

# Online Bookstore

By

Priya Shah

Submitted to  
The Faculty of the Information Engineering Technology Program  
In Partial Fulfillment of the Requirements  
For  
The degree of Bachelor of Science  
In Information Engineering Technology

3/14/2000

Date

Masit

Date

3/14/2000

Date

University of Cincinnati  
OMI College of Applied Science

March 2000

# Online Bookstore

By

Priya Shah

Submitted to

The Faculty of the Information Engineering Technology Program  
In Partial Fulfillment of the Requirements

For

The Degree of Bachelor of Science  
In Information Engineering Technology

© Copyright 2000 Priya Shah

The author grants to the Information Engineering Technology Program permission to reproduce and distribute copies of this document in whole or part.

Priya Shah

Author: Priya Shah

3/14/2000

Date

Robert E. Schlemmer

Faculty Project Advisor: Prof. Robert Schlemmer

Mar. 14, 2000

Date

Kim Myers

Department Head: Dr. Kim Myers

3/14/2000

Date

## Dedication

I would like to thank my advisor, Prof. Robert Schlemmer for his expert guidance and encouragement during my senior design project work at the University of Cincinnati. I would also like to thank Dr. Ashraf Saad for introducing me to the wonderful world of Web development. In addition, I would like to express my deep gratitude to Dr. Sam Geonetta for exchanging ideas and providing support and encouragement through my senior design project and undergraduate studies. I consider myself fortunate to be associated with these professors.

This senior design project would not have been possible without unselfish help rendered to me by my friends Robert Brown, Brian Dietrick, and Sean Snider in developing the code. I would also like to acknowledge my friends Mark Hoar, Jen Fritz, Ashika Wijesekera, and Jason Stroh for their valuable suggestions and moral support throughout the project. I thank them all for all their help.

Also I would like to give special thanks to my husband Pranav Shah for his continued support and encouragement throughout my undergraduate studies.

Finally, I would like to thank all the people associated with the University of Cincinnati who have provided me continued support during my undergraduate degree.

# Table of Contents

<b>Section</b>	<b>Page</b>
Dedication	i
Table of Contents	ii
List of Figures	iii
Abstract	iv
1. Statement of the Problem	1
1.1 Definition of the Need	1
2. Review of the Literature.	3
3. Description of the solution.	9
a. User Profile.	9
b. Design Protocols.	9
4. Objectives of the project (“Deliverables”).	14
5. Design and Development.	15
a. Budget	15
b. Timeline	16
c. Software	17
d. Hardware	17
6. Proof of design	18
7. Conclusions and recommendations.	53
8. Notes	58
9. References	61

# List of Figures

Figure	Page
1. Basic architecture of the project	18
2. Database Diagram	19
3. Default.htm	22
4. BookSearch.asp	24
5. SearchResult.asp	26
6. BookDetails.asp	28
7. BookQuantity.asp	31
8. CartItems.asp	34
9. Page1.asp	37
10. Newmember.htm	38
11. NewLogin.asp	39
12. Pre-Submit.asp	41
13. PreviewOrder.asp	44
14. FinalOrder.asp	46
15. CancelOrder.htm	49
16. CancelConfirm.asp	50
17. Help.htm	52

# Abstract

The following report covers the design criteria needed to develop my senior design project, Online Bookstore. The purpose of this project was to create a Web site through which the users can place online orders for books.

The material covered within demonstrates for the reader how to search and select books and place order for selected books using the Web site. To aid faster searches, the categories of the books are also listed.

# Online Bookstore

## 1. Statement of the Problem

I was interested in doing a project that covered the subjects of Internet Technology and Database Management skills. The project involved creating and hosting a Web site that is used to make online purchases of books. Design and development of the Web site and database model are the integral parts of the project.

I did not do this project to compete with existing online bookstore such as AMAZON.COM. I did this project because it is in the new, exciting field of Electronic Commerce (E-Commerce). The term E-Commerce means selling products or services on the Web. This field has high growth potential. A lot of businesses are going online, with great job opportunities in this field. By doing this project, I have acquired skills like Internet Programming, implementing a SQL Server database, Web site development, creating Active Server Pages, and HTML. Professionals with these skills are highly sought after in the job market today. I want to be a part of this E-Commerce field.

### 1.1 Definition of the Need

My experience as an online bookstore customer has shown that one can do things online that one cannot do in person. Online bookstores offer more products, better service, and useful features unheard of in the “real” world. Because of the interactive nature of online bookstores, customers can jump around the bookstore using simple searches that take seconds to complete. Online bookstores are cost effective and time saving.

2. In my online bookstore, the user can search for books by providing the Title, Category or ISBN. When the user searches for books by providing the category, all the books under that category are listed for him. All the information he needs to know to make a wise decision of purchasing specific books is available to him by pressing a few keys in few seconds. This saves him time, energy and saves him from standing in long lines for checkout at the traditional bookstores.

## 2. Review of the literature

Online Bookstore project is within the field of E-Commerce. E-commerce is selling products or services on the Internet. Many companies are entering the world of e-commerce because the increasing number of Web surfers, the promise of lower sales cost and the feedback from target markets attracts them. Revenues from online shopping are expected to increase dramatically over the next few years. The annual online sales in year 2000 are expected to approach \$6.6 billion. (34)

Some companies are hosting independent Web sites that supplement traditional catalogs or other sales efforts. Others are joining Internet malls that include a collection of online storefronts that are accessible through a single site.

Companies entering the e-commerce world move through the same three stages: hosting their Web site, adding graphics and interactivity to incorporate continual customer input and finally tailoring their marketing efforts to specific customers. To increase the sales through these Web sites the companies have to constantly update them, offering incentives such as lower pricing or try-before-you-buy offers. Since most purchases are made using a credit card, security by encrypting the data has to be provided to assure the customers.

A corporate e-business model and a supporting IT architecture are critical factors to the success of any e-commerce initiative. IT architecture should be secure, scalable, robust, and adaptable to ever-changing business and IT environments. The following are the four phases in organization's e-commerce development cycle.

- 1) Hosting Web site and publishing company information, products, and services.
- 2) Link the site to Web enabled applications and give back-end support.

3) Provide traditional transactional capabilities.

4) Mine the customer data.(15)

With the use of e-commerce, the companies can target more number of potential customers, lower customer service costs, and increase sales. Nick Wreden, the author of the article, "E-commerce: A Marketing Revolution," (34) says, "It bypasses the constraints of paper-based promotional efforts." Because of e-commerce the information about the various products and services is available at one's fingertips, offering greater convenience to customers.

To attract the customers these e-commerce sites have to be user friendly and attractive. Security is the key factor in e-commerce as most purchases are made using credit cards. In his article, "E-commerce: Report From The Trenches," Ron Higgins states "Security must be an integral part of the design phase of every e-commerce solution. It is much easier to build in security from the beginning than to retrofit it into an existing solution." The Web sites should be robust in structure to withstand a higher number of hits. As competitors are literally a mouse click away, the availability of the Web sites is very important. To fulfill increasing customer demands the Web sites should be scalable. As Higgins says, " The majority of classic applications are built on time-tested transaction and database subsystems. These provide the transactional context required to maintain the integrity of a business's most precious asset – it's data. The security, availability and scalability connected with classic applications make them prime candidates for jump-starting e-commerce initiative." (15)

E-business model's IT architecture is critical to the success of any e-commerce initiative. The information technology manager has to ensure that the final IT architecture is secure, scalable and adaptable to ever-changing business and IT environments.

Every E-commerce Web site should have a good user interface to attract the customers. While designing the interface the designer has to keep in mind that it has a consistent theme and navigation incorporated throughout the site. This allows the customers/users to create a "mental mode" for the site. (32) The colors used for the site should be pleasant to the eye. Developing the Web site considering yourself as a user will always help during the development process. Rosenfeld and Morville say "Drawing on your sensitivity to web consumer's needs will make a great difference as you start designing and building your web site. Reaching such a level of user-centered awareness sets you aside from every other web developer; in a profession with such a low barrier of entry, it may be all you have to ensure that your work stands out". Developers should always keep in mind the technical and "Look and Feel" issues during the Web interface development. The technical issues can be the logical interactivity and bandwidth/download issues. "Look and Feel" issues consist of aesthetics, functionality, and good copyediting. When using images on the Web site the download time of images is an important consideration. Utility of the Web site should be the prime consideration during the development process. (25, 2)

Every E-Commerce project requires a strong database back-end. The thing that makes each of these sites work is a relational database. Since these sites have to handle a tremendous number of hits, they use big, expensive RDBMSs (relational database

management systems). The database is used to store the customers' data, the products' data, orders, etc. The back-end is the major part of any E-commerce application. The names for the RDBMSs they use are familiar: Oracle, Sybase, Microsoft SQL Server, and Informix. (13) Conceptually, these programs offer a grab bag of features and technologies that synergize nicely:

- They store data in *tables*. Tables are remarkably similar to fixed-width text files, but their fields (also known as "columns") can contain many different structured data types. Examples are integer, character string, money, date, and Binary Large Object (BLOB).
- They provide administrative facilities to manage tables.
- Tables and administrative facilities are guarded through sophisticated user/password/domain protection schemes.
- A user can interact with data through powerful-yet-relatively-easy languages, SQL for example. Even better, a user can build stored SQL statements that he can take advantage of without having to know the language himself.

A final advantage of databases is that so many people before you have used them; virtually all seed data will already be in some kind of database format. (10) In addition to being relatively easy to create and access, a relational database has the important advantage of being easy to extend. After the original database creation, a new data category can be added without requiring that all existing applications be modified.

Relational database is a set of tables containing data fitted into predefined categories. Each table (which is sometimes called a *relation*) contains one or more data categories in columns. Each row contains a unique instance of data for the categories defined by the columns. For example, a typical business order entry database would include a table that described a customer with columns for name, address, phone number, and so on. Another table would describe an order: product, customer, date, sales price, and so on. A user of

the database could obtain a *view* of the database that fitted the user's needs. For example, a branch office manager might like a view or report on all customers that had bought products after a certain date. A financial services manager in the same company could, from the same tables, obtain a report on accounts that needed to be paid. (24) A user can write stored procedures for faster data access. To dynamically access the data from the Web pages, Active Server Pages (ASP) are created.

Active Server Pages, commonly referred to as ASPs, is Microsoft's solution to server-side scripting. With simple HTML pages, the client (a Web surfer) requests a Web page from a server. The server sends the file to the client, and the page is shown on the client's browser. With Active Server Pages, the server gets a chance to alter the file before sending it to the user. So, for every request for a file with an .ASP extension, the server runs the file through a DLL called ASP.DLL, which parses the ASP commands. A user can have his ASP code connect to a database (SQL, Access, Oracle, Informix, or any ODBC-compliant database) and dynamically insert the data into his HTML pages. This leads to some very powerful possibilities including E-Commerce, customizable sites, data entering / retrieving systems run over the Internet, and a number of other possibilities. ASP integrates well with Windows NT and it is relatively easy to create a database-driven site with the combination of Microsoft developed tools to develop ASP pages. (14)

ActiveX Data Object (ADO) code can also be used to access the database from the Web pages. ADO is an application program interface (API) from Microsoft that lets programmers writing Windows applications get access to relational and nonrelational databases from both Microsoft and other database providers. For example, if a user wanted to write a program that would provide users of a Web site with data from a MS

SQL Server database or an Oracle database, he could include ADO program statements in an HTML file that he then identified as an Active Server Page (ASP). When a user requested the page from the Web site, the page sent back could include appropriate data from a database, obtained using ADO code.

Like Microsoft's other system interfaces, ADO is an object-oriented interface. It is also part of an overall data access strategy from Microsoft called Universal Data Access (UDA). Microsoft says that rather than try to build a universal database as IBM and Oracle have suggested, why not provide universal access to various kinds of existing and future databases? In order for this to work, Microsoft and other database companies provide a "bridge" program between the database and Microsoft's OLE DB, the low-level interface to databases. OLE DB is the underlying system service that a programmer using ADO is actually using. A feature of ADO, Remote Data Service, supports "data-aware" ActiveX controls in Web pages and efficient client-side caching. As part of ActiveX, ADO is also part of Microsoft's overall Component Object Model (COM), its component-oriented framework for putting programs together.

### 3. Description of the solution

#### 3.a User profile

The targeted users of this online bookstore will be moderately experienced computer users and Web browsers. Users are comfortable with logically structured Web site that is easy to use. I have created a Web site that is easy to use, has minimum complexity, and ease of navigation. Users will benefit from the online help provided for each step in the process of ordering books.

#### 3.b Design Protocols

Screen 1 – Figure 3 on page 22

This orientation screen welcomes users to the Online Bookstore Web site. It has navigation buttons and an image map to link to the key pages (Help, Book Search, Shopping Cart, and Order Cancellation). This screen is the “Home” screen for the site.

Screen 2 - Figure 4 on page 24

This is the search screen. The user is allowed to search books according to the Title, Category and ISBN. Text input fields (for *Title*, *ISBN*, and *Category*) and a *Submit* command button are on this screen. After the user enters data in one or more text fields, he has to press the *Submit* button to advance to the next screen.

### Screen 3 - Figure 5 on page 26

Based on the data entered in the text fields in Screen 2, the *Books* table in the SQL Server database is accessed and the related records are retrieved and displayed. A hyperlink<sup>1</sup> is provided for each retrieved record. The user can click on a link and see the details about the particular book.

### Screen 4 - Figure 6 on page 28

When the user clicks on the link for a particular book, its details are displayed on this screen. There are two buttons provided on this screen:

- 1) *Add to Cart* - After pressing the *Add to Cart* button, the book gets added to the shopping cart and the next screen is displayed.
- 2) *Book Search* - After pressing this button, the user is taken back to Screen 2, the Book Search screen to search for more books.

### Screen 5 - Figure 7 on page 31

On this screen the title and price of the selected book are displayed. Then the user is asked to enter the quantity for the selected book. There is a *Submit* button on this screen, when it is pressed it takes you to the next screen.

