$row2[2];
            $IIC=$row2[3];
            $ICa=$row2[4];
            $III=$row2[5];

$To{$i|}=(($CIQ * $IIC)+($CIQ * $ISC));
echo "<tr><td>";
blueFont("Arial","$IIS");
echo "</td><td>";
blueFont("Arial","$IIN");
echo "</td>";
echo "<td>";
$IIC=number_format($IIC,"2",".", "thousands_sep");
blueFont("Arial","$$IIC");
echo "</td>";

echo "<td><center>";
echo "<FORM ACTION=\"$Relative/removeItem.php3\" METHOD=\"POST\">";
echo "<input type=\"hidden\" name=\"BagItemsID\" value=\"$CCI\">";
echo "<input type=\"hidden\" name=\"UID\" value=\"$SUID\">";
echo "<input type=\"SUBMIT\" NAME=\"SUBMIT\" value=\"remove\">";
echo "</center></td></tr>";
echo "</form>";
$!--;
        }
    }
}

```

```

while ($j < $i) {
    $total=$total+$to[$j];
    $j++;
}

echo "<td colspan='2'>";
redFont("Arial","<div align=right><b><i>Total Cost: </i></b></div>");
echo "<td>";
$total=number_format($total,"2",".", "thousands_sep");
if ($SES == "1") {
    blueFont("Arial","<b><center>$total</b>");
    blueFont("Arial\" size=\"-2\", \" Plus Shipping</center>");
} elseif ($SES != "1") {
    blueFont("Arial","<b><center> $total </center></b>"); }

echo "<td align=center colspan=\"2\">";
echo "<FORM ACTION=\" $Relative/emptyBag.php3\" METHOD=\"POST\">";
echo "<input type=\"hidden\" NAME=\"UID\" value=\"$SUID\">";
echo "<input type=\"SUBMIT\" NAME=\"SUBMIT\" value=\"empty bag\">";
echo "</center></td></tr>";
echo "</table>";
echo "</form>";

echo "<table cellpadding=10><tr><td>";
echo "<FORM ACTION=\" $Relative/index.php3\" METHOD=\"POST\">";
echo "<input type=\"hidden\" NAME=\"UID\" value=\"$SUID\">";
echo "<input type=\"SUBMIT\" NAME=\"Continue\" value=\"<< Continue Shopping\">";
echo "</form>";
echo "</td><td>";

echo "<FORM ACTION=\" $Relative/checkout.php3\" METHOD=\"POST\">";
echo "<input type=\"hidden\" NAME=\"UID\" value=\"$SUID\">";
echo "<input type=\"SUBMIT\" NAME=\"CHECK OUT\" value=\"Check Out >>\">";
echo "</form>";
echo "</td></tr></table>";

}

commonFooter($Relative,$SUID);
?>

```

6) addItem.php3 code (in the /admin directory)

```

<?
require("bag.php3");
Brand();
DBInfo();
Root();
adminHeader("$Company","Add An Item");

```



```

echo "<center><TABLE width=\"50%\" BORDER=\"0\" CELLPADDING=\"4\"
CELLSPACING=\"4\">";
echo "<tr><td>";
fontFace("Arial","Type new item into the boxes, and then SUBMIT!<br>");
echo "</td></tr>";
echo "<FORM ACTION=\"./addItemResponse.php3\" METHOD=\"POST\">";
echo "<tr><td>";
blueFont("Arial","Authors<br>");
echo "<input type=\"text\" name=\"Authors\"
size=\"40\"></td></tr><tr><td>";
blueFont("Arial","Title<br>");
echo "<input type=\"text\" name=\"Title\" size=\"40\"></td></tr>";

echo "<tr><td>";
blueFont("Arial","Item Cost (US$)<br>");
echo "<input type=\"text\" name=\"ItemCost\"
size=\"40\"></td></tr><tr><td>";

echo "<tr><td>";
blueFont("Arial","Select A Category<br>");
echo "<select name=\"Category\" size=\"1\">";
mysql_connect("$DBHost","$DBUser","$DBPass");
$result=mysql("$DBName","SELECT * FROM Category ORDER BY Category");
while ($row = mysql_fetch_row($result)) {
echo "<option value=\"$row[1]\"> $row[0] </option>";
}
echo "</select></td></tr>";

echo "<tr><td>";
blueFont("Arial","Select A Subject<br>");
echo "<select name=\"Subject\" size=\"1\">";
mysql_connect("$DBHost","$DBUser","$DBPass");
$result=mysql("$DBName","SELECT * FROM Subjects ORDER BY Subject");
while ($row = mysql_fetch_row($result)) {
echo "<option value=\"$row[1]\"> $row[0] </option>";
}
echo "</select></td></tr>";

