Online Transactions Using Asp.net & Ado.net

RamaRao Gose

Assoc .Prof in IT Dept, Cjits,jangaon, ap.india

Abstract: The business-to-consumer aspect of electronic commerce (e-commerce) is the most visible business use of the World Wide Web. The primary goal of an e-commerce site is to sell goods and services online.

This project deals with developing an e-commerce website for Online Book Sale. It provides the user with a catalog of different books available for purchase in the store. In order to facilitate online purchase a shopping cart is provided to the user. The system is implemented using a 3-tier approach, with a backend database, a middle tier of Microsoft Internet Information Services (IIS) and ASP.NET, and a web browser as the front end client.

In order to develop an e-commerce website, a number of Technologies must be studied and understood. These include multi-tiered architecture, server and client side scripting techniques, implementation technologies such as ASP.NET, programming language (such as C#, VB.NET), relational databases (such as MySQL, Access).

This is a project with the objective to develop a basic website where a consumer is provided with a shopping cart application and also to know about the technologies used to develop such an application.

This document will discuss each of the underlying technologies to create and implement an e-commerce website.

Keywords: e-commerce-online-asp.net-ado.net-server-client-mysql-html-

1. Introduction

E-commerce is fast gaining ground as an accepted and used business paradigm. More and more business houses are implementing web sites providing functionality for performing commercial transactions over the web. It is reasonable to say that the process of shopping on the web is becoming commonplace.

The objective of this project is to develop a general purpose e-commerce store where any product (such as books, CDs, computers, mobile phones, electronic items, and home appliances) can be bought from the comfort of home through the Internet. However, for implementation purposes, this paper will deal with an online book store.

An online store is a virtual store on the Internet where customers can browse the catalog and select products of interest. The selected items may be collected in a shopping cart. At checkout time, the items in the shopping cart will be presented as an order. At that time, more information will be needed to complete the transaction. Usually, the customer will be asked to fill or select a billing address, a shipping address, a shipping option, and payment information such as credit card number. An e- mail notification is sent to the customer as soon as the order is placed.

2. Literature Review

Electronic Commerce (e-commerce) applications support the interaction between different parties participating in a commerce transaction via the network, as well as the management of the data involved in the process [2].

The increasing importance of e-commerce is apparent in the study conducted by researchers at the GVU (Graphics, Visualization, and Usability) Center at the Georgia Institute of Technology. In their summary of the findings from the eighth survey, the researchers report that “e-commerce is taking off both in terms of the number of users shopping as well as the total amount people are spending via Internet based transactions”.

Over three quarters of the 10,000 respondents report having purchased items online. The most cited reason for using the web for personal shopping was convenience (65%), followed by availability of vendor information (60%), no pressure form sales person (55%) and saving time (53%).

Although the issue of security remains the primary reason why more people do not purchase items online, the GVA survey also indicates that faith in the security of e-commerce is increasing. As more people gain confidence in current encryption technologies, more and more users can be expected to frequently purchase items online [11].

A good e-commerce site should present the following factors to the customers for better usability [11]:

- Knowing when an item was saved or not saved in the shopping cart.
- Returning to different parts of the site after adding an item to the shopping cart.
- Easy scanning and selecting items in a list.
- Effective categorical organization of products.
- Simple navigation from home page to information and order links for specific products.
- Obvious shopping links or buttons.
- Minimal and effective security notifications or messages.
- Consistent layout of product information.

Another important factor in the design of an e-commerce site is feedback [4]. The interactive cycle between a user and a web site is not complete until the web site responds to a command entered by the user. According to Norman [5], "feedback--sending back to the user informa tion about what action has actually been done, what result has been accomplished--is a well-known concept in the science of control and information theory. Imagine trying to talk to someone when you cannot even hear your own voice, or trying to draw a picture with a pencil that leaves no mark: there would be no feedback".

Web site feedback often consists of a change in the visual or verbal information presented to the user. Simple examples include highlighting a selection made by the user or filling a field on a form based on a user's selection from a pull-down list. Another example is using the sound of a cash register to confirm that a product has been added to an electronic shopping cart.