### Screen 6 - Figure 8 on page 34

This is the Shopping Cart items screen. The newly added book is displayed in the list of selected books. If the user has already added book(s) to the shopping cart then these books are also displayed. The amount (price x quantity) for each selected book item is displayed. To change the quantity of selected book item, the user has to select the book

---

<sup>1</sup> Please check the detailed information about all the underlined phrases in the Notes section.

again by going through book search screen and enter the new quantity. This updates the selected quantity for the specific book. There are two buttons provided on the screen:

- 1) *Book Search* - After pressing this button, the user is taken back to Screen 2, the Book Search screen to search more books.
- 2) *Proceed to Checkout* - After pressing this button, the next screen is displayed.

#### Screen 7 - Figure 9 on page 37

It is a login screen. The screen displays input text fields for *User ID* and *password*.

There are two options on this screen:

- 1) *Submit* – After pressing this button, the user is taken to Screen 8.
- 2) *New Member*– When clicked on this hyperlink, the user is taken to *New Login* screen.

#### New Login Screen - Figure 10 on page 38

This is the screen to create a new login account for a new user. The screen displays input fields for *UserID* and *Password*. There is a *Submit* button on this screen. When pressed it takes the user to the Login Confirmation screen.

#### Login Confirmation - Figure 11 on page 39

If the entered UserID is unique, i.e., not been used already, a message is displayed that says that new UserID is created. And the user is asked to continue by clicking the *Next* button to advance to the next screen, Screen 8. If the user has entered an existing UserID then he is asked to go to the previous screen to enter a new UserID.

#### Screen 8 - Figure 12 on page 41

On this screen there are fields for *E-mail*, *Address*, *Credit Card Number*, *Type of Card*, and *Card Expiry Date*. For a returning customer, data for these fields will be displayed which is retrieved from his history stored in the database. He gets to edit this information. The new customer will have to enter this data. All the customers will have to select a shipping method from the list of shipping methods displayed in the list box. There is a shipping method table provided to refer to the shipping costs. Finally, the customer has to press the *Submit* button to review his order.

#### Screen 9 - Figure 13 on page 44

On this screen, the books amount, shipping cost, and total amount for the order are displayed to the user. There is *Submit* button on this screen. When it is pressed the user is taken to the next screen.

#### Screen 10 - Figure 14 on page 46

On this screen the order number is displayed for the user.

#### Order Cancellation Screen - Figure 15 on page 49

On this screen, an input text field is provided to enter the order number for the order that the user wishes to cancel and a *Submit* button. When the user clicks this button, he is taken to the Order Cancel Confirmation screen.

### Order Cancel Confirmation - Figure 16 on page 50

If the user has entered an existing order number in the previous screen, a order cancellation confirmation is displayed to him. If he has entered a non-existing order number then he is asked to go to the previous screen to enter a correct order number.

### Help - Figure 17 on page 52

This screen will provide information to the user about how to search for books, place purchase orders, and cancel placed orders.

## **4. Objectives of the project (“Deliverables”)**

The following were the completion objectives:

- An interactive Web site
- An enhanced user interface
- Secured connection
- Effective database design
- Customer login
- Category wise book search
- Title wise book search
- ISBN wise book search
- Online purchase order
- Order cancellation

## 5. Design and Development

### 5.a. Budget

Hardware requirements:

Cost

<u>Web Server</u>	\$ 4,000
<u>Database Server</u>	\$ 4,000
Pentium III Processor 500 or above	\$ 250

Software requirements

<u>Visual InterDev 6.0</u>	\$ 550
<u>MS SQL Server 7.0</u>	\$ 978
<u>Internet Information Server 4.0</u>	Free with Windows NT
MS Word 97/2000	\$ 170
Windows NT Server and Workstation	\$ 13 each
Internet Explorer 4.X or above	Free

The software and hardware resources required for the completion of the project are available in the College of Applied Science's facilities.

## **5.b Timeline**

### **Senior Design 1**

Literary review

Research

### **Senior Design 2**

Finalized database design

Finalized interface design

Mastered MS SQL Server 7

Mastered MS Visual InterDev 6

Built skeleton of the project

Demonstrated Working Quick Prototype

### **Autumn Quarter**

Further research

Started building functionality

### **Senior Design 3**

Completed coding

Completed Testing/Troubleshooting

Documented the project

### **March 3**

Submit final project

Submit final documentation

### **Final's Week**

Demonstrate final project

## **Throughout the project**

Gathered information

Built bibliography

### **5.c Software**

The software requirements for the project are:

- Visual InterDev 6.0
- MS SQL Server 7.0
- Internet Information Server 4.0
- MS Word 97/2000
- Windows NT Server and Workstation
- Internet Explorer 4.X or above

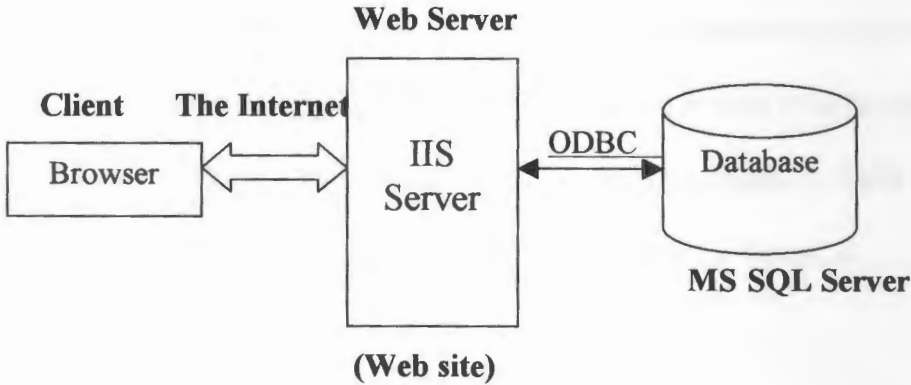
### **5.d Hardware**

The hardware requirements for the project are:

- Web Server
- Database Server
- Client PC

## 6. Proof of design

The basic architecture of the project consists of three components client (browser), Web server, and the database. The basic architecture for the project is displayed in Figure 1 below.



**Figure 1. Basic Architecture of the Project**

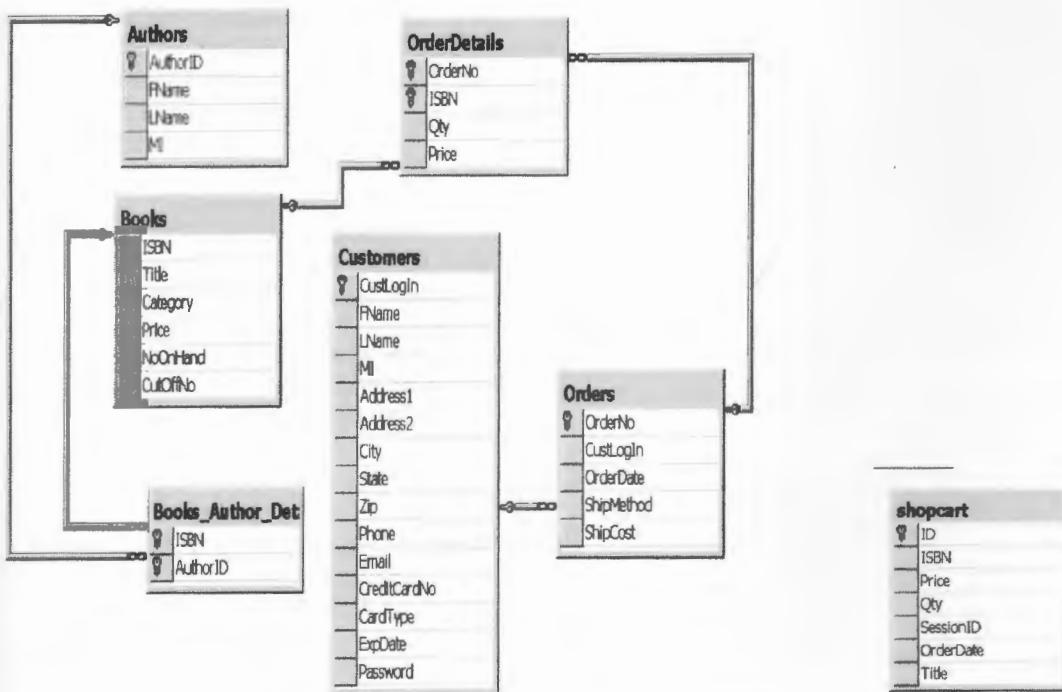
This project is best viewed using the Internet Explorer (4.X or above) browser on the client. The Web site is hosted on the Web server. The Web site consists of various HTML and Active Server Pages. When a user enters the URL of the project in his browser, it sends a request to the server, which then fetches the Deafault.html page and sends it to his browser.

Internet Information Server (IIS) 4.0 is installed on the Web server. IIS is Microsoft's Web server that runs on Windows NT platforms. It comes bundled with Windows NT 4.0. Because IIS is tightly integrated with the operating system, it is relatively easy to administer.

This project, as an E-commerce project, required a strong database to support the application. Open DataBase Connectivity is setup between the Web server and the Database server. The goal of ODBC is to make it possible to access any data from any

application, regardless of which database management system (DBMS) is handling the data. ODBC manages this by inserting a middle layer, called a database driver, between an application and the DBMS. The purpose of this layer is to translate the application's data queries into commands that the DBMS understands.

I have created a SQL Server database “shah”, which is hosted on a SQL Server. After normalizing the data, I created seven tables to store the relevant data in support of the application. Primary keys and foreign keys have been set up in each table to enforce referential integrity. There are a number of stored procedures created to build the functionality of the project. The database diagram can be seen in Figure 2.



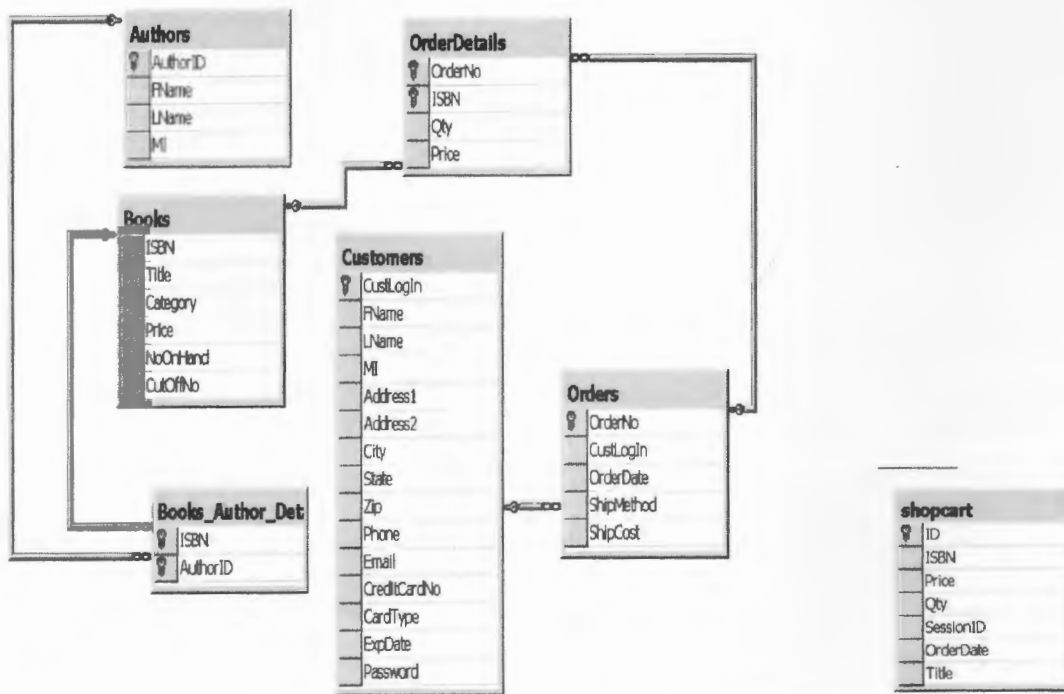
**Figure 2. Database Diagram**

The tables created in the database are:

- Books – This table stores information about all the books in the bookstore.

application, regardless of which database management system (DBMS) is handling the data. ODBC manages this by inserting a middle layer, called a database driver, between an application and the DBMS. The purpose of this layer is to translate the application's data queries into commands that the DBMS understands.

I have created a SQL Server database "shah", which is hosted on a SQL Server. After normalizing the data, I created seven tables to store the relevant data in support of the application. Primary keys and foreign keys have been set up in each table to enforce referential integrity. There are a number of stored procedures created to build the functionality of the project. The database diagram can be seen in Figure 2.



**Figure 2. Database Diagram**

The tables created in the database are:

- **Books** – This table stores information about all the books in the bookstore.

- Authors – This is the master table where names and information about authors is saved.
- Books\_Author\_Details – As the number of authors for each book is not constant I had to create Many-to-Many relationship between Books and Authors table. To keep track of books with their authors, I have created this table.
- Customers – This table stores customers' information.
- Orders – This table stores information about all the orders placed.
- OrderDetails – As one book can be on multiple orders and each order can have multiple books, Many-to-Many relationship had to be created between the tables Books and Orders. By creating OrderDetails table this relationship has been established.
- Shopcart – This is a temporary table that stores information about the user's shopping cart items. After the order is placed, the data in this temporary table is destroyed.