echo "<tr><td>";
blueFont("Arial","Select A Subsubject<br>");
echo "<select name=\"Subsubject\" size=\"1\">";
mysql_connect("$DBHost","$DBUser","$DBPass");
$result=mysql("$DBName","SELECT * FROM Subsubjects ORDER BY
Subsubject");
while ($row = mysql_fetch_row($result)) {
echo "<option value=\"$row[1]\"> $row[0] </option>";
}
echo "</select></td></tr>";

echo "<tr><td colspan=\"2\">";
blueFont("Arial","ItemDescription <br>");
echo "<textarea name=\"ItemDescription\" rows=\"10\" cols=\"50\"
wrap></textarea></td></tr>";

echo "<input type=\"hidden\" name=\"ItemID\" value=\"\">";

```

```

echo "<tr><td colspan=\"2\" align=\"center\"><INPUT TYPE=\"submit\"
NAME=\"Submit\" VALUE=\"Submit\"></form></TABLE></center>";

adminFooter($Relative);
?>

```

7) removeItem.php3 code (in the /admin directory)

```

<?
require("bag.php3");
Brand();
DBInfo();
Root();
adminHeader("$Company", "Remove An Item");

echo "<center><TABLE width=\"50%\" BORDER=\"0\" CELLPADDING=\"4\"
CELLSPACING=\"4\"><TR>";
echo "<FORM ACTION=\"./removeItem1.php3\" METHOD=\"POST\">";
echo "<tr><td colspan=\"2\">";
fontFace("Arial", "<br><b>Fill the Title and/or Authors of the item to
remove, and then SUBMIT!</b><br><br>");
echo "</td></tr>";
echo "<tr><td>";
blueFont("Arial", "Title");
echo "</td><td>";
echo "<input type='text' name='Title' size='60' maxlength='150' >";
echo "</td></tr>";
echo "<tr><td>";
blueFont("Arial", "Authors");
echo "</td><td>";
echo "<input type='text' name='Authors' size='60' maxlength='150' >";
echo "</td></tr>";

echo "<tr><td>&nbsp;   </td><td align=\"center\"><br><INPUT
TYPE=\"submit\" NAME=\"Submit\"
VALUE=\"Submit\"></form></TABLE></center>";

adminFooter($Relative);
?>

```

8) removeItem1.php3 code (in /admin directory)

```

<?
require("bag.php3");
Brand();
DBInfo();
Root();
adminHeader("$Company", "Remove An Item");

if ($Title != "" and $Authors != ""){
$SearchItem = "Title=\"$Title\" and Authors=\"$Authors\"";
}
if ($Title != "" and $Authors == ""){
$SearchItem = "Title=\"$Title\"";
}
if ($Title == "" and $Authors != ""){
$SearchItem = "Authors=\"$Authors\"";
}

```

```

}
if ($Title == "" and $Authors == "") {
Brand();
//commonHeader("$Company","Oops...");
fontSize("-1","blue","Arial","<center>You forgot to fill the item.
Please go back and complete the form.</center>");
commonFooter($Relative,$UserID);
exit;
}

mysql_connect("$SDBHost","$SDBUser","$SDBPass");
$result=mysql("$SDBName","SELECT * FROM Items WHERE $SearchItem ");
while($row=mysql_fetch_row($result)) {
$IS=$row[0];
$IN=$row[1];
$ID=$row[2];
$IC=$row[3];
//SCa=$row[4];
$II=$row[5];
}
echo "<center><TABLE width=\"50%\" BORDER=\"0\" CELLPADDING=\"4\"
CELLSPACING=\"4\"><TR>";
echo "<FORM ACTION=\"./removeItem2.php\" METHOD=\"POST\">";
echo "<tr><td>";
fontFace("Arial","Is this item is what you want to delete? Yes, SUBMIT!
No, Back.<br><br>");
echo "</td></tr>";
echo "<tr><td>";
blueFont("Arial","Authors<br>");
echo "<input type='text' name=\"Authors\" size=\"40\"
value=\"$IS\"></td></tr><tr><td>";
blueFont("Arial","Title<br>");
echo "<input type=\"text\" name=\"Title\" size=\"40\"
value=\"$IN\"></td></tr>";

echo "<tr><td>";
blueFont("Arial","Cost<br>");
if ($IC == "0.00") {
$IC = ""; }
echo "<input type=\"text\" name=\"ItemCost\" size=\"40\"
value=\"$IC\"></td></tr><tr><td>";
echo "<tr><td>";
echo "</tr>";

echo "<tr><td colspan='2'>";
blueFont("Arial","ItemDescription<br>");
$ID=ereg_replace("\"","&quot;",$ID);
echo "<textarea name=\"ItemDescription\" rows=\"10\" cols=\"50\"
wrap>$ID</textarea></td></tr>";
echo "<input type=\"hidden\" name=\"ItemID\" value=\"$II\">";
echo "<tr><td colspan=\"2\" align=\"center\"><INPUT TYPE=\"submit\"
NAME=\"Submit\" VALUE=\"Submit\"></form></TABLE></center>";

adminFooter($Relative);
?>

```

9) removeItem2.php3 code (in the /admin directory)