Completed orders should be acknowledged quickly. This may be done with an acknowledgment or fulfillment page. The amount of time it takes to generate and download this page, however, is a source of irritation for many e-commerce users. Users are quick to attribute meaning to events. A blank page, or what a user perceives to be "a long time" to receive an acknowledgment, may be interpreted as "there must be something wrong with the order." If generating an acknowledgment may take longer than what may be reasonably expected by the user, then the design should include intermediate feedback to the user indicating the progress being made toward acknowledgment or fulfillment.

Finally, feedback should not distract the user. Actions and reactions made by the web site should be meaningful. Feedback should not draw the user's attention away from the important tasks of gathering information, selecting products, and placing orders.

3. Implementation Technologies

The objective of this project is to develop an online book store. When the user types in the URL of the Book Store in the address field of the browser, a Web Server is contacted to get the requested information. In the .NET Framework, IIS (Internet Information Service) acts as the Web Server. The sole task of a Web Server is to accept incoming HTTP requests and to return the requested resource in an HTTP response. The first thing IIS does when a request comes in is to decide how to handle the request. Its decision is based upon the requested file's extension. For example, if the requested file has the .asp extension, IIS will route the request to be handled by asp.dll. If it has the extens ion of .aspx, .ascx, etc, it will route the request to be handled by ASP.NET Engine.

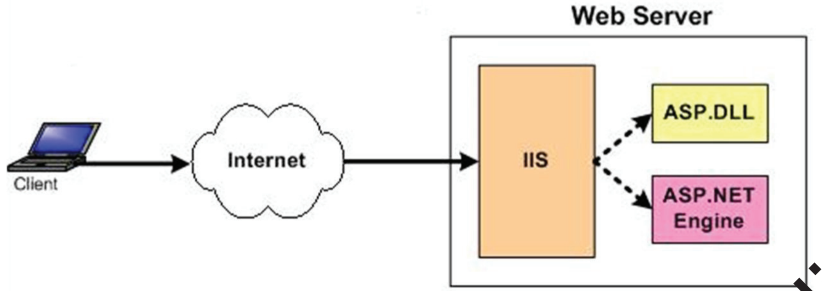

Figure 21 Relations between IIS and ASP.NET

The ASP.NET Engine then gets the requested file, and if necessary contacts the database through ADO.NET for the required file and then the information is sent back to the Client's browser. Figure 21 shows how a client browser interacts with the Web server and how the Web server handles the request from client.

1. Internet Information Services (IIS)

IIS is a set of Internet based services for Windows machines. Originally supplied as part of the Option Pack for Windows NT, they were subsequently integrated with Windows 2000 and Windows Server 2003). The current (Windows 2003) version is IIS 6.0 and includes servers for **FTP** (a software standard for transferring computer files between machines with widely different operating systems), **SMTP** (Simple Mail Transfer Protocol, is the de facto standard for email transmission across the Internet) and **HTTP/HTTPS** (is the secure version of HTTP, the communication protocol of the World Wide Web) [12].

Features: The web server itself cannot directly perform server side processing but can delegate the task to ISAPI (Application Programming Interface of IIS) applications on the server. Microsoft provides a number of these including ones for Active Server Page and ASP.NET.

Compatibility: Internet Information Services is designed to run on Windows server operating systems. A restricted version that supports one web site and a limited number of connections is also supplied with Windows XP Professional.

Microsoft has also changed the server account that IIS runs on. In versions of IIS before 6.0, all the features were run on the System account, allowing exploits to run wild on the system. Under 6.0 many of the processes have been brought under a Network Services account that has fewer privileges. In particular this means that if there were an exploit on that feature, it would not necessarily compromise the entire system.

2 ASP.NET

ASP.NET is a programming framework built on the common language runtime that can be used on a server to build powerful Web applications. ASP.NET has many advantages – both for programmers and for the end users because it is compatible with the .NET Framework. This compatibility allows the users to use the following features through ASP.NET:

a) **Powerful database-driven functionality:** ASP.NET allows programmers to develop web applications that interface with a database. The advantage of ASP.NET is that it is object-oriented and has many programming tools that allow for faster development and more functionality.

b) **Faster web applications:** Two aspects of ASP.NET make it fast -- compiled code and caching. In ASP.NET the code is compiled into "machine language" *before* a visitor ever comes to the website. Caching is the storage of information in memory for faster access in the future. ASP.NET allows programmers to set up pages or areas of pages that are commonly reused to be cached for a set period of time to improve the performance of web applications. In addition, ASP.NET allows the caching of data from a database so the website is not slowed down by frequent visits to a database when the data does not change very often.

c) **Memory leak and crash protection:** ASP.NET automatically recovers from memory leaks and errors to make sure that the website is always available to the visitors.