Following code can be used to create the tables required for this project:

```
CREATE TABLE [dbo].[Authors] (
    [AuthorID] [int] IDENTITY (1, 1) NOT NULL ,
    [FName] [varchar] (15) NOT NULL ,
    [LName] [varchar] (20) NOT NULL ,
    [MI] [char] (1) NULL
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[Books] (
    [ISBN] [char] (10) NOT NULL ,
    [Title] [varchar] (30) NOT NULL ,
    [Category] [varchar] (15) NOT NULL ,
    [Price] [smallmoney] NOT NULL ,
    [NoOnHand] [tinyint] NOT NULL ,
    [CutOffNo] [tinyint] NOT NULL
)
GO
```

```
CREATE TABLE [dbo].[Books_Author_Details] (
    [ISBN] [char] (10) NOT NULL ,
    [AuthorID] [int] NOT NULL
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[Customers] (
    [CustLogIn] [varchar] (10) NOT NULL ,
    [FName] [varchar] (15) NULL ,
    [LName] [varchar] (20) NULL ,
    [MI] [char] (1) NULL ,
    [Address1] [varchar] (25) NULL ,
    [Address2] [varchar] (25) NULL ,
    [City] [varchar] (15) NULL ,
    [State] [char] (2) NULL ,
```

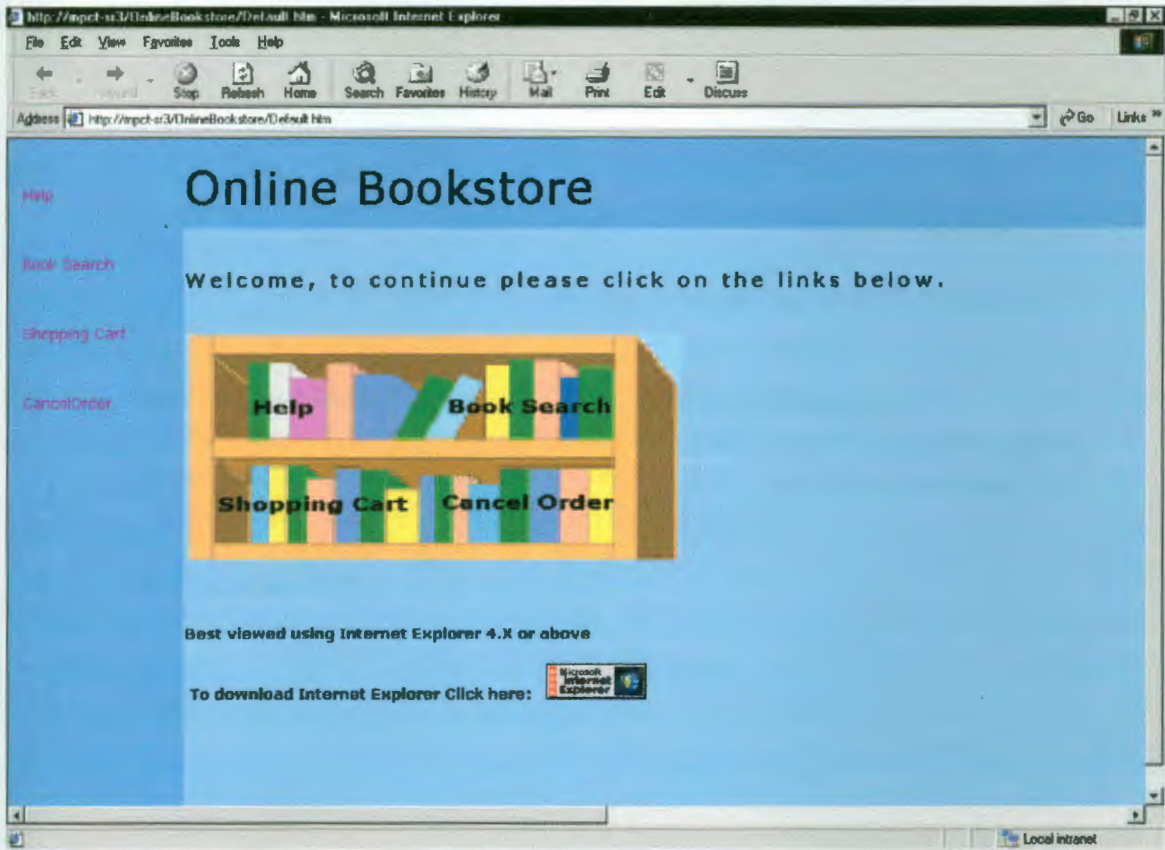
```
[Zip] [varchar] (10) NULL ,
[Phone] [char] (10) NULL ,
[Email] [varchar] (30) NULL ,
[CreditCardNo] [char] (16) NULL ,
[CardType] [char] (8) NULL ,
[ExpDate] [smalldatetime] NULL ,
[Password] [varchar] (10) NOT NULL
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[OrderDetails] (
  [OrderNo] [int] NOT NULL ,
  [ISBN] [char] (10) NOT NULL ,
  [Qty] [tinyint] NULL ,
  [Price] [smallmoney] NULL
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[Orders] (
  [OrderNo] [int] IDENTITY (1000, 1) NOT NULL ,
  [CustLogIn] [varchar] (10) NULL ,
  [OrderDate] [smalldatetime] NULL ,
  [ShipMethod] [varchar] (10) NULL ,
  [ShipCost] [smallmoney] NULL
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[shopcart] (
  [ID] [int] IDENTITY (1, 1) NOT NULL ,
  [ISBN] [char] (10) NOT NULL ,
  [Price] [smallmoney] NOT NULL ,
  [Qty] [tinyint] NOT NULL ,
  [SessionID] [varchar] (30) NOT NULL ,
  [OrderDate] [smalldatetime] NOT NULL ,
  [Title] [varchar] (30) NOT NULL
) ON [PRIMARY]
GO
```

Following are the screens created to build the functionality of the project:



**Figure 3. Screen 1:Default.htm**

This orientation screen welcomes users to the Online Bookstore Web site. This screen is the “Home” page. This screen is used as a launching pad for the project. It has hyperlinks to logically connect to important screens in the project. For page level navigation PageNavbars are used. There is a picture placed on this screen that acts as an image map to link to various pages in the Web site. There is also a link provided to download Internet Explorer 5.0, as this Web site can be best-viewed using Internet explorer browser. The links on this page takes you to following pages:

- Help.htm
- BookSearch.asp
- CartItems.asp
- CancelOrder.htm



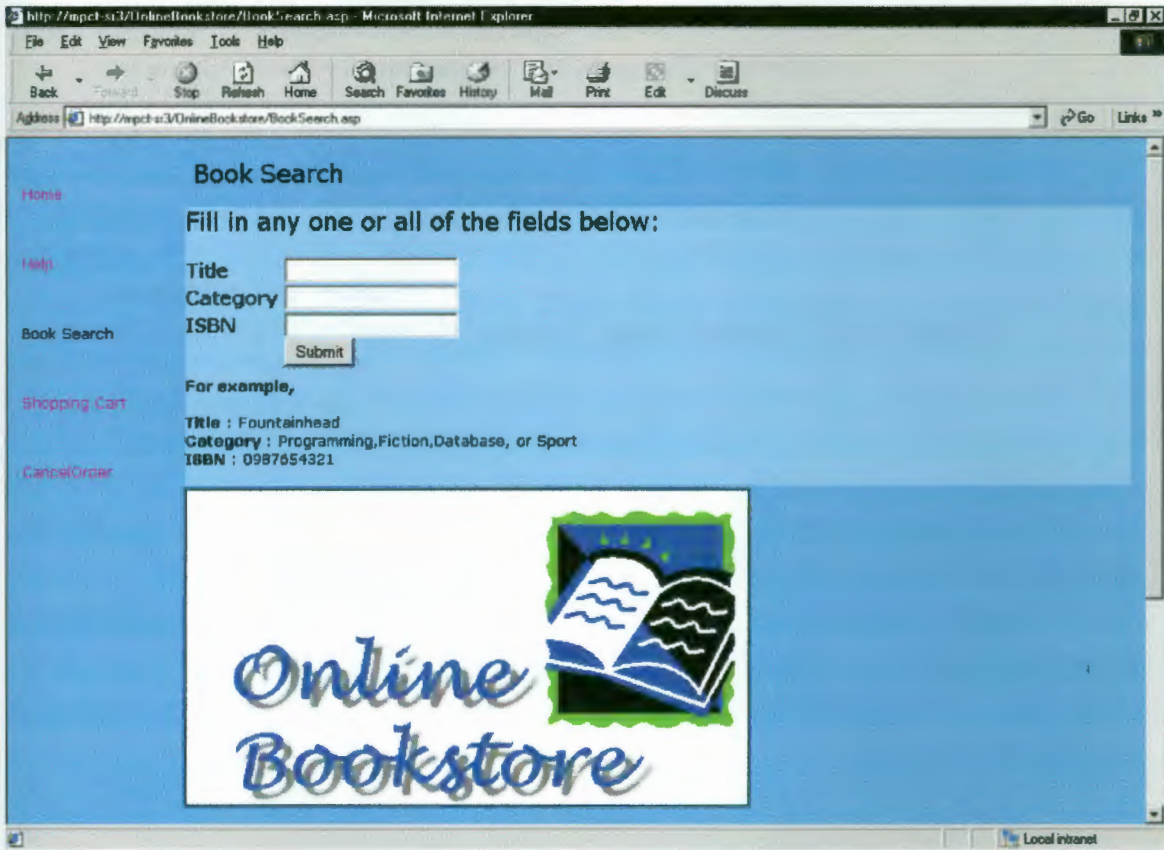


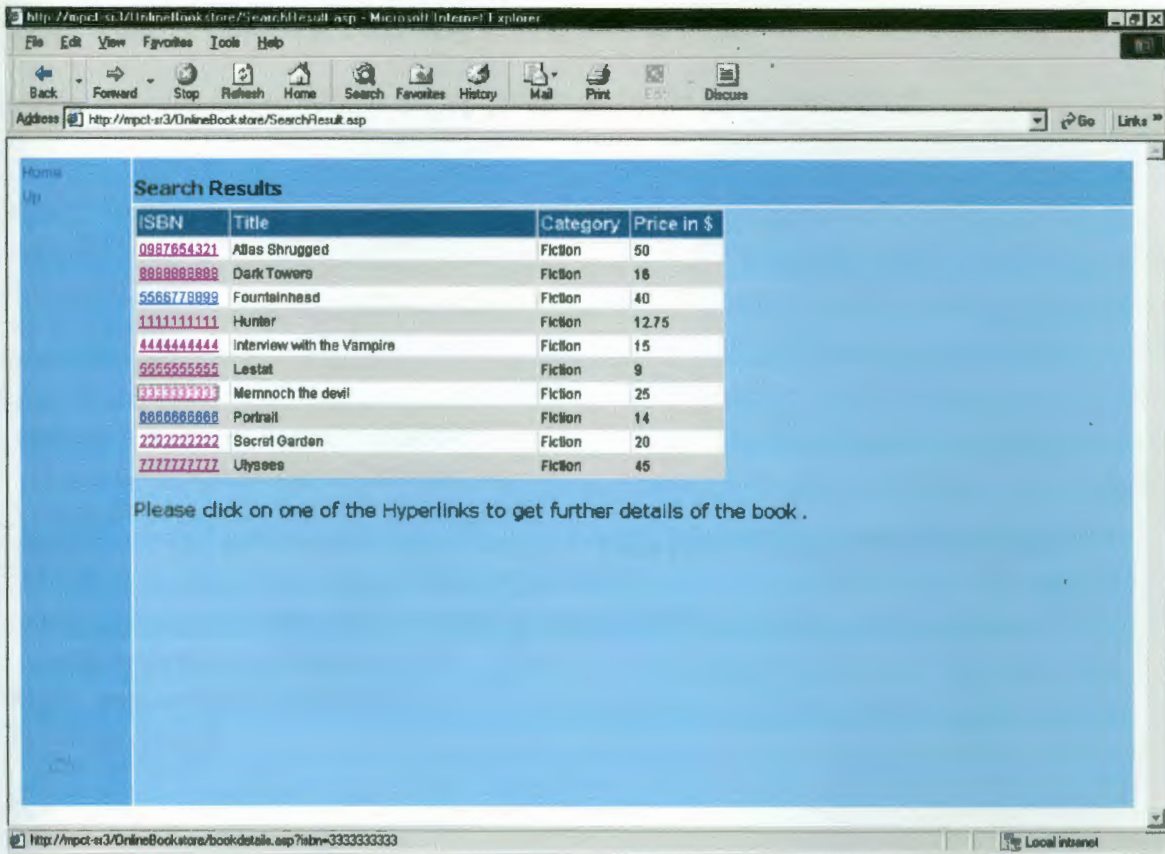
Figure 4. Screen 2: BookSearch.asp

This is the search screen. The user will be allowed to search books according to the Title, Category and ISBN. On this screen there are three text input fields provided for *Title*, *ISBN*, and *Category*. There is a *Submit* button provided on this screen. When this button is clicked, it takes the user to the next page – SearchResult.asp

The following code is used to create the textboxes and submit button:

```
<form method="post" action="SearchResult.asp" id="form1" name="form1">
  <table align="left" border="0" cellPadding="0" cellSpacing="0" width="240">
    <tr>
      <td width="100"><strong><font size="4"
face="Verdana">Title</font></strong></td>
      <td width="140"><input name="txtTitle"></td></tr>
```





**Figure 5. Screen 3: SearchResult.asp**

There are two Design Time Controls (DTC) placed on this screen. Recordset1 is the Recordset object and Grid1 is the grid object placed on the screen. The Recordset object acts as a source to the Grid object. The Recordset1 object uses the conBookStore connection to talk to the database and the following SQL statement is its source of data:

```
SELECT ISBN, Title, Category, Price FROM Books
WHERE (ISBN = '0987654321') OR (Title = 'Atlas Shrugged') OR (Category = 'Fiction')
```

The values shown in the above statement for the fields ISBN, Title, and Category are supplied to the Recordset before opening the Recordset. These values were obtained from the previous screen BookSearch.asp. A user entered these values in the previous screen.