```
<?
require("bag.php3");
Brand();
DBInfo();
Root();
adminHeader("$Company","Remove An Item");

echo "<center><TABLE width=\"60%\" BORDER=\"0\" CELLPADDING=\"10\"
CELLSPACING=\"10\"><TR>";
echo "<tr><td align=\"center\">";
mysql_connect("$DBHost","$DBUser","$DBPass");
$result=mysql("$DBName","SELECT ItemID,Title FROM Items WHERE
ItemID='$ItemID'");
while ($row = mysql_fetch_row($result)) {
$II=$row[0];
$IN=$row[1];
}

blueFont("Arial","<h3>Are you sure you want to delete this item</h3>");
redFont("Arial","<h2>$IN</h2><p>");
echo "<form action=\"removeItemResponse.php3\" method=\"post\">";
echo "<input type=\"hidden\" name=\"ItemID\" value=\"$II\">";
echo "<INPUT TYPE=\"submit\" VALUE=\"YES\"></form><p>";
echo "<form action=\"$Relative/admin/index.php3\" method=\"post\">";
echo "<INPUT TYPE=\"submit\" VALUE=\"NO\"></form><p>";
echo "</td></tr></table></center>";

adminFooter($Relative);
?>
```

10) removeItemResponse.php3 code (in the /admin directory)

```
<?
require("bag.php3");
DBInfo();
Root();
mysql_connect("$DBHost","$DBUser","$DBPass");
mysql("$DBName","DELETE FROM Items WHERE ItemID='$ItemID'");
exec("rm $WebRoot/images/$ItemID.jpg");
Header("Location: ../searchReceipts.php3");
?>
```

11) searchReceipt.php3 code (in the /admin directory)

```
<?
require("bag.php3");
Brand();
Root();
adminHeader("$Company","Find A Receipt");

echo "<center><TABLE width=\"80%\" BORDER='0' CELLPADDING='10'
CELLSPACING='10'>";
echo "<FORM ACTION='../searchReceiptsResponse.php3?criteria=order'
METHOD='POST'>";
echo "<tr><td colspan=\"3\">";
fontFace("Arial","<b>Choose your search method...</b>");
echo "</td><tr>";
```

```

echo "<tr><td>";
blueFont("Arial","By Order Number:");
echo "</td><td>";
echo "<input type='text' name='OrderNumber' size='50'>";
echo "</td><td>";
echo "<INPUT TYPE='submit' NAME='Submit' VALUE='Submit'>";
echo "</td></tr></form>";

echo "<FORM ACTION='./searchReceiptsResponse.php3?criteria=name'
METHOD='POST' ENCTYPE='x-www-form-urlencoded'>";
echo "<tr><td>";
blueFont("Arial","By User's Name:");
echo "</td><td>";
echo "<input type='text' name='User' size='50'>";
echo "</td><td>";
echo "<INPUT TYPE='submit' NAME='Submit' VALUE='Submit'>";
echo "</td></tr></form>";

SMn=date("m");
SLMn=date("F");
SDy=date("d");
SYr=date("Y");
echo "<FORM ACTION='./searchReceiptsResponse.php3?criteria=date'
METHOD='POST' ENCTYPE='x-www-form-urlencoded'>";
echo "<tr><td>";
blueFont("Arial","By Date:");
echo "</td><td>";
echo "<select name='Month'>";
echo "<option value='SMn'> SLMn </option>";
echo "<option value=''> Any Month </option>";
echo "<option value='01'> January </option>";
echo "<option value='02'> February </option>";
echo "<option value='03'> March </option>";
echo "<option value='04'> April </option>";
echo "<option value='05'> May </option>";
echo "<option value='06'> June </option>";
echo "<option value='07'> July </option>";
echo "<option value='08'> August </option>";
echo "<option value='09'> September </option>";
echo "<option value='10'> October </option>";
echo "<option value='11'> November </option>";
echo "<option value='12'> December </option>";
echo "</select>";
echo "&nbsp; &nbsp; &nbsp; ";
echo "<select name='Day'>";
echo "<option value='SDy'> SDy </option>";
echo "<option value=''> Any Day </option>";
echo "<option value='01'> 01 </option>";
echo "<option value='02'> 02 </option>";
echo "<option value='03'> 03 </option>";
echo "<option value='04'> 04 </option>";
echo "<option value='05'> 05 </option>";
echo "<option value='06'> 06 </option>";
echo "<option value='07'> 07 </option>";
echo "<option value='08'> 08 </option>";
echo "<option value='09'> 09 </option>";
echo "<option value='10'> 10 </option>";

```