ASP.NET also supports code written in more than 25 .NET languages (including VB.NET, C#, and Jscript.Net). This is achieved by the Common Language Runtime (CLR) compiler that supports multiple languages.

3.2.1. Authentication in ASP.NET

There are two separate authentication layers in an ASP.NET application. All requests flow through IIS before they are handed to ASP.NET, and IIS can decide to deny access before ASP.NET even knows about the request. Here is how the process works [14]:

1. IIS checks to see if an incoming request is coming from an IP address that is allowed access to the domain. If not, the request is denied.
2. IIS performs its own user authentication, if it is configured to do so. By default, IIS allows anonymous access and requests are authenticated automatically.
3. When a request is passed from IIS to ASP.NET with an authenticated user, ASP.NET checks to see whether impersonation is enabled. If so, ASP.NET acts as though it were the authenticated user. If not, ASP.NET acts with its own configured account.
4. Finally, the identity is used to request resources from the operating system. If all the necessary resources can be obtained, the user's request is granted; otherwise the request is denied.

3.3. MySQL Database

In this project, MySQL is used as the backend database. MySQL is an open-source database management system. The features of MySQL are given below:

- MySQL is a relational database management system. A relational database stores information in different tables, rather than in one giant table. These tables can be referenced to each other, to access and maintain data easily.
- MySQL is open source database system. The database software can be used and modify by anyone according to their needs.
- It is fast, reliable and easy to use. To improve the performance, MySQL is multi-threaded database

engine. A multithreaded application performs many tasks at the same time as if multiple instances of that application were running simultaneously.

In being multithreaded MySQL has many advantages. A separate thread handles each incoming connection with an extra thread that is always running to manage the connections. Multiple clients can perform read operations simultaneously, but while writing, only hold up another client that needs access to the data being updated. Even though the threads share the same process space, they execute individually and because of this separation, multiprocessor machines can spread the thread across many CPUs as long as the host operating system supports multiple CPUs. Multithreading is the key feature to support MySQL’s performance design goals. It is the core feature around which MySQL is built.

MySQL database is connected to ASP.NET using an ODBC driver. Open Database Connectivity (ODBC) is a widely accepted application-programming interface (API) for database access. The ODBC driver is a library that implements the functions supported by ODBC API. It processes ODBC function calls, submits SQL requests to MySQL server, and returns results back to the application. If necessary, the driver modifies an application's request so that the request conforms to syntax supported by MySQL.

4. Integrating IIS and ASP.NET

When a request comes into IIS Web server its extension is examined and, based on this extension, the request is either handled directly by IIS or routed to an ISAPI extension. An ISAPI extension is a compiled class that is installed on the Web server and whose responsibility is to return the markup for the requested file type. By default, IIS handles the request, and simply returns the contents of the requested file [13].

This makes sense for static files, like images, HTML pages, CSS files, external JavaScript files, and so on. For example, when a request is made for a .html file, IIS simply returns the contents of the requested HTML file.

For files whose content is dynamically generated, the ISAPI extension configured for the file extension is responsible for generating the content for the requested file. For example, a Web site that serves up classic ASP pages has the .asp extension mapped to the asp.dll ISAPI extension. The asp.dll ISAPI extension executes the requested ASP page and returns its generated HTML markup. If the Web site serves up ASP.NET Web pages, IIS has mapped the .aspx to aspnet_isapi.dll, an ISAPI extension that starts off the process of generating the rendered HTML for the requested ASP.NET Web page.

The aspnet_isapi.dll ISAPI extension is a piece of *unmanaged code*. That is, it is not code that runs in the .NET Framework. When IIS routes the request to the aspnet_isapi.dll ISAPI extension, the ISAPI extension routes the request onto the ASP.NET engine, which is written in *managed code* - managed code is code that runs in the .NET Framework.