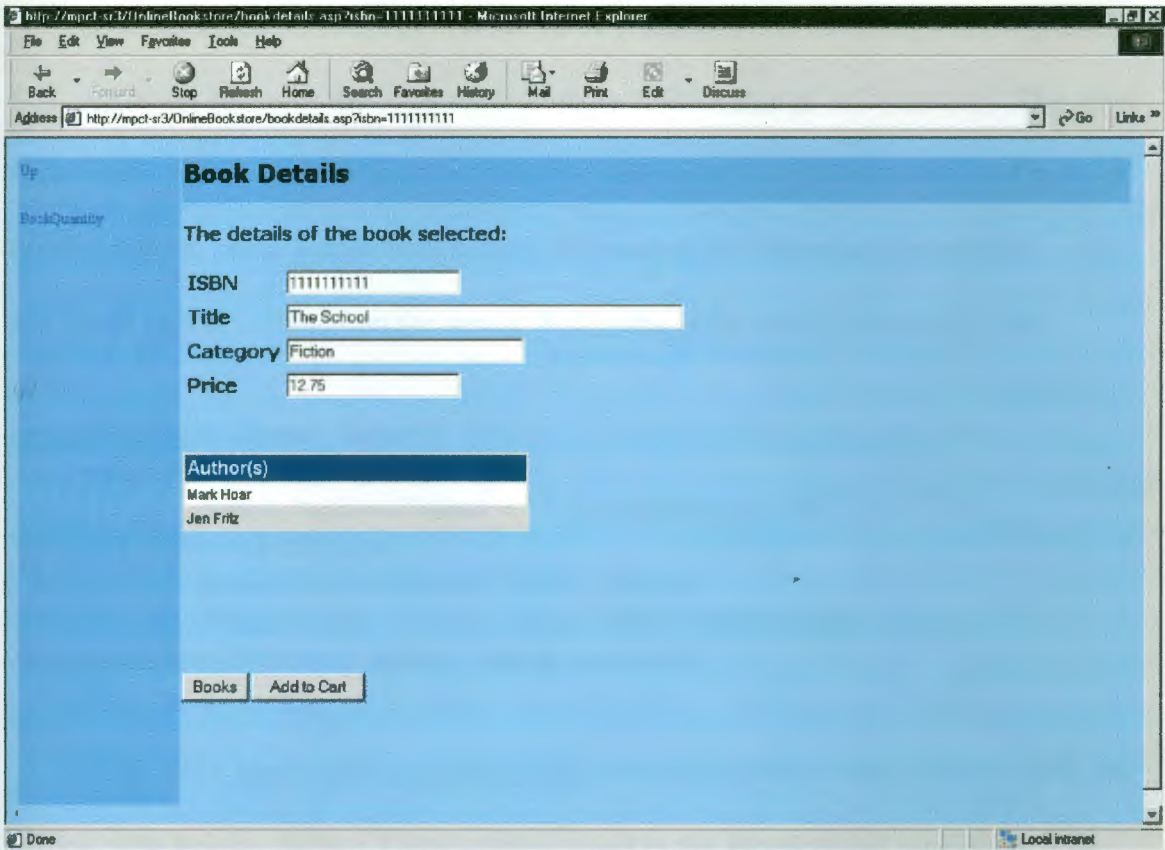
This code retrieves the records from the Books table that match the data entered for

ISBN, Title, or Category. Following code is used to obtain the entered data in the previous screen and to submit the sql statement to the Recordset object:

```
<%  
dim strURL  
dim strSQL  
dim vParam  
dim vParam1  
dim vParam2  
vParam = Request.Form("txtISBN")  
vParam1 = Request.Form("txtTitle")  
vParam2 = Request.Form("txtCategory")  
strSQL = "SELECT ISBN, Title, Category, Price FROM Books WHERE (ISBN = ""  
strSQL = strSQL + vParam + "" ) OR (Title = ""  
strSQL = strSQL + vParam1 + "" ) OR (Category = ""  
strSQL = strSQL + vParam2 + "" )"  
Recordset1.setSQLText(strSQL)  
Recordset1.open  
  
%>
```

This is Active Server Pages (ASP) code, hence it is enclosed in “<%” and “%>“. The output of Recordset1 is the source for the Grid1 object. The Grid1 object has four columns – ISBN, Title, Category, and Price. The ISBN column’s values are all hyperlinked to allow the user to click on one of the hyperlinks at a time to advance to the next screen to see further details about the specific book. The following code is used to hyperlink each ISBN in the grid:

```
= "<a href=bookdetails.asp?isbn=" + [ISBN] + ">" + [ISBN] + "</a>"
```



**Figure 6. Screen 4: BookDetails.asp**

This screen displays the details of a specific book the user has selected in the previous screen. On this screen two Recordset objects, a grid object and five textboxes are placed. Recordset1 acts as a source for all the textboxes and Recordset2 acts as a source for the grid object. The value of the ISBN selected in the previous screen is obtained by using the following code:

```
dim intISBN
intISBN = Request.QueryString("ISBN")
```

The source for Recordset1 is a SQL statement. This SQL statement is as follows:

```
SELECT ISBN, Title, Category, Price FROM Books WHERE (ISBN = '1111111111')
```

The ISBN value '111111111' is the value selected when executing the program. This SQL statement retrieves the book record whose ISBN value is '111111111'. The other fields in this record are displayed by using the textboxes. To get the name(s) of authors for this specific book, Recordset2 is used. The source for Recordset2 is a stored procedure that is executed on the server. Following is the stored procedure code:

```
create procedure Books_Authors
    @ISBN varchar(10)AS
    Select Authors.Fname + " " + isnull(Authors.MI, " ") + " " +
    Authors.LName as FullName
    From Authors, Books_Author_Details
    Where Books_Author_Details.ISBN=@ISBN and
    Authors.AuthorID=Books_Author_Details.AuthorID
```

As there is a Many-to-Many relationship between the Books and Authors table, the joining table Books\_Author\_Details table is used in the above stored procedure. This stored procedure requires a parameter, which is the value of ISBN column that is selected. This parameter is passed to the stored procedure by using the following code:

```
Sub Recordset2_onbeforeopen()
dim vParam
vParam = Request.QueryString("ISBN")
Recordset2.setParameter 1, vParam
End Sub
```

As seen above, the code for passing the parameter to the stored procedure is placed in the Recordset2's onbeforeopen event. This will ensure that this code will get executed before the Recordset will be opened. This recordset2 returns the names of the authors for the selected book.

The value of ISBN is stored in a session variable for future use by using the following code:

```
session("transferISBN")=intISBN
```

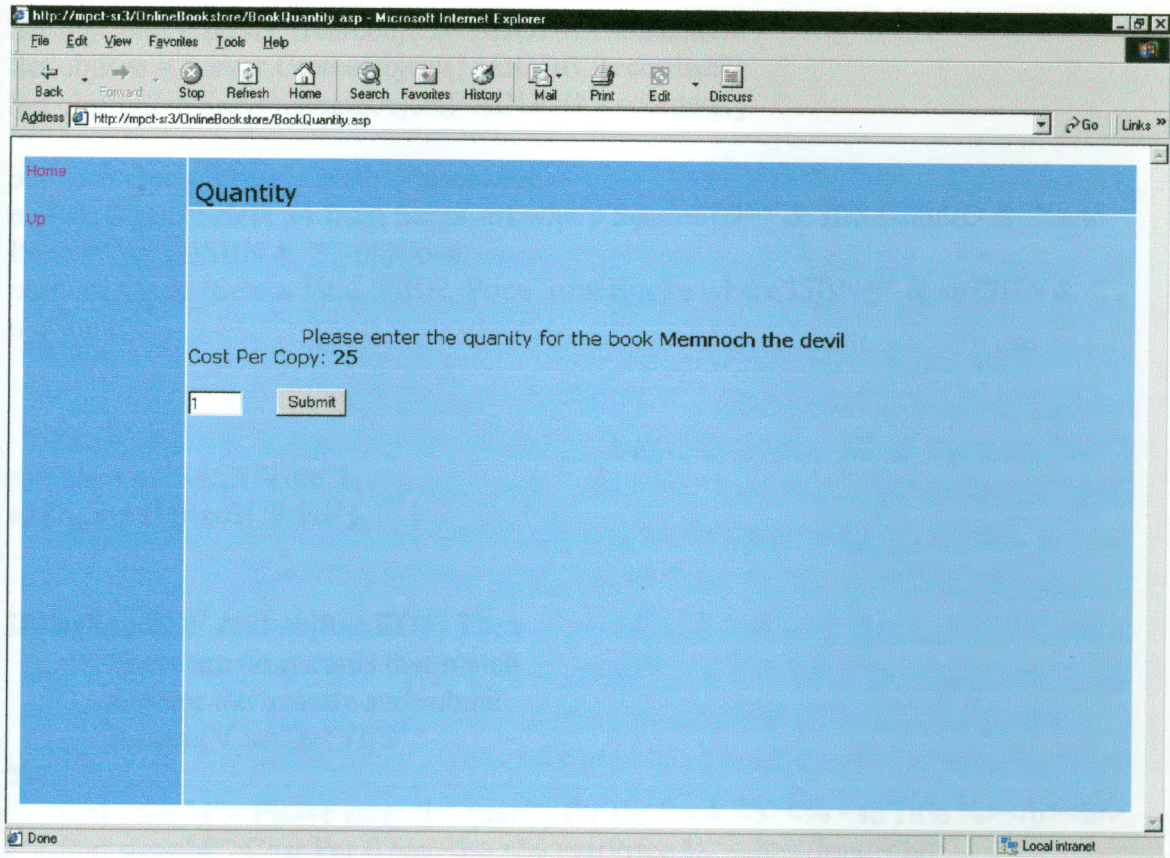
This "transferISBN" session variable is created in the 'Global.asa' page.

There are two command buttons on this screen. When Book Search button is placed it executed the following code to take the user to the BookSearch.asp page.

```
<script LANGUAGE="vbscript">  
<!--  
sub cmdBooks_onclick()  
    window.navigate("BookSearch.asp")  
end sub  
//-->  
</script>
```

When 'Add to Cart' button is pressed the code in its click event gets executed. This code takes the user to the next page, which is BookQuantity.asp page. The code is as follows:

```
<script LANGUAGE="vbscript">  
<!--  
sub cmdAddToCart_onclick()  
    window.navigate ("BookQuantity.asp")  
end sub  
//-->  
</script>
```



**Figure 7. Screen 5: BookQuantity.asp**

On this screen the user can enter the quantity for the selected book. If the book selected is already in the user's shopping cart then he is allowed to change the quantity for the book to be ordered. The submit button on this screen when pressed, takes the user to the next screen CartItems.asp . The following ADO code is used to achieve the same:

```
<%
```

```
dim strISBN, strSessionID, strTitle, strPrice
```

```
strISBN = Session("transferISBN")
```

```
strSessionID = Session("SessionID")
```

```
strSessionID= Session.SessionID
```

```
Dim objConn, objRec, objRec2
```

```
Set objConn = server.createobject("ADODB.Connection")
Set objRec = Server.CreateObject("ADODB.Recordset")
Set objRec2 = Server.createobject("ADODB.Recordset")
```

```
objConn.Open "priya", "shah", "password"
objRec.Open "Select 34 from ShopCart where SessionID='" & strSessionID & "' and
ISBN='" & strISBN & "'", objConn
objRec2.Open "Select Title, ISBN, Price from Books where ISBN='" & strISBN & "'",
objConn
```

```
strTitle = objRec2("Title")
strPrice = objRec2("Price")
```

```
If (objRec.BOF And objRec.EOF) Then
    'There are no records that match
    'Ask for the quantity and submit
    Session("Cart")="YES"
```

```
%>
```

```
<font size="4"> Please enter the quantity for the book <b><% =strTitle %></b><br>
<font size="4">Cost Per Copy:<b> <% =strPrice %></b></font><br>
<form ACTION="CartItems.asp" METHOD="POST" id="form2"
```

```
name="form2">
```

```
<input TYPE="TEXT" NAME="Qty" VALUE="1" WIDTH="30px" size="5">
<input TYPE="HIDDEN" NAME="ISBN" VALUE="<%=strISBN %>">
<input TYPE="HIDDEN" NAME="Title" VALUE="<%=strTitle %>">
<input TYPE="HIDDEN" NAME="Price" VALUE="<%=strPrice %>">
<input TYPE="HIDDEN" NAME="Edit" VALUE="NO">
<input TYPE="SUBMIT" NAME="Submit" VALUE="Submit">
```

```
</form>
```

```
<%
```

```
Else
```

```
'They have this item in their shopping cart already
'tell them they do and ask them to make a change to their quantity
'and submit
Session("Cart")="YES"
```

```
%>
```

```
<font size="4">You already have this item in your cart... <br>
To make a change to the quantity, change it and click change.<br>
Cost Per Copy: <% =strPrice %><br></font>
```

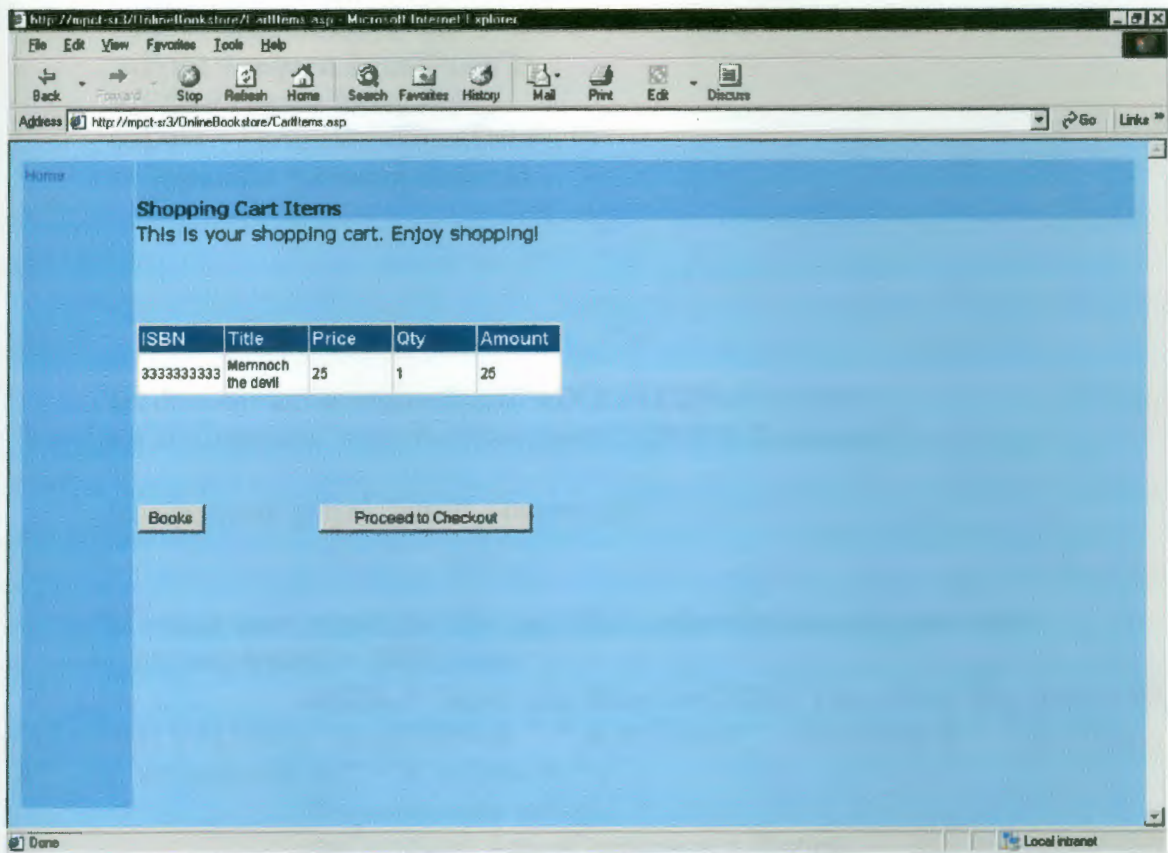
```
<form ACTION="CartItems.asp" METHOD="POST" id="form1 "
name="form1">
<input TYPE="TEXT" NAME="Qty" VALUE="<%=objRec("Qty") %> ">
<input TYPE="HIDDEN" NAME="ISBN" VALUE="<%=strISBN %>">
<input TYPE="HIDDEN" NAME="Title" VALUE="<%=strTitle %>">
<input TYPE="HIDDEN" NAME="Price" VALUE="<%=strPrice %>">
<input TYPE="HIDDEN" NAME="Edit" VALUE="YES">
<input TYPE="SUBMIT" NAME="Submit" VALUE="Change">
</form>
```

```
<%
End if
objRec2.Close
set objRec2 = nothing

objrec.Close
Set objRec = Nothing
objConn.Close
Set objConn = Nothing

%>
```

This ADO code uses the DSN “priya” set up on the SQL server.