```

echo "<option value='11'> 11 </option>";
echo "<option value='12'> 12 </option>";
echo "<option value='13'> 13 </option>";
echo "<option value='14'> 14 </option>";
echo "<option value='15'> 15 </option>";
echo "<option value='16'> 16 </option>";
echo "<option value='17'> 17 </option>";
echo "<option value='18'> 18 </option>";
echo "<option value='19'> 19 </option>";
echo "<option value='20'> 20 </option>";
echo "<option value='21'> 21 </option>";
echo "<option value='22'> 22 </option>";
echo "<option value='23'> 23 </option>";
echo "<option value='24'> 24 </option>";
echo "<option value='25'> 25 </option>";
echo "<option value='26'> 26 </option>";
echo "<option value='27'> 27 </option>";
echo "<option value='28'> 28 </option>";
echo "<option value='29'> 29 </option>";
echo "<option value='30'> 30 </option>";
echo "<option value='31'> 31 </option>";
echo "</select>";
echo "&nbsp; &nbsp; &nbsp; ";

Sd5=Yr-5;
Sd4=Yr-4;
Sd3=Yr-3;
Sd2=Yr-2;
Sd1=Yr-1;

echo "<select name='Year'>";
echo "<option value='Yr'> Yr </option>";
echo "<option value=''> Any Year </option>";
echo "<option value='Sd1'> Sd1 </option>";
echo "<option value='Sd2'> Sd2 </option>";
echo "<option value='Sd3'> Sd3 </option>";
echo "<option value='Sd4'> Sd4 </option>";
echo "<option value='Sd5'> Sd5 </option>";
echo "</select>";
echo "</td><td>";
echo "<INPUT TYPE='submit' NAME='Submit' VALUE='Submit'>";
echo "</td></tr></form></TABLE></center>";
adminFooter($Relative);
?>

```

9. References

- [1] Jesus Castagnetto et al, *Professional PHP Programming*, Wrox Press Ltd., Birmingham, 1999.
- [2] Ian Sommerville, *Software Engineering*, Addison-Wesley Publishing Company, 1996.
- [3] Julie Meloni, *Authenticate and track users with PHP*,
<http://hotwired.lycos.com/webmonkey/templates/print-template.html>.
- [4] CINDI system, <http://cindi.cs.concordia.ca/>
- [5] PHP official site, *manual and introduction*, <http://www.php.net>
- [6] *MySQL introduction and manual*, <http://www.mysql.com>
- [7] *Apache introduction and manual*, <http://www.apache.org>
- [8] *Linux introduction*, <http://www.linux.com>
- [9] dack.com, *Best practices for designing shopping cart and checkout interface*,
http://www.dack.com/web/shopping_cart_print.html
- [10] Open Source Initiative, open source official site, <http://www.opensource.org>
- [11] <http://www.informit.com/>
- [12] <http://www.asptoday.com/>
- [13] <http://www6.163.com/it/item/0,2186,5091,00.html>
- [14] iiscart, <http://www.iiscart.com/>
- [15] salescart shopping cart, <http://www.salescart.com/>
- [16] easycart, <http://easycart.com/>
- [17] eastland hybrid, <http://www.eastland.com/>
- [18] eastland Java, <http://www.eastland.com/>
- [19] eastland ColdFusion, <http://www.eastland.com/>
- [20] Dansie Shopping Cart, <http://www.cambist.com/dansiecart.html>
- [21] <http://www.amazon.com/>
- [22] <http://www.ebay.com/>
- [23] *Netcraft survey*, <http://www.netcraft.com/survey>
- [24] *Shopping cart review criteria* <http://sellitontheweb.com/ezine/criteria2.shtml>
- [25] PHP-based User authentication, <http://www.zend.com/zend/tut/authentication.php>
- [26] <http://www.phpbuilder.com/forum/read.php3?num=1&id=101145&thread=101127>

- [27] Online bookstore Chapters, <http://www.chapters.ca>
- [28] Internet Operating System Counter, <http://www.leb.net/hzo/ioscount/index.html>
- [29] Internet Cookie Report, May 1, 2001,
http://www.securityspace.com/s_survey/data/man.200104/cookieReport.html
- [30] The Cookie Concept, <http://www.cookiecentral.com/>
- [31] The Energy Department's Computer Incident Advisory Capability (CIAC),
Internet Cookies, <http://www.ciac.org/ciac/bulletins/i-034.shtml>
- [32] Cookie Worries Are Unfounded, U.S. Government Says,
<http://www.techweb.com/wire/story/TWB19980316S0015>
- [33] Cookie and Privacy, <http://privacy.net/cookies/>
- [34] HTAccess Authentication , <http://faq.clever.net/htaccess.htm>
- [35] Configure the Web server to use authentication and encryption technologies,
where required, <http://www.cert.org/security-improvement/practices/p080.html>