The ASP.NET engine is strikingly similar to IIS in many ways. Just like IIS has a directory mapping file extensions to ISAPI extensions, the ASP.NET engine maps file extensions to *HTTP handlers*. An HTTP handler is a piece of managed code that is responsible for generating the markup for a particular file type.

5. Database Connectivity

In e-commerce applications it is very typical for the Web server to contact the database to get information as needed. ASP.NET uses a technology called ActiveX Data Objects.NET (ADO.NET) to connect to the database.

5.1 ADO.NET

Classic ASP pages used ActiveX Data Objects (ADO) to access and modify databases. ADO is a programming interface used to access data. This method was efficient and fairly easy for developers to learn and implement. However, ADO suffered from a dated model for data access with many limitations, such as the inability to transmit data so it is easily and universally accessible. Coupled with the move from standard SQL databases to more distributed types of data (such as XML), Microsoft introduced ADO.NET.

Although ADO.NET is known as the next evolution of ADO, it is very different from its predecessor. Whereas ADO was connection-based, ADO.NET relies on short, XML message-based interactions with data sources. This makes ADO.NET much more efficient for Internet-based applications.

A fundamental change from ADO to ADO.NET was the adoption of XML for data exchanges. XML is a text-based markup language, similar to HTML that presents an efficient way to represent data. This allows ADO.NET to reach and exchange. It also gives ADO.NET much better performance because XML data is easily converted to and from any type of data.

Another major change is the way ADO.NET interacts with databases. ADO requires "locking" of database resources and lengthy connections for its applications, but ADO.NET does not; it uses disconnected data sets, which eliminates lengthy connections and database locks. This makes ADO.NET much more scalable because users are not in contention for database resources.

In ADO.NET there are two core objects that allow us to work with data initially: the DataReader and the DataSet. In any .NET data access page, before we connect to a database, we first have to import all the necessary namespaces that will allow us to work with the objects required. Namespace in .NET is a set of classes that can be used while creating an application. The .NET Framework has about 3,500 classes which can be accessed through a namespace. The application will be using a technology known as Open DataBase Connectivity (ODBC) to access the database; therefore we must first import necessary namespaces. Below is a sample namespace declaration used by .NET.

```
<%@ Import Namespace="System" %>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.Odbc" %>
```

After all the necessary namespaces are imported, a connection to the database is made.

```
OdbcConnetion odbcCon = new OdbcConnection ("DRIVER = {MySQL ODBC 3.51
                              Driver}; SERVER=localhost; DATABASE=project;
                              UID=root; PASSWORD=pwd");
odbcCon.Open();
```

The above statement creates a connection to the database with an OdbcConnection object. This object tells ASP.NET where to go to get the data it needs. Since the data is stored in the same computer as the application, the SERVER is given as *localhost*. Next we open the connection object. Listed below are the common connection object methods we could work with:

- ***Open*** - Opens the connection to our database
- ***Close*** - Closes the database connection
- ***Dispose*** - Releases the resources on the connection object. Used to force garbage collecting, ensuring no resources are being held after our connection is used.
- ***State*** - Tells you what type of connection state your object is in, often used to check whether the connection is still using any resources.

Once the connection is made, in order to access the data in a database, ADO.NET relies on two components: DataSet and Data Provider [20]. These components are explained below.

Dataset

The dataset is a disconnected, in- memory representation of data. It can be considered as a local copy of the relevant portions of the database. The DataSet resides in memory and the data in it can be manipulated and updated independent of the database. If necessary, changes made to the dataset can be applied to the central database. The data in DataSet can be loaded from any valid data source such as a text file, an XML database, Microsoft SQL server database, an Oracle database or MySQL database.

Data Provider

The Data Provider is responsible for providing and maintaining the connection to the database. A DataProvider is a set of related components that work together to provide data in an efficient and performance driven manner. Each DataProvider consists of the following component classes:

- The Connection object which provides a connection to the database
- The Command object which is used to execute a command
- The DataReader object which provides a read only, connected recordset
- The DataAdapter object which populates a disconnected DataSet with data and performs the update.

The Connection Object

The Connection object creates the connection to the database. Microsoft Visual Studio .NET provides two types of Connection classes: the SqlConnection object, which is designed specifically to connect to Microsoft SQL Server 7.0 or later, and the OleDbConnection object, which can provide connections to a wide range of database types like Microsoft Access and Oracle. The Connection object contains all of the information required to open a connection to the database.