**Figure 8. Screen 6:CartItems.asp**

On this screen the contents of the user's shopping cart are displayed. There is a Recordset1 object and a grid1 object on the screen. Before displaying the records in his shopping cart, the book selected by the user in the previous screen is added to the Shopcart table. If the user has already selected the book and has just changed the quantity of the book then the existing record in the Shopcart table for that book is updated and the new quantity replaces the previously entered quantity. This code is executed before the Recordset1 is opened. The following code is used to do the same:

```
Sub Recordset1_onbeforeopen()
Dim strISBN, strTitle, strQty, strPrice, strSessionID, iRecs, strSQL
If UCASE(Session("Cart"))="YES" Then

    strISBN = request.form("ISBN")
    strTitle = request.form("Title")
```

```
strQty = request.form("Qty")
strPrice = request.form("Price")
strEdit = request.form("Edit")
strDate = FormatDateTime(Date(), 0)
strSessionID = Session.sessionid
```

```
set dbconn=server.createobject("ADODB.CONNECTION")
set objCommand = server.createobject("ADODB.Command")
```

```
dbconn.open "priya", "shah", "password"
```

```
'If values were passed into this page then update the shopping cart table
if UCase(strEdit) = "NO" Then
```

```
    strSQL = "Insert into ShopCart(ISBN, Title, Price, Qty, SessionID,
OrderDate) VALUES (" & strISBN & ", " & strTitle & ", " & strPrice & ", " & strQty
& ", " & strSessionID & ", " & strDate & ")"
```

```
    'Response.write strSQL
    objCommand.ActiveConnection = dbconn
    objCommand.CommandText = strSQL
    objCommand.Execute iRecs
    'Response.Write iRecs
```

```
ElseIf UCase(strEdit) = "YES" Then
```

```
    strSQL = "Update ShopCart SET ISBN=" & strISBN & ", Title="
& strTitle & ", Price=" & strPrice & ", Qty=" & strQty & ", OrderDate=" & strDate & "
WHERE SessionID=" & strSessionID & """
```

```
    'Response.write strSQL
    objCommand.ActiveConnection = dbconn
    objCommand.CommandText = strSQL
    objCommand.Execute iRecs
    'Response.Write iRecs
```

```
End if
```

```
Session("Cart")=""
```

```
set objCommand = Nothing
dbconn.Close
set dbocnn = nothing
End If
```

```
Recordset1.setParameter 1, session.sessionid
```

```
End Sub
```

The following stored procedure is the source for the Recordset1 object:

```
CREATE PROCEDURE shopcartdata  
@p1 varchar(30) as  
SELECT ISBN, Title, Price, Qty, sessionID FROM shopcart  
WHERE SessionID = @p1
```

This stored procedure returns the records from the Shopcart table that were stored in the same SessionID, which is passed as a parameter to the Recordset object. The Grid1 object displays the records the stored procedure returns.

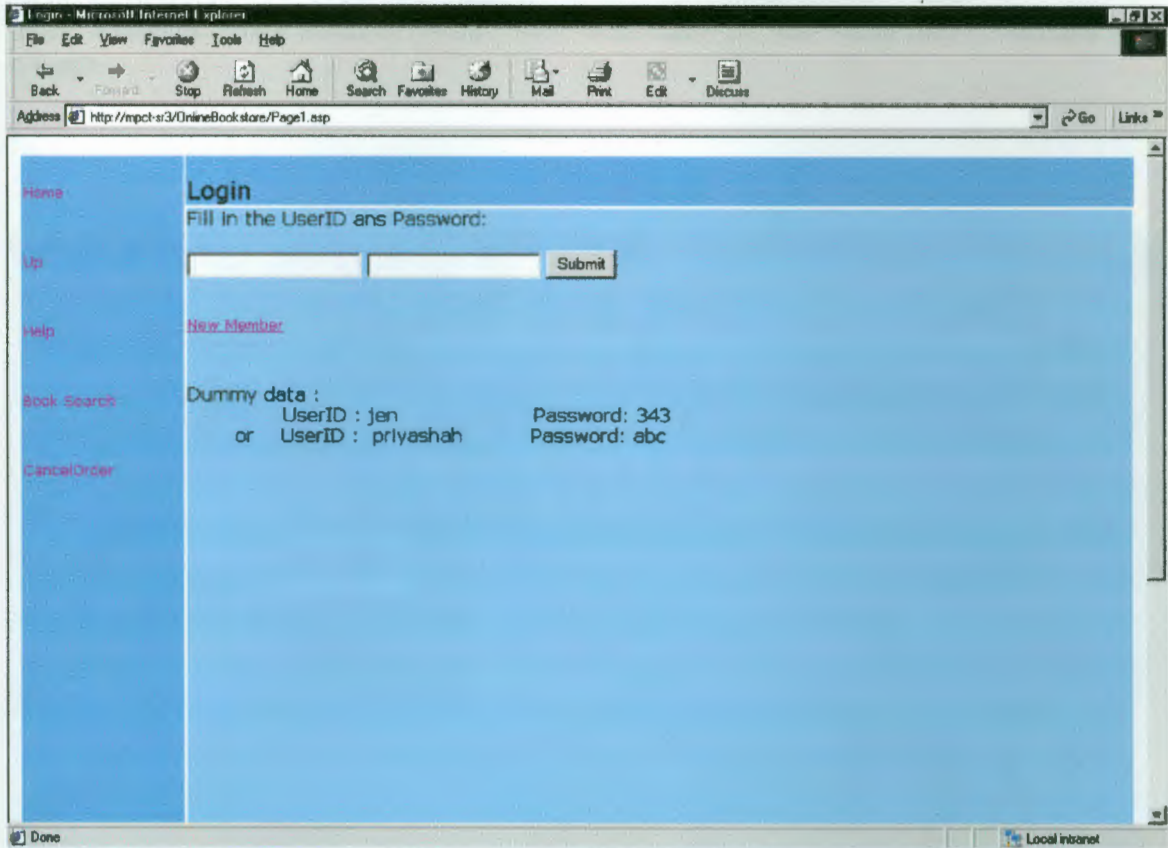
There are two buttons on the screen. When the user presses the *Books* button, the following code is executed and the user is taken to the BookSearch.asp page.

```
<script LANGUAGE="vbscript">  
<!--  
sub cmdBooks_onclick()  
  window.navigate("BookSearch.asp")  
end sub  
//-->  
</script>
```

When the user presses 'Proceed to Checkout' button, the following code gets executed which takes the user to the next screen.

```
<script LANGUAGE="vbscript">  
<!--  
sub cmdProceedtoCheckout_onclick()  
  window.navigate ("Page1.asp")  
end sub  
//-->
```

</script>



**Figure 9. Screen 7: Page1.asp**

This is the login screen. On this screen there are two textboxes and a submit button.

When the Submit button is clicked the user is taken to the next page Pre-Submit. asp .

This input data entry form is used for returning users who already have created an

account on this site. The following code is used to accomplish this:

```
<form method="post" action="Pre-Submit.asp" id="form1" name="form1">  
<input name="txtLogin">  
<input name="txtPassword" Type="password">  
<input type="submit" value="Submit" id="submit1" name="submit1">  
<br><br><br>  
</form>
```

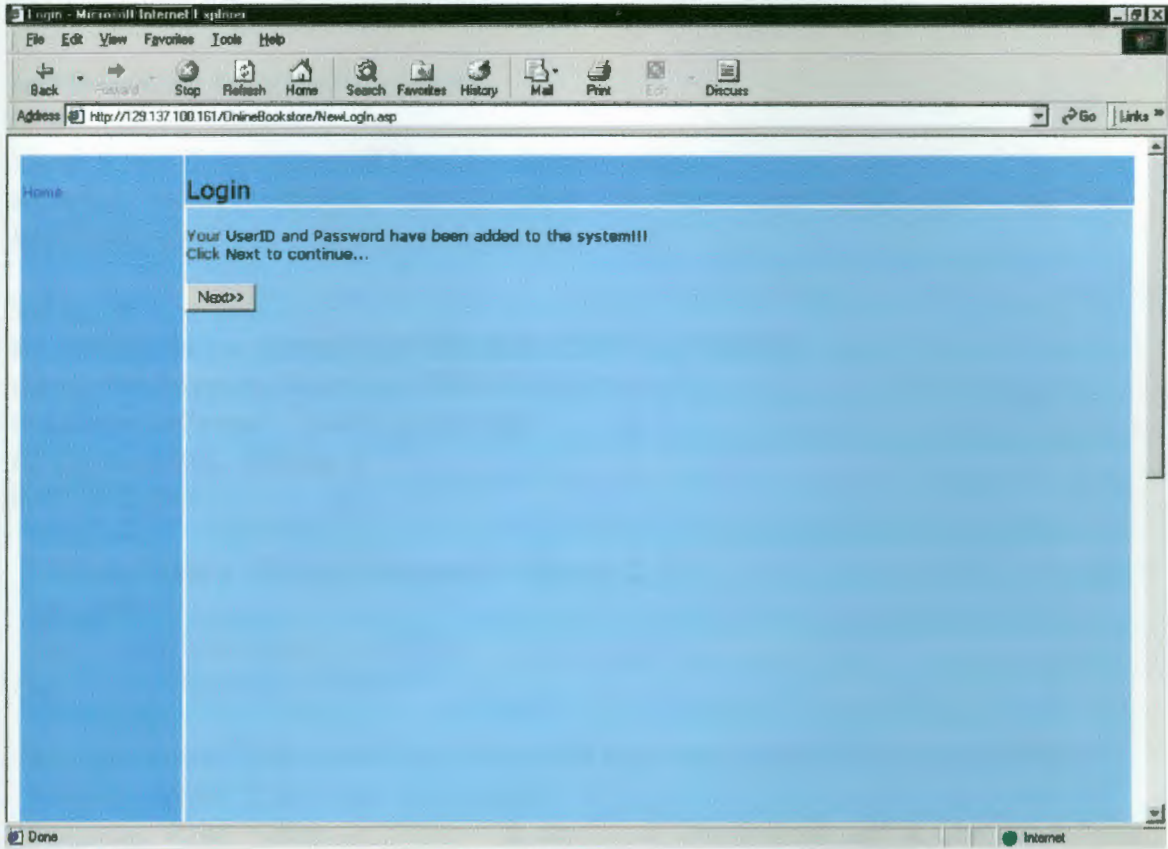
There is also a hyperlink created by clicking this hyperlink, the user can create a new user login ID after clicking this hyperlink.



```

<p><font size="4">Password</font> <input type="password" size="15"
name="txtPassword"></p>
<p> <input type="submit" name="cmdSubmit" value="Submit"></p></form>

```



**Figure 11. Login Confirmation: NewLogin.asp**

On this page the Customers table is checked to see if the UserID entered by the user on the previous screen exists in the table. If the UserID already exists in the table the user is asked to go to the previous screen to enter a new UserID. If the entered UserID does not exist then a new record is added to the Customers table with the data entered for UserID and Password fields. The following ADO code is written to do the same:

```

<%
dim dbconn
dim rst
dim strUID

```

```
dim strPW
dim strVerify
dim strSQL
```

```
'get the values entered on the form
```

```
strUID=request.form("txtLogIn")
```

```
strPW=request.form("txtPassword")
```

```
'check to see if the requested UserID is already assigned
```

```
strSQL= "Select CustLogIn, Password from Customers where (CustLogIn = " & "" & strUID & """)"
```

```
'set up rst
```

```
set dbconn=server.createobject("ADODB.CONNECTION")
```

```
set rst=server.createobject("ADODB.RECORDSET")
```

```
dbconn.open "priya", "shah", "password"
```

```
rst.Open strSQL, dbconn, 2, 3
```

```
if rst.EOF then
```

```
    rst.Close
```

```
    rst.Open "select 34 from Customers", dbconn, 2, 3
```

```
    rst.AddNew
```

```
    rst.Fields("CustLogIn") = strUID
```

```
    rst.Fields("Password") = strPW
```

```
    rst.Update
```

```
    response.write "Your UserID and Password have been added to the system!!!<br>"
```

```
    Response.Write "Click Next to continue... "
```

```
    Response.Write "<form ACTION=" & chr(13) & "Pre-Submit.asp" & Chr(13) & "METHOD=" & Chr(13) & "POST" & Chr(13) & "id=form1 name=form1>"
```

```
    Response.Write "<input TYPE=" & Chr(13) & "HIDDEN" & Chr(13) & " NAME=" & Chr(13) & "txtLogin" & Chr(13) & " VALUE=" & Chr(13) & strUID & Chr(13) & ">"
```

```
    Response.Write "<input TYPE=" & chr(13) & "HIDDEN" & Chr(13) & " NAME=" & Chr(13) & "txtPassword" & Chr(13) & " VALUE=" & Chr(13) & strPW & Chr(13) & ">"
```

```
    Response.Write "<input TYPE=" & chr(13) & "SUBMIT" & Chr(13) & " VALUE=" & Chr(13) & "Next>>" & Chr(13) & " id=SUBMIT1 name=SUBMIT1>"
```

```
    Response.Write "</form> "
```

```
else
```

```
    response.write "The UserID you entered is assigned to another user! <br>"
```

```
    response.write "Please go back and enter a different UserID. <br>"
```

