

Teaching ASP.NET in an Information Systems Curriculum

Dr. David C. Wallace
dcwalla@ilstu.edu

James R. Wolf Jr.
jrwolf@ilstu.edu
School of Information Technology
Illinois State University
Normal, Illinois 61790-5150, USA

Abstract

The challenge of providing computer science graduates with the optimal balance between the old and new technology is a never ending process. This paper introduces Microsoft's ASP.Net technology which can be used as a bridge to integrate mainframe and client/server technologies. The paper also describes practical ways to use this bridge technology in the classroom and discusses our experiences using ASP.NET in a recent course.

(**Keywords:** Internet, Web, WWW, Mainframe, E-Commerce and ASP.NET)

1. INTRODUCTION

While many universities have eliminated, or are phasing out, their mainframe-related offerings and a growing number of today's post-secondary students perceive mainframe computing as a relic from the past, according to Durward et al. (2004), 60 percent of business data assets are still stored in mainframe computers. According to Krill (2004), there are over 200 billion lines of COBOL code currently used in business applications and the total is growing. Balasubramanian (2004) estimates that more than \$1.5 trillion (U.S.) has been invested in COBOL applications worldwide and that more COBOL transactions (30 billion) than webpage hits, take place every day.

According to Koma (2003) and Thibodeau (2005), mainframes continue to remain a vital tool for industry, the pool of skilled mainframe personnel is relatively small and the number is getting smaller each year. Despite the growth of client/server and Web based applications,

mainframes have evolved into high-end servers integrating themselves into a dynamic client/server, web based, and mainframe application environment. This result creates a level of stability and dependability that make IT infrastructures highly dependable. Unfortunately, more than half of the enterprise IT managers are over the age of 50 and will retire within the next five to seven years. The resulting exodus of older IT workers will create a large skill gap that technology institutes, colleges, and universities will not be able to fulfill.

At the same time, the importance of Internet technologies has exploded in recent years. Consumer shopping on the Internet is expected to grow to around \$1 trillion by the end of this year, and Electronic Data Interchange (EDI) is expected to grow to around \$500 billion (United States Department of Commerce, 2004). This global growth in Internet-based commerce creates a demand for IS personnel that can develop Internet applications, and these sectors are pressuring academic institutions to provide IT professionals with the skills

that are needed for this rapidly changing technological environment.

To keep pace with industry demands, academic institutions are continuously revamping their curriculum to accommodate the latest information technology developments. The desire to keep up with the latest Internet technologies becomes a tight-rope walk as universities attempt to integrate new technologies while maintaining offerings in the older, but still needed, mainframe technologies.

As the technology evolves, the mainframe and Internet (client/server) environments will integrate into one complete, comprehensive system. Skills developed in the more traditional mainframe environment will need to be incorporated into a complex networking system that can seamlessly transfer information from one environment to another without giving up security or efficiency. Microsoft's .NET technology has been one of the leaders in developing the bridging tools needed to integrate these two environments.

IT professionals need to possess the skills for both environments as organizations seek individuals who can play the role of an integrator and lead complex project teams requiring skill levels from a wide variety of IS resources. According to Murach (2005), there is general agreement among trainers that the best way to prepare a programmer for the real world is to provide them with an understanding of: 1) Java, 2) .NET with C# or Visual Basic, and 3) COBOL/CICS on the mainframe (Martens 2005).

With limited course offerings, institutions must be able to incorporate the client-server technologies of the Internet with the traditional mainframe technologies and the bridging technology that facilitate interaction.

This primary objective of this paper is to introduce Microsoft's ASP.NET technology, which can be used as a bridge to integrate mainframe and client/server technologies, as a teaching tool. We describe practical ways to use this bridge technology in the classroom and discuss our experiences using ASP.NET in a recent course. The remainder of the paper is structured as follows: The next section describes how to setup the client side software, Windows XP and .NET as well as Microsoft's IIS and Visual Studio.NET, the software for the sever

side. The third section describes the course content and the ASP.NET assignments from a recent course where we used ASP.NET. The fourth section describes a short survey we conducted to gauge students' reaction to the course material and assignments. We conclude by summarizing our course experiences with lessons learned and suggests ways to avoid some common problems our students encountered.

2. SOFTWARE SETUP

Both authors are faculty members at a large public university in the Midwestern United States which offered a senior-level topics course in web site development using ASP.NET for the 2003/2004 academic year. Visual Studio.Net was used as the integrated development platform because of the ease in which it allows both client-side and server-side processing on a single personal computer. Microsoft's Internet Information Services (IIS) web server works with Visual Studio.Net for program development as well as testing and debugging of the program. In addition, Visual Studio.NET includes features for stepping through the logic of a program, line-by-line, and gives students the ability to set break points and watch variables during execution. Both capabilities enhance student learning.

Since many of the students who participated in this class owned their own computers, they were able to use the university's software agreement with Microsoft to acquire Visual Studio.NET.