The Command Object

The Command object is represented by two corresponding classes: SqlCommand and OleDbCommand. Command objects are used to execute commands to a database across a data connection. The Command objects can be used to execute stored procedures on the database, SQL commands, or return complete tables directly. Command objects provide three methods that are used to execute commands on the database:

ExecuteNonQuery: Executes commands that have no return values such as INSERT, UPDATE or DELETE.
ExecuteScalar: Returns a single value from a database query
ExecuteReader: Returns a result set by way of a Data Reader object

The DataReader Object

The DataReader object provides a read-only, connected stream recordset from a database. Unlike other components of the Data Provider, DataReader objects cannot be directly instantiated. Rather, the DataReader is returned as the result of the Command object's ExecuteReader method. The SqlCommand.ExecuteReader method returns a SqlDataReader object, and the OleDbCommand.ExecuteReader method returns an OleDbDataReader object. The DataReader can provide rows of data directly to application logic when one does not need to keep the data cached in memory. Because only one row is in memory at a time, the DataReader provides the lowest overhead in terms of system performance but requires the exclusive use of an open Connection object for the lifetime of the DataReader.

The DataAdapter Object

The DataAdapter is the class at the core of ADO .NET's disconnected data access. It is essentially the middleman facilitating all communication between the database and a DataSet. The DataAdapter is used either to fill a DataTable or DataSet with its Fill method. After the memory-resident data has been manipulated, the DataAdapter can commit the changes to the database by calling the Update method. The DataAdapter provides four properties that represent database commands:

Select Command
Insert Command
Delete Command
Update Command

When the Update method is called, changes in the DataSet are copied back to the database and the appropriate InsertCommand, DeleteCommand, or UpdateCommand is executed. .NET follows the below process, Figure 24, to connect to the database and retrieve data to the application [21]