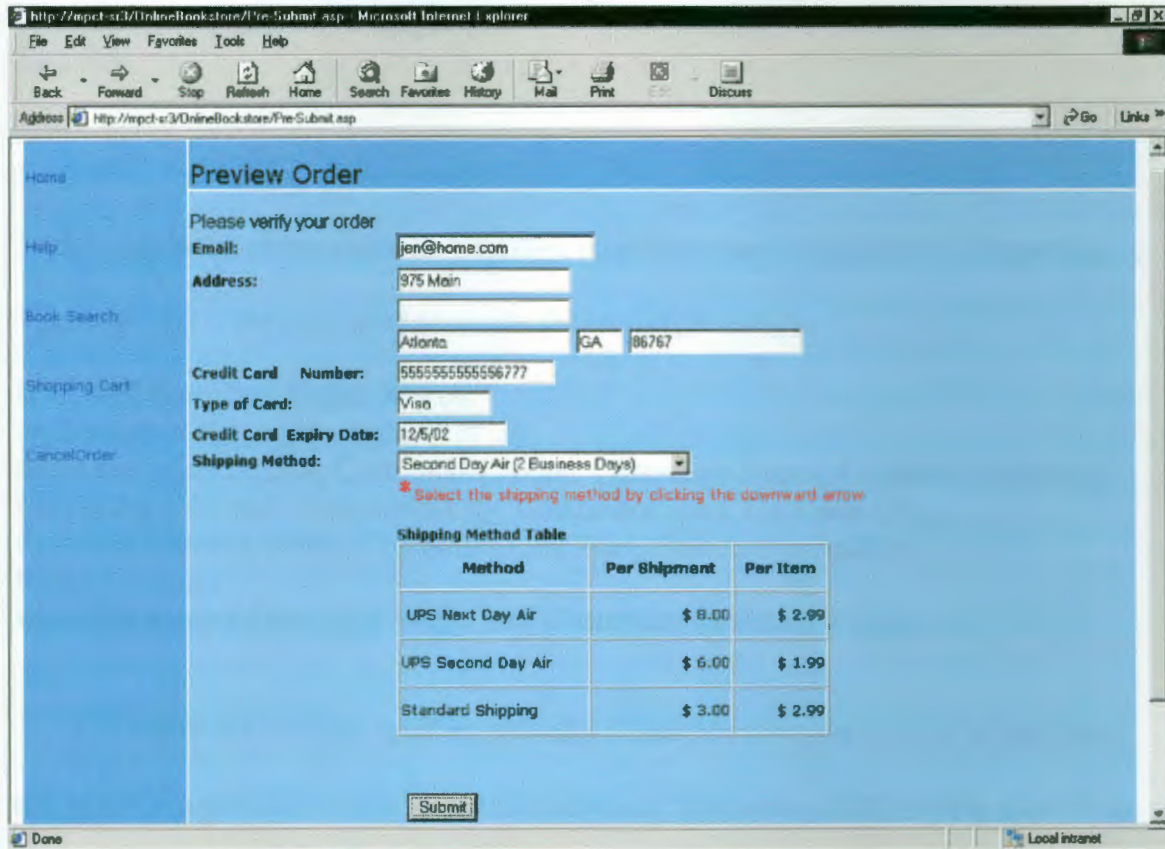
```
    Response.Write "<form ACTION=" & chr(13) & "newmember.htm" & Chr(13) & "METHOD=" & Chr(13) & "Get" & Chr(13) & ">"
```

```
    Response.Write "<input TYPE=" & chr(13) & "SUBMIT" & Chr(13) & " VALUE=" & Chr(13) & "&lt;&lt;Back" & Chr(13) & ">"
```

```

Response.Write "</FORM>"
end if
rst.Close
%>

```



**Figure 12. Screen 8: Pre-Submit.asp**

On this screen there is a Recordset object, nine textboxes and a drop-down list box.

The source for the Recordset object is a stored procedure. This stored procedure requires two parameters. The parameters are the Login information like UserID and Password provided by the user in the previous screen. By using the following code the parameters are passed to the stored procedure:

```
<script ID="serverEventHandlersVBS" LANGUAGE="vbscript" RUNAT="Server">
```

```

Sub rcdLogin_onbeforeopen()
dim vParam

```

```

dim vParam1
vParam = Request.Form("txtLogin")
vParam1 = Request.Form("txtPassword")
rcdLogin.setParameter 1, vParam
rcdLogin.setParameter 2, vParam1
session("IDLogin") = vParam
'response.write session("IDLogin")
End Sub

```

```
</script>
```

As seen in the above code the UserID is stored in a session variable for future use.

The following is the code used to create the stored procedure:

```

CREATE procedure Login_Search
@p1 varchar(10) , @p2 varchar(10) as
select Customers.Email, Customers.LName, Customers.Fname, Customers.Address1,
Customers.Address2, Customers.City, Customers.State, Customers.Zip,
Customers.CreditCardNo, Customers.CardType, Customers.ExpDate
from Customers
where Customers.CustLogIn = @p1 and Customers.Password = @p2

```

For future use the total number of books ordered in the current order is calculated and saved in a session variable called 'TotalBooks'. Following ADO code is used to do the same. This code sums the quantities for books in the shopcart for the current session.

```

<%
dim strSQL, iRecs
dim rs
set dbconn=server.createobject("ADODB.CONNECTION")
set objCommand = server.createobject("ADODB.Command")

dbconn.open "priya", "shah", "password"
dbconn.DefaultDatabase = "shah"

strSQL = "SELECT sum(Qty) FROM ShopCart WHERE SessionID = ""
strSQL = strSQL + session.SessionID + """"
objCommand.ActiveConnection = dbconn

objCommand.CommandText = strSQL
set rs = objCommand.Execute(iREcs)

```

```
session("TotalBooks") = rs.fields(0).value
rs.close
set rs = nothing
'response.write session("NumBooks")
%>
```

There is a Drop Down list box object on this screen to get user input about the shipping method for the particular order. To calculate the shipping cost for the order the shipping method selected by the user has to be found out for each order. The following code is used to trap the selected shipping method and passed the value to the next screen.

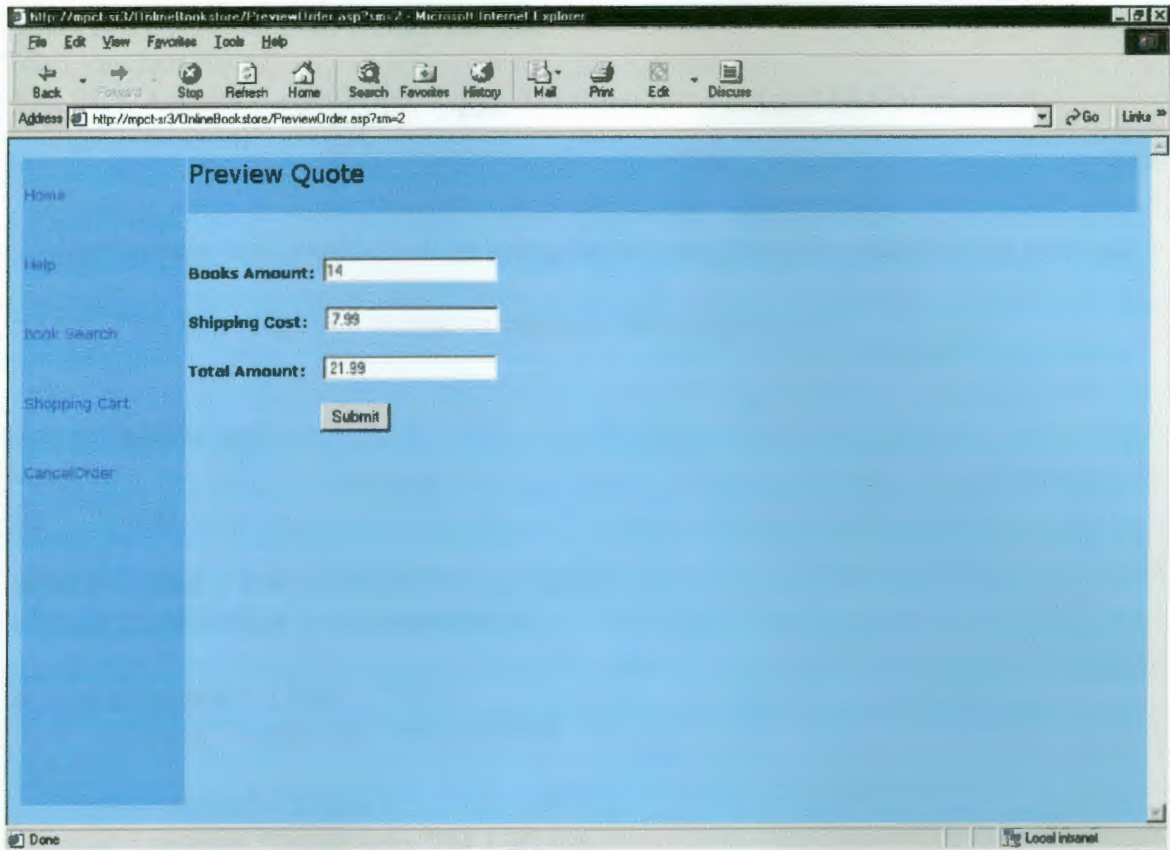
```
<script LANGUAGE="vbscript">
<!--
sub cmdSubmit_onclick()

    dim GoingTo

    GoingTo = "PreviewOrder.asp?sm=" & document.thisForm.select1.value

    window.navigate(GoingTo)
end sub
//-->
</script>
```

In the above code the 'select1' is the name of the drop down list box on this screen.



**Figure 13. Screen 9: PreviewOrder.asp**

On this screen the books amount, shipping cost, and total amount for the order are displayed to the user. There is Submit button on this screen. When it is pressed the user is taken to the next screen. To calculate the books amount a stored procedure is used. The stored procedure requires the SessionID as a parameter. This SessionID parameter is passed to the stored procedure using the following code:

```
<script ID="serverEventHandlersVBS" LANGUAGE="vbscript" RUNAT="Server">
Sub Recordset1_onbeforeopen()
Recordset1.setParameter 1, session.sessionID
End Sub
</script>
```

The following is the code used to create BooksAmount stored procedure:

```
CREATE PROCEDURE BooksAmount
```

```
@p1 varchar(30) as
```

```
SELECT Amount = sum(Price34 qty), TotalBooks = sum(qty) FROM shopcart
```

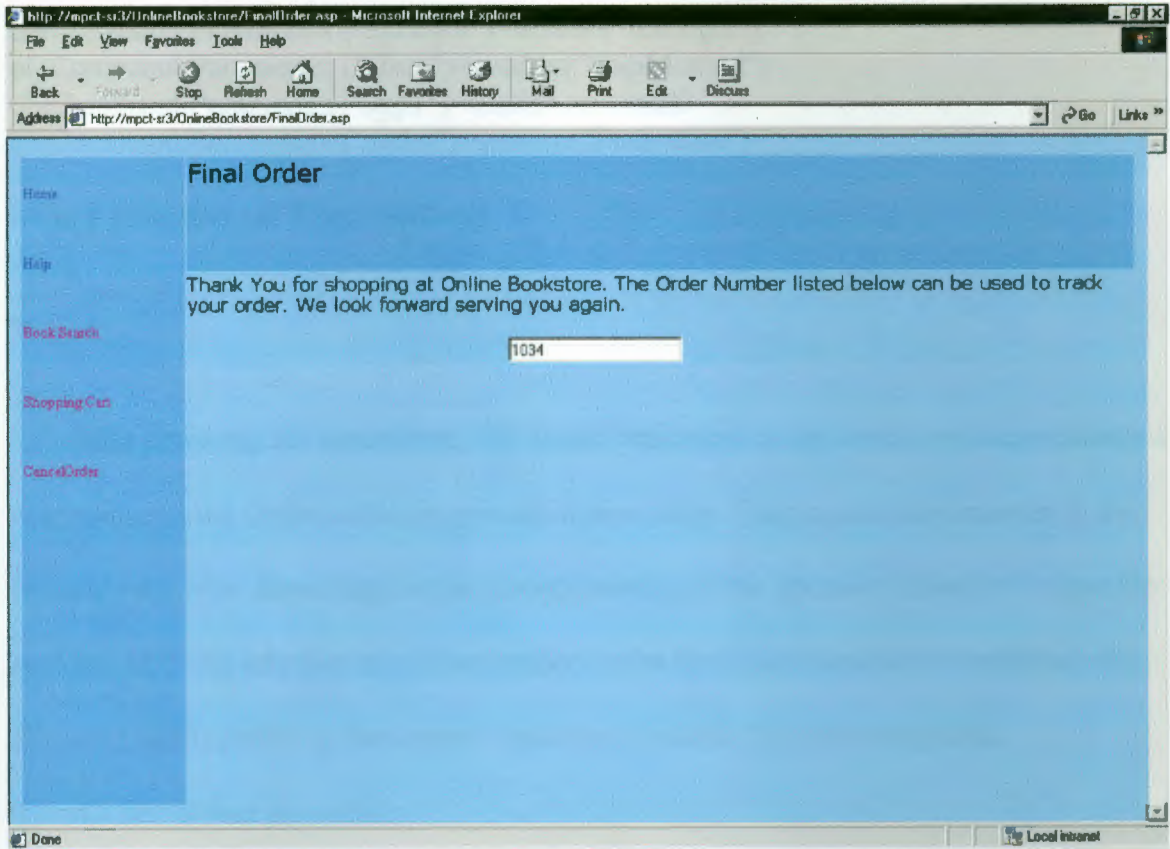
```
WHERE SessionID = @p1
```

The shipping cost is calculated by using the shipping method selected in the previous screen. The following code is used to calculate the shipping cost:

```
<%  
dim strShipMethod  
dim cost  
dim TotCost  
  
strShipMethod = Request.QueryString("sm")  
session("ShipMethod") = strShipMethod  
  
if strShipMethod = 1 then  
    cost = session("TotalBooks") 34 2.99 + 8  
else  
    if strShipMethod = 2 then  
        cost = session("TotalBooks")34 1.99 + 6  
  
    else  
        cost = session("TotalBooks")34 0.99+ 3  
  
    end if  
end if  
  
%>
```