2.1. Windows XP and .NET Setup

It is important to stress that the order of installation for the proper operations of IIS and Visual Studio.NET. Since IIS is not normally installed on computers when they are purchased or even when Windows XP professional is installed, the first thing students must do is to go to Add/Remove Programs in the Control Panel of the Start Menu and select Windows Components. Figure 1 shows how the students can check to see if IIS is installed on their computer. If IIS is not installed, students can simply select IIS Details (Figure 2) and install the IIS Snap-in.



Figure 1. Windows Components

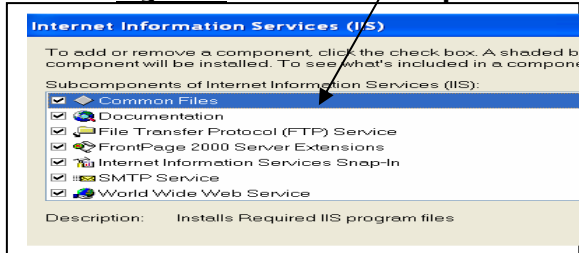


Figure 2. IIS Details

When students select the Next button, Windows will ask them to place their Windows XP professional CD in the appropriate drive. Once IIS is installed, students can install the Visual Studio.NET software. It is important to note that IIS must be installed before Visual Studio.

ASP.NET requires a basic understanding of the windows operating system including the ability to assign the appropriate permissions to files and folders. This is especially true for users of Windows XP professional. Prior to installing the software, the instructor provided a short introduction to Windows XP professional focusing on the security aspects of Windows XP. We observed that the students' computers were often setup with defaults that prevented the students from changing the security settings on the files and folders within their system. In order to complete the course exercises, the students must have administrative privileges on their computers. In order to check the security aspects of files and folders, the student should open Windows Explorer. Under Tools, the student will find folder options listed on a drop-down menu. Students should select Folder Options and locate the View tab. Figure 3 illustrates the various options associated with files and folders. The simple file sharing option is located near the bottom of the list. Students should de-select this option in order to control the file sharing and privileges on files and folders.

After following these setup procedures, students should check to see if they can change the sharing and permissions on their files and folders. To do this, they need to select Windows Explorer and then locate the directory that will contain their first project. In our case, the instructor provided the students with the necessary folder labels for the projects. Within Windows Explorer, right click the project folder and choose the appropriate properties from the drop-down menu. Figure 4 illustrates the properties folder before the change and after de-selecting the simple file sharing box.

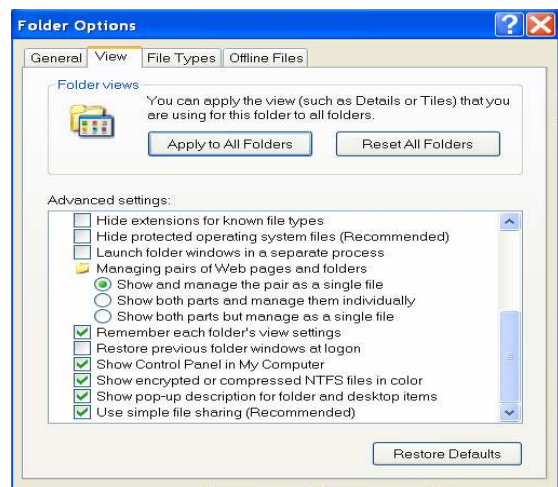


Figure 3. File and Folder Options

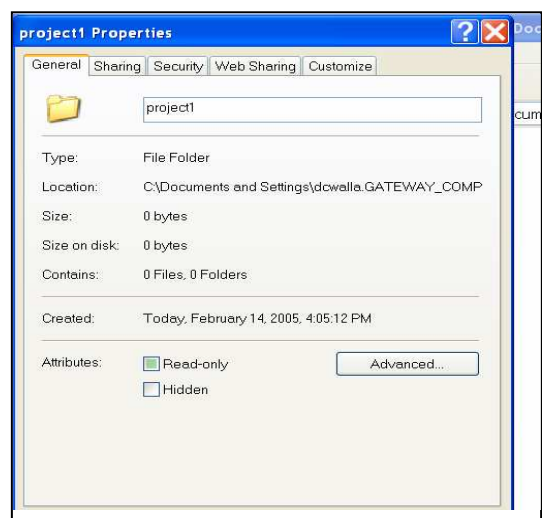


Figure 4. Security Properties

Once the students are able to see the security tab, they can set the permissions and file sharing characteristics of the project folders and the files within the folders. Figure 5 illustrates the user and group permissions available. Without the ability to set permissions and file sharing characteristics, the students will not be able to test their web sites because the web server will not grant them the required access to files, folders and databases contained within these folders. If these permissions are not set correctly, students will get "access denied" errors and "unupdatable database" errors from the web server.

Within the user and group box, students should be able to see the Administrator at this time. Next, students should add a group within Windows XP called "aspnet" and the group should be given Full Control. This can be accomplished by simply selecting the Add button; and on the next screen typing "aspnet". These groups should appear within the box as illustrated in Figure 5. After completing this step, students should select the Full Control button for aspnet.