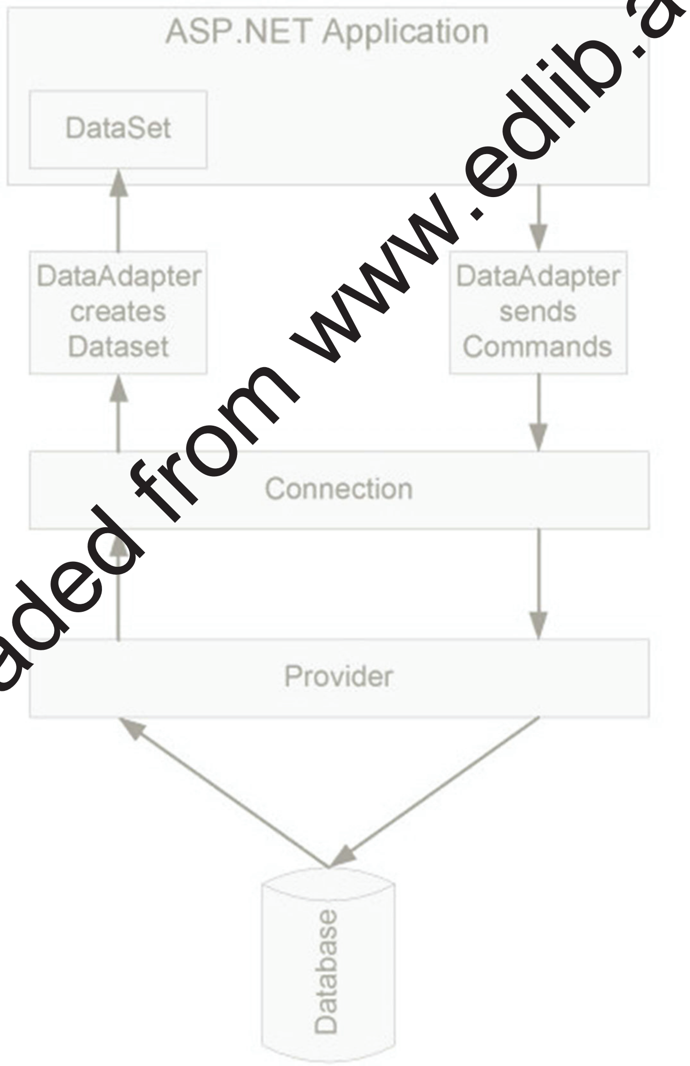

Figure 24 ADO.NET Architecture

- When an ASP.NET application needs to access the database, it submits an appropriate request to ADO.NET through a DataAdapter object, which in turn sends a command to the Connection object.
- The Connection object establishes a connection to the database and submits the request sent by DataAdapter.
- The Connection object connects to the database through a Provider such as ODBC.NET. The Provider acts as a translator between the Connection object and the database. It translates the request for data to database’s language and brings back the data, if needed.
- The Provider sends the data back to the DataAdapter through the Connection object and DataAdapter places the data in a DataSet object residing in application’s memory.

Instead of storing data in a DataSet, a DataReader can be used to retrieve data from the database. Results are returned in a resultset which is stored in the network buffer on the client until a request is made to **Read** method of the DataReader. Using the DataReader can increase the application performance by retrieving as soon as the data is available, rather than waiting for the entire results of the query to be returned [22].

A DataSet can be used to interact with data dynamically such as binding to a Web Form, cache locally in the application, provide hierarchical XML view of the data, etc. If such functionalities are not required by the application, a DataReader can be used to improve the performance of the application. By using a DataReader, the memory can be saved that is used by the DataSet, as well as the processing required to Fill the contents of a DataSet.

When a DataReader is used, a DataAdapter is not required to send the data to the application. In this project, DataReader is used to read the data and Command object called ExecuteNonQuery is used to write into the database.

5.2 Connecting ASP.NET application to a Database

The steps required to connect our ASP.NET application to the MySQL database and access the data are given below:

1. Import the required namespaces.

```
using System;
using System.Data;
using System.Data.Odbc;
```

2. Create a connection object.

```
string myConnectionString;
myConnectionString = “DRIVER = {MySQL ODBC 3.51 Driver}; SERVER = localhost;
                      DATABASE = project; UID = root; PASSWORD = ‘”
OdbcConnection odbcCon = new odbcConnection(myConnectionString)
```

3. Create a SQL query

```
string str;
str="Select * from Customer where UserID='admin’;
```

4. Create a Command object to run the SQL query

```
        odbcCmd=new OdbcCommand(str,odbcCon);
```

5. DataReader to read the result

```
        OdbcDataReader odbcReader;
        String text, text2;
        while (odbcReader.Read())
        {
                text = odbcReader["UserID"].ToString();
                text2 = odbcReader[“FirstName”].ToString();
        }
```

6. Close odbcReader and odbcConnection

```
        odbcReader.Close();
        odbcCon.Close();
```

The data can now be used as desired by the application.

6. Limitations and Future Development

There are some limitations for the current system to which solutions can be provided as a future development:

1. The system is not configured for multi- users at this time. The concept of *transaction* can be used to achieve this.
2. The Website is not accessible to everyone. It can be deployed on a web server so that everybody who is connected to the Internet can use it.
3. Credit Card validation is not done. Third party proprietary software can be used for validation check.

As for other future developments, the following can be done:

1. The Administrator of the web site can be given more functionalities, like looking at a specific customer’s profile, the books that have to be reordered, etc.
2. Multiple Shopping carts can be allowed.

7. Conclusion

The Internet has become a major resource in modern business, thus electronic shopping has gained significance not only from the entrepreneur’s but also from the customer’s point of view. For the entrepreneur, electronic shopping generates new business opportunities and for the customer, it makes comparative shopping possible. As per a survey, most consumers of online stores are impulsive and usually make a decision to stay on a site within the first few seconds. “Website design is like a shop interior. If the shop looks poor or like hundreds of other shops the customer is most likely to skip to the other site”[16]. Hence we have designed the project to provide the user with easy navigation, retrieval of data and necessary feedback as much as possible.

In this project, the user is provided with an e-commerce web site that can be used to buy books online. To implement this as a web application we used ASP.NET as the Technology. ASP.NET has several advantages such as enhanced performance, scalability, built- in security and simplicity. To build any web application using ASP.NET we need a programming language such as C#, VB.NET, J# and so on. C# was the language used to build this application. For the client browser to connect to the ASP.NET engine we used Microsoft’s

Internet Information Services (IIS) as the Web Server. ASP.NET uses ADO.NET to interact with the database as it provides in- memory caching that eliminates the need to contact the database server frequently and it can easily deploy and maintain an ASP.NET application. MySQL was used as back-end database since it is one of the most popular open source databases, and it provides fast data access, easy installation and simplicity.

A good shopping cart design must be accompanied with user-friendly shopping cart application logic. It should be convenient for the customer to view the contents of their cart and to be able to remove or add items to their cart. The shopping cart application described in this project provides a number of features that are designed to make the customer mo re comfortable.

This project helps in understanding the creation of an interactive web page and the technologies used to implement it. The design of the project which includes Data Model and Process Model illustrates how the database is built with different tables, how the data is accessed and processed from the tables. The building of the project has given me a precise knowledge about how ASP.NET is used to develop a website, how it connects to the database to access the data and how the data and web pages are modified to provide the user with a shopping cart application.

Acknowledgments

In completing this graduate project I have been fortunate to have help, support and encouragement from many people. I would like to acknowledge them for their cooperation.

First, I would like to thank HARISH REDDY, my project advisor, for guiding me through each and every step of the process with knowledge and support. Thank you for your advice, guidance and assistance.

I would also like to thank SALMON, my project committee members, who showed immense patience and understanding throughout the project and provided suggestions.

8. Bibliography

Articles

1. Chen, L. (2000). Enticing Online Consumers: A Technology Acceptance Perspective Research- in-Progress. *ACM Proceedings, SIGCPR*.
2. Diwakar, H., Marathe, M. (2000). The architecture of a one-stop web-window shop. December, *ACM SIGecom Exchanges*, Volume 2 Issue 1.
3. orrison, M., Morrison, J., and Keys, A. (2002). Integrating Web Sites and Databases. *Communications of the ACM*, September, Volume 45, Issue 9.
4. Kubilus, N. J. (2000). Designing an e-commerce site for users. September 2000, *Crossroads*, Volume 7 Issue 1.
5. Norman, D.A. The Design of Everyday Things. *Doubleday*, New York, 1994.
6. Tilson, R., Dong, J., Martin, S., Kieke, E. (1998). A comparison of two current e- commerce sites. September, *Proceedings of the 16th annual international conference on Computer documentation*.

Books

7. Anderson, R., Francis, B., Homer, A., Howard, R., Sussman, D. and Watson. (2001) *Professional ASP.NET*. Wrox Press Ltd.
8. Brown, S., Burdick, R., Falkner, J., Galbraith, B., Johnson, R., Kim, L., Kochmer, C., Kristmundsson, T. and Li S (2001). *Professional JSP*. Wrox Press Ltd.
9. Walther, S. (1998) *Active Server Pages*. SAMS Net.
10. Wagner, R., Daniels, K., Griffin, G., Haddad, C. and Nasr, J. (1997) *JavaScript Unleashed*. SAMS

Net.

11. Wiley, Y. M. J. & Sons. (1997) *Creating the Virtual Store: Taking Your Web Site from Browsing to Buying.*

Websites

12. http://encyclopedia.laborlawtalk.com/IIS for information on IIS
13. http://aspnet.4guysfromrolla.com/articles/020404-1.aspx for relationship between IIS and ASP.NET.
14. http://216.15.201.66/dpec/course.htm?fullpg=http%3A//216.15.201.66/dpec/cours es/wac312/waho06.htm&acro=wac312 for security authentication in ASP.NET
15. http://samples.gotdotnet.com/quickstart/aspplus/doc/mtstransactions.aspx for information on Transactions in ASP.NET.
16. http://www.x-cart.com/articles/design_development.html for online customer behavior.
17. http://aspnet.4guysfromrolla.com/articles/011404-1.aspx for relation between IIS and ASP.NET.
18. http://www.informatik.uni-bremen.de/uniform/gdpa_d/methods/m- fctd.htm for definition of Functional Decomposition.
19. http://www.agilemodeling.com/artifacts/dataFlowDiagram.htm for definition of Data Flow Diagram.
20. http://www.startvbdotnet.com/ado/default.aspx for information on ADO.NET
21. http://mypage.iusb.edu/~hhakimza/505/index.html for ADO.NET objects.
22. http://msdn.microsoft.com for ADO.NET objects.