The total cost of the order is calculated by using the following code:

```
<%  
TotCost = cost + Textbox1.value  
%>
```



**Figure 14. Screen 10: FinalOrder.asp**

On this screen the user is displayed the order number for his reference. The order number is obtained by executing a stored procedure. This stored procedure requires four parameters to be passed. The following ADO code is used to call the stored procedure and pass the parameters:

```
<%  
  
Dim objConn, iRecs, objCommand, rs  
  
Set objConn = server.createobject("ADODB.Connection")  
set objCommand = server.createobject("ADODB.Command")  
  
objConn.Open "priya", "shah", "password"  
  
objCommand.ActiveConnection = objConn.ConnectionString  
objCommand.CommandType = 4  
objCommand.CommandText = "NewOrder"
```

```

objCommand.Parameters ("@Login")=session("IDLogin")
objCommand.Parameters ("@sm")=session("ShipMethod")
objCommand.Parameters ("@p1")=session.sessionID
objCommand.Parameters ("@dat")=session("Todaydate")

set rs = objCommand.Execute(iRecs)

```

```
%>
```

After receiving the parameters, the 'insert' statement in the stored procedure inserts a new record in the Orders table to generate a new order. Then records are inserted in the OrderDetails table depending on the records existing in the Shopcart table for the specific session. After the relevant records are added to the OrderDetails table the records in the Shopcart table specific to the current session are deleted. The following is the 'NewOrder' stored procedure:

```

CREATE procedure NewOrder
@Login as varchar(10), @sm as int,
@p1 as varchar(30),@dat as datetime

as

begin transaction

insert into Orders
(CustLogIn, OrderDate, ShipMethod)
values (@Login,@dat,@sm)

Insert into OrderDetails (OrderNO, ISBN, Qty, Price)
select Orders.OrderNo, Shopcart.ISBN, Shopcart.Qty, Shopcart.Price
from Orders, Shopcart
where Shopcart.sessionID = @p1

delete from shopcart where sessionID = @p1

Commit transaction

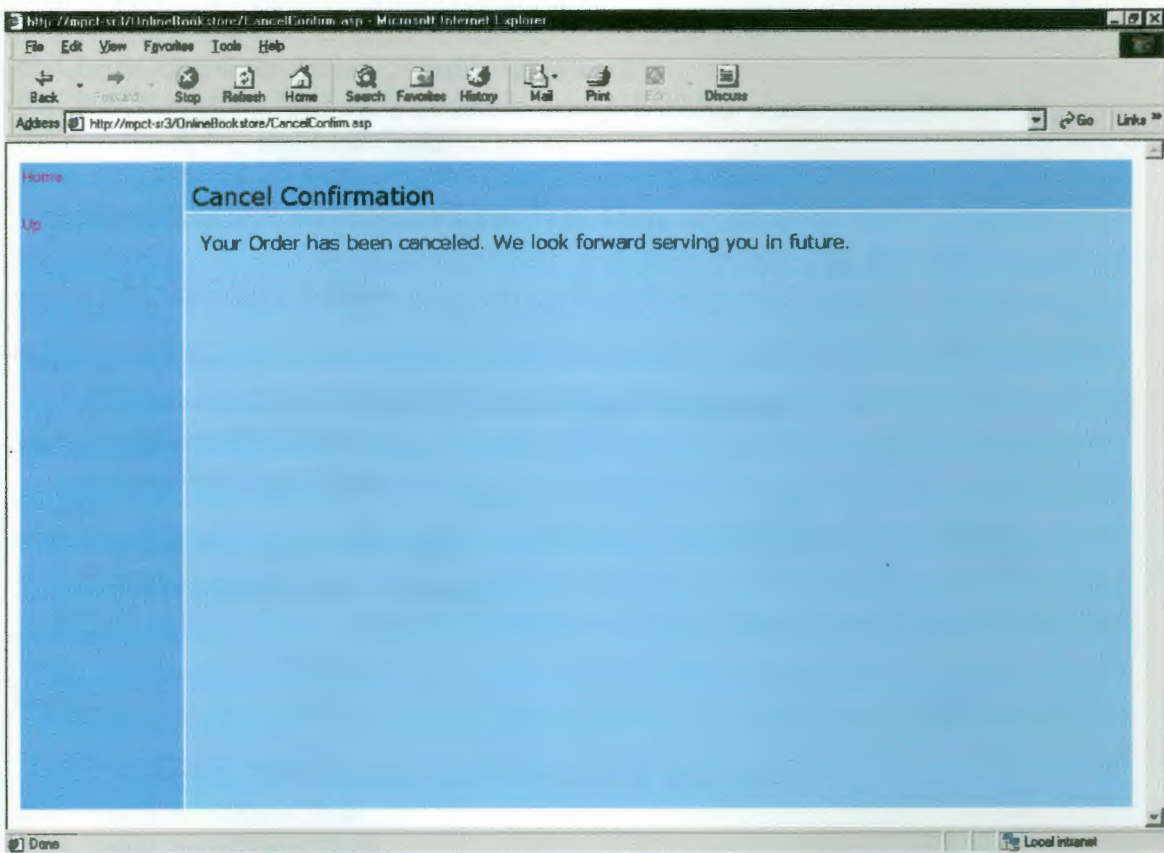
```

```
select OrderNo from Orders
where Orders.CustLogIn = @Login and OrderDate = @dat
```

As seen in the stored procedure new order number is returned to be displayed on the screen. Following code is used to display the generated order number:

```
<input id="text1" name="text1" value="<%=rs.Fields(0).Value%>">
```





**Figure 16. Order Cancel Confirmation: CancelConfirm.asp**

If the user has entered an existing order number in the previous screen, an order cancellation confirmation is displayed to him. If he has entered a non-existing order number then he is asked to press the '<<Back' button to go to the previous screen to enter a correct order number. The following ADO code is used to accomplish this:

```
<%  
Dim strOrderNo  
strOrderNo = Request.Form("txtCancelOrderNo")  
  
Dim objConn, iRecs, objCommand, rs  
Dim objCheckConn, objCheckRec  
Dim bOrderDeleted  
  
bOrderDeleted = False  
  
Set objCheckConn = Server.CreateObject("ADODB.Connection")  
Set objCheckRec = Server.CreateObject("ADODB.Recordset")
```

```
objCheckConn.Open "priya", "shah", "password"  
objCheckRec.Open "SELECT OrderNo from ORDERS where ORDERNo=" &  
strOrderNo & """, objCheckConn
```

```
If objCheckRec.BOF and objCheckRec.EOF Then
```

```
    bOrderDelete = False
```

```
    'Close the Check OrderID Recordset and Connection  
    objCheckRec.Close  
    objCheckConn.Close
```

```
    Set objCheckRec = Nothing  
    Set objCheckConn = Nothing
```

```
Else
```

```
    'Close the CheckConnection and recordset and go on...  
    objCheckRec.Close  
    objCheckConn.Close
```

```
    Set objCheckRec = Nothing  
    Set objCheckConn = Nothing
```

```
    Set objConn = server.createobject("ADODB.Connection")  
    set objCommand = server.createobject("ADODB.Command")  
    'set rs = server.createobject("ADODB.recordset")  
    objConn.Open "priya", "shah", "password"
```

```
    objCommand.ActiveConnection = objConn.ConnectionString  
    objCommand.CommandType = 4  
    objCommand.CommandText = "Cancel_Order"  
    objCommand.Parameters ("@P1")= strOrderNo  
    set rs = objCommand.Execute(iRecs)
```

```
    bOrderDeleted = True
```

```
End If
```

The following code is used to display the appropriate messages to the user:

```
If bOrderDeleted = true then
```

Response.Write"<p><br><font size=4>&nbsp;   Your Order has been canceled. We look forward serving you in future. </font></p> <p>&nbsp;   "

Else

Response.Write "Were sorry... The Order Number that you have entered was not found in our system.<BR>"

Response.Write "Please go back and enter a valid Order Number."

Response.Write "<form ACTION=" & chr(13) & "CancelOrder.htm" & Chr(13) & " METHOD=" & Chr(13) & "Get" & Chr(13) & " id=form1 name=form1>"

Response.Write "<input TYPE=" & chr(13) & "SUBMIT" & Chr(13) & " VALUE=" & Chr(13) & "&lt;&lt;Back" & Chr(13) & " & 1>"

Response.Write "</FORM>"

End If

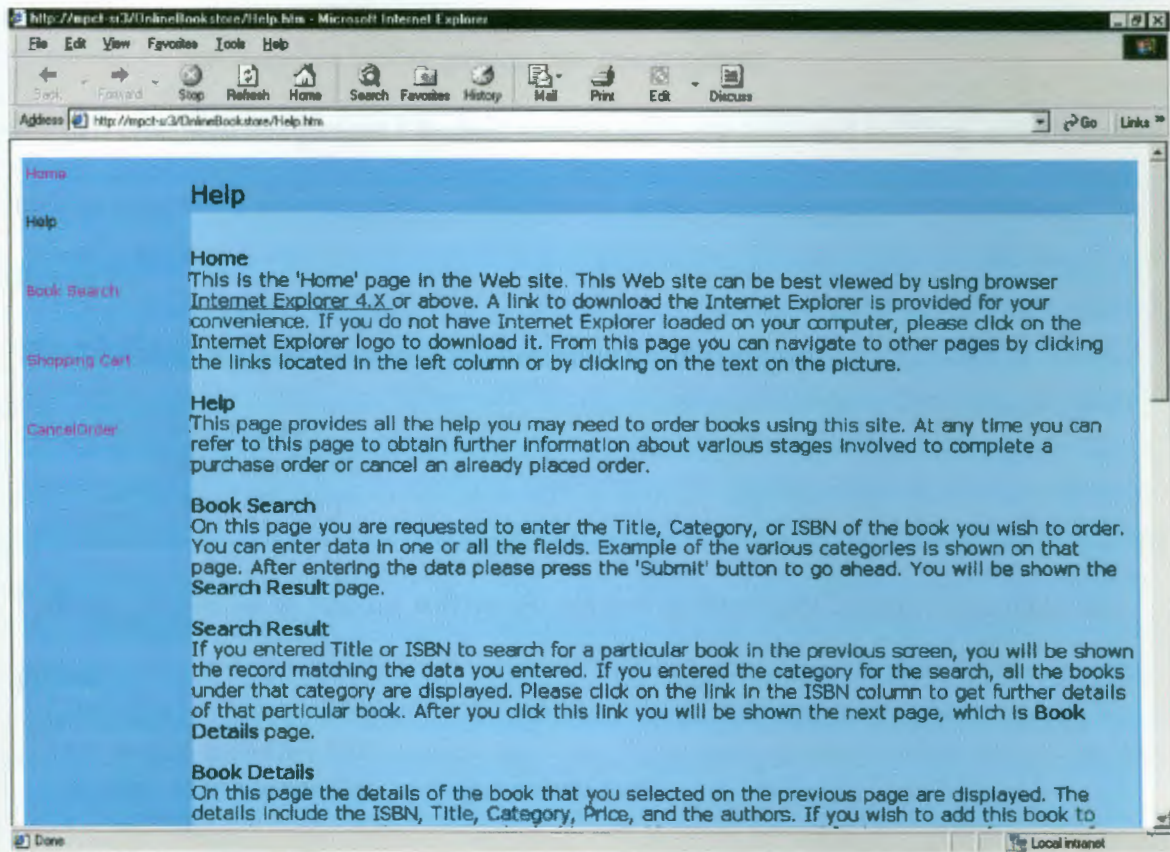


Figure 17. Help: Help.htm

This screen will provide information to the user about how to search for books, place purchase orders, and cancel placed orders.

## Conclusions and Recommendations

I selected the Online Bookstore project because it covered the subjects of Internet Technology and Database Management skills. I wanted to get into the wonderful world of web development by doing this project. I also wanted to make use of database management skills that I have acquired over the last few years. This online bookstore project was an ideal project to make use of the various skills that I acquired while pursuing this bachelor's degree in Information Engineering Technology and the industry experience I had acquired before getting into this program.

To complete this project I learned various software tools and working with three-tier architecture was a new experience. The system architecture diagram can be seen in Figure 1.

The following were the hardware requirements for the project:

- Web Server
- Database Server
- Client Machine

Following is the list of various software tools and technologies I used to complete the project:

- Visual InterDev 6.0
- MS SQL Server 7.0
- HTML
- Active Server Pages (ASP)
- ActiveX Data Objects (ADO)
- VB Script
- Internet Information Server 4.0

I selected MS SQL Server 7.0 as a back-end because it is a widely used RDBMS in the industry today. Existing online bookstores such as AMAZON.COM and

BarnesandNoble.com are successfully using it to store their data. It offers expanded memory, transaction capabilities, graphic tools for creating objects, managing users, and tracking performance. My overall experience working with MS SQL Server 7.0 has been very positive.

I selected MS Visual InterDev 6.0 as a Web site development tool because it is a widely used Rapid Application Development (RAD) tool. Creating HTML and ASP pages using Visual InterDev is very easy. Building effective Web documents and input forms using stylesheets, images and other media to establish a complete look and feel for the Web application is very easy. An efficient link to the back-end to access data is easier to setup in this environment. A stable link to the MS SQL Server database can be achieved. It offers the flexibility of using VB Script and/or Java Script as scripting languages for the pages.

I used ADO code to dynamically access the data as it offers higher control over data access. I used VB Script to program my ASP pages.

The following is a table below that shows the various skills I acquired in different areas by taking different courses while pursuing my Bachelor's degree in Information Engineering Technology.

Area	Software tools & technology	Courses
Database	MS SQL Server	Database Management II
Web	HTML MS Visual InterDev 6.0 VB Script	The Internet Special Topics in IET Object Oriented Programming I, II, III
User Interface	Web page and Graphic Design	Multimedia Production Graphics for Presentations and Publication
Security and Web Server	Internet Information Server 4.0	Computer Networking III

In short, I utilized various skills I acquired by taking a number of courses to develop my senior design project.

I faced various challenges during the development phase. I took the Special Topics in IET (Visual InterDev 6), Database Management II (MS SQL Server 7), and Senior Design II courses during the same quarter. To display a working prototype of the project at the end of Senior Design II was the first major challenge that I faced. I was comfortable working with SQL Server in a very short time because I had already taken the Database Management I course, and my professional experience in the industry had made me aware of the RDBMS issues. Visual InterDev 6 offered a very steep learning curve. The technology was very new to me. I faced many problems

connecting to the database. Once the database connectivity was done it provided me an efficient connection to my back-end.

I have written a number of stored procedures for faster data access. Some of these stored procedures require parameters to be passed for execution. On some of my ASP pages I used the DTC controls to aid data access. Passing the parameters from the pages to execute the stored procedures was the next challenge. After more in-depth research I found the solution to passing the parameters to the stored procedures. The solution was to pass the parameters through the Onbeforeopen event of the Recordset object.

On the SearchResult.asp page I placed a Grid object that displays the output of a SQL query. I needed the ISBN column of each record to be hyperlinked. Coding the hyperlink functionality for ISBN column for each record was a challenge. After more research and experiments I found the following solution:

```
"<a href=bookdetails.asp?isbn=" + [ISBN] + ">" + [ISBN] + "</a>"
```

The next challenge was using ADO code for the login screens. For all the screens before the login screen I was using Visual InterDev's DTC controls to access the data. These DTC controls write the underlying ADO code for the developer. These DTC controls are not flexible because the developer is not allowed to change or alter the underlying ADO code. Complex requirements cannot be fulfilled using the DTC controls, writing all the ADO code yourself gives you more control over the results. I had to learn ADO scripting for the same. I recommend using ADO code to code the data access screens instead of using Recordset objects as ADO code offers more control over the task.

The next major challenge was coding the shopping cart. I had to add a shopping cart table to my database at this stage of the project. I wrote ADO code to add records to the shopping cart.

The last challenge was placing the order items into the Orders and OrderDetails tables from the Shopcart table. I defined session variables to hold some of my data. I wrote a stored procedure to append records to the Orders and OrderDetails tables.

If I have to commercially make use of this Online Bookstore project, I will need to do some modifications to this project. I will have to enhance the user interface. I recommend using Dreamweaver to create the Web pages for better look and feel of the Web site. I will have to make use of frames to give it a more professional look and feel. Currently, I am using dummy data for the project. To commercially use this project I will have to get data of real books from various publishers. Currently I am assuming the credit card information that the user enters is correct. For commercially using this bookstore I will need to interface a credit card validation mechanism to verify the credit card information entered is correct. I will also have to enhance the security offered by the site, as customers would be entering their sensitive data such as credit card information.

Overall, the design and development process of this Online Bookstore project was a very challenging and satisfying experience. I acquired many new technical skills while working on this project. I also utilized my existing skills for the completion of the project. The experience that I acquired while working on this Senior Design project has made me a better Information Technology engineer.

## Notes

1. **Hyperlink:** An element in an electronic document that links to another place in the same document or to an entirely different document. Typically, you click on the hyperlink to follow the link. Hyperlinks are the most essential ingredient of all hypertext systems, including the World Wide Web.
2. **Web server:** A computer that delivers (serves up) web pages.
3. **Database server:** A computer that hosts the database.
4. **Internet Information Server:** Microsoft's Web server that runs on Windows NT platforms. In fact, IIS comes bundled with Windows NT 4.0. Because IIS is tightly integrated with the operating system, it is relatively easy to administer. However, currently IIS is available only for the Windows NT platform.
5. **Client:** a client is an application that runs on a personal computer or workstation and relies on a server to perform some operations.
6. **Browser:** Short for Web browser, a software application used to locate and display Web pages. The two most popular browsers are Netscape Navigator and Microsoft Internet Explorer.
7. **HTML:** Short for HyperText Markup Language, the authoring language used to create documents on the World Wide Web.
8. **Active Server Pages:** A specification for a dynamically created Web page with a .ASP extension that utilizes ActiveX scripting -- usually VB Script or Jscript code. When a browser requests an ASP page, the Web server generates a page with HTML code and sends it back to the browser.
9. **ODBC:** Abbreviation of Open DataBase Connectivity, a standard database access method developed by Microsoft Corporation. The goal of ODBC is to make it possible to access any data from any application, regardless of which database management system (DBMS) is handling the data. ODBC manages this by inserting a middle layer, called a database driver, between an application and the DBMS. The purpose of this layer is to translate the application's data queries into commands that the DBMS understands. For this to work, both the application and the DBMS must be ODBC-compliant -- that is, the application must be capable of issuing ODBC commands and the DBMS must be capable of responding to them.
10. **URL:** Abbreviation of Uniform Resource Locator, the global address of documents and other resources on the World Wide Web. The first part of the address indicates what protocol to use, and the second part specifies the IP address or the domain name where the resource is located.

11. **E-Commerce:** Conducting business on-line. This includes, for example, buying and selling products with digital cash and via Electronic Data Interchange (EDI).
12. **Database:** A collection of information organized in such a way that a computer program can quickly select desired pieces of data. You can think of a database as an electronic filing system. Traditional databases are organized by fields, records, and files. A field is a single piece of information; a record is one complete set of fields; and a file is a collection of records.
13. **Normalizing:** In relational database design, the process of organizing data to minimize duplication. Normalization usually involves dividing a database into two or more tables and defining relationships between the tables. The objective is to isolate data so that additions, deletions, and modifications of a field can be made in just one table and then propagated through the rest of the database via the defined relationships. There are three main normal forms, each with increasing levels of normalization.
14. **Tables:** Refers to data arranged in rows and columns. In relational database management systems, all information is stored in the form of tables.
15. **Primary Key:** It is a unique value that identifies each record in a table.
16. **Foreign Key:** A key field that identifies records in a different table is called a foreign key.
17. **Referential Integrity:** A feature provided by relational database management systems (RDBMS's) that prevents users or applications from entering inconsistent data. Most RDBMS's have various referential integrity rules that you can apply when you create a relationship between two tables.
18. **Database diagram:** It is a graphical representation of all the tables in a database displaying the relationships among all the tables.
19. **Stored Procedure:** In database management systems (DBMSs), an operation that is stored with the database server. Typically, stored procedures are written in SQL. They're especially important for client-server database systems because storing the procedure on the server side means that it is available to all clients. And when the procedure is modified, all clients automatically get the new version.
20. **Image Map:** Image map is a graphical image that has a set of two-dimensional coordinates associated with the image. For each of the coordinates there is an optional URL associated. When users click somewhere on the image within the bounds of one of the coordinates in the set, the browser sends that user to the associated URL.

21. **Design Time Control (DTC):** DTC controls are form elements that you define while you are editing a Web document during design time. These controls are created using Java scripts and have features that enable you to bind them directly to the database. DTC controls can respond to event messages and perform other tasks that HTML tag controls cannot.
22. **Many-to-Many Relationship:** Many-to-Many relationship is set when you might want to join tables where more than one row matches more than one row in a second table.
23. **Event:** An action or occurrence detected by a program. Events can be user actions, such as clicking a mouse button or pressing a key, or system occurrences, such as running out of memory.
24. **ADO:** Short for ActiveX Data Objects, Microsoft's newest high-level interface for data objects. ADO can be used to access all sorts of different types of data, including web pages, spreadsheets, and other types of documents.
25. **DSN:** In the Windows environment, once you have installed a database (e.g., Access, SQL Server, etc.) an additional step is required to access this data via ODBC. A data source must be added to access the data to provide a "short cut," that allows you to access a database server name, location, database name, and login combination with a single data source name. To add a data source, double-click on the ODBC icon in the Windows Control Panel. There will be several tabs; two of these tabs can be used to add access to data sources. Adding a data source under the "User DSN" tab adds a data source that is only accessible to the current user. Adding a data source under the "System DSN" tab adds a data source that is accessible by any user on that machine.

## References

1. "Analyst: E-commerce growth will beat forecasts"  
<http://news.cnet.com/news/0-1007-200-343198.html?tag=> , June 3 1999.
2. Brockwood, Ted. "Getting Started With SQL".  
<http://www.webdevelopersjournal.com/articles/sql.html>. March 1999.
3. Browning, Bob. "Electronic Commerce Tutorial 1".  
<http://Webdevelopersjournal.com/columns/ecommerce1.html>. September 17, 1998.
4. Browning, Bob. "Electronic Commerce Tutorial 2".  
<http://Webdevelopersjournal.com/columns/ecommerce2.html>. September 17, 1998.
5. Browning, Bob. "Electronic Commerce Tutorial 3".  
<http://Webdevelopersjournal.com/columns/ecommerce3.html>. September 17, 1998.
6. Clarke, Roger. "Roger Clarke's Electronic Commerce Pages"  
<http://www.anu.edu.au/people/Roger.Clarke/EC/index.html>. February 26, 1999.
7. Coulter, Francie and Norman Iannarelli. "Online Bookstores Beat the Real World".  
<http://www.internet.com/corporate/releases/97.09.17-world.html>. September 17, 1997.
8. Cox, Beth. " Survey: Online Shoppers Cut Spending At Traditional Stores".  
<http://internet.com>. June 2, 1999.
9. "Database". [www.webopedia.com](http://www.webopedia.com). January 2000.
10. Dice, Richard. "Choosing the Right Database System".  
[http://hotwired.lycos.com/webmonkey/98/24/index3a\\_page3.html?tw=backend](http://hotwired.lycos.com/webmonkey/98/24/index3a_page3.html?tw=backend).  
February 15,2000
11. Francis, Kauffman, Llibre, Sussman, and Ullman.  
*Beginning active Server Pages 2.0* Birmingham: WROX Press, 1998.
12. Geller, David. "The Seven Deadly Sins of E-commerce"  
[http://www.ecommercetimes.com/small\\_business/getting\\_started/deadly-sins-a.shtml](http://www.ecommercetimes.com/small_business/getting_started/deadly-sins-a.shtml). February 24, 2000.
13. Greenspan, Jay. "Your First Database".  
<http://hotwired.lycos.com/webmonkey/99/13/index0a.html?tw=backend>.  
February 15,2000
14. Griffith, Cary. "ASP technology: pros and cons".  
*Cincinnati Computer User*. January, 2000. 16.

15. Higgins, Ron. "E-Commerce: Report From The Trenches,"  
<http://www.beyondcomputingmag.com>. September 1999.
16. Hilderbrandt, David. "10 Tips for e-transformation,"  
<http://www.beyondcomputingmag.com/archive/1999/08-99/frontline.html>.  
October, 1999.
17. Matozzo, Michele. "Online Bookstore Story".  
<http://www.aip.org/epr/sep95/article7.html>. September, 1995.
18. *Microsoft Visual InterDev 6.0 Programmer's Guide*  
Washington: Microsoft Press, 1998.
19. Morris, Charlie. "Developing Databases For the Web and Intranets".  
[http://www.webdevelopersjournal.com/books/developing\\_databases.html](http://www.webdevelopersjournal.com/books/developing_databases.html).  
October 27, 1999
20. Morris, Bruce. "Developing Large Commerce-Oriented Web Sites".  
<http://Webdevelopersjournal.com/articles/commerce.htm>. October, 1997.
21. Mueller Paul and Tom Sheldon. *Microsoft Internet Information Server 4*  
California: Osborne, 1998.
22. Murray, James. "Shopping Cart Software".  
<http://Webdevelopersjournal.com/software/shopping.html>. October 13, 1997.
23. Reinshagen, Dirk. "Inside ASP"  
<http://www.Webreview.com/pub/97/11/21/feature/sidebar.html>.  
November 27, 1997.
24. "Relational Database". [www.whatis.com](http://www.whatis.com). February 2000.
25. Rosenfeld, Louis and Peter Morville.  
*Information Architecture for the World Wide Web*.  
California: O'Reilly & Associates, 1998.
26. Special CNET News "Analyst: E-commerce growth will beat forecasts"  
<http://www.news.com/News/item/0,4,37350,00.html>. June 3, 1999.
27. "Sun Guide to Web Style".  
[http://www.sun.com/styleguide/others/Printing\\_Version.html](http://www.sun.com/styleguide/others/Printing_Version.html) . June 1999.
28. "The Duwamish Books Sample Application".  
<http://msdn.microsoft.com/voices/sampleapp.asp>. May, 1999.

29. Udell, Jon. "Build it Yourself". *The Industry Standard*.  
November 16, 1998. 62 .
30. Veen, Jeffrey. "The Foundations of Web Design".  
<http://hotwired.lycos.com/webmonkey/98/07/index0a.html>. February 15,2000
31. Weber, Jonathan. "In Search of The One Stop Shop". *The Industry Standard*.  
November 16, 1998. 32-33 .
32. Weinschenk, Susan., Pamela Jamar, and Sarah Yeo. *GUI design Essentials* .  
New York : John Wiley & Sons, Inc., 1997
33. "What exactly are Active Server Pages".  
<http://4guysfromrolla.com/Webtech/faq/Beginner/faq1.shtml>. February, 2000.
34. Wreden, Nick. "E-commerce: A Marketing Revolution,"  
<http://www.beyondcomputingmag.com> . January/February 1997.
35. Wreden, Nick. "Net Effects: The New E-Business Model"  
<http://www.beyondcomputingmag.com> . July/August 1999.
36. Wreden, Nick. "E-business: Technology Unites the World"  
<http://www.beyondcomputingmag.com>. June 1998.
37. Wong, Chee "Building a Simple E-commerce Web site"  
[http://webdevelopersjournal.com/articles/online\\_store\\_builder.html](http://webdevelopersjournal.com/articles/online_store_builder.html)
38. Wynkoop, Stephen. *Using Microsoft SQL Server*.  
Indiana : QUE , 1999.

Primarily, my focus was on the other existing Web sites of online bookstores like [www.amazon.com](http://www.amazon.com), [www.barnesandnoble.com](http://www.barnesandnoble.com), [www.borders.com](http://www.borders.com), and [www.books.com](http://www.books.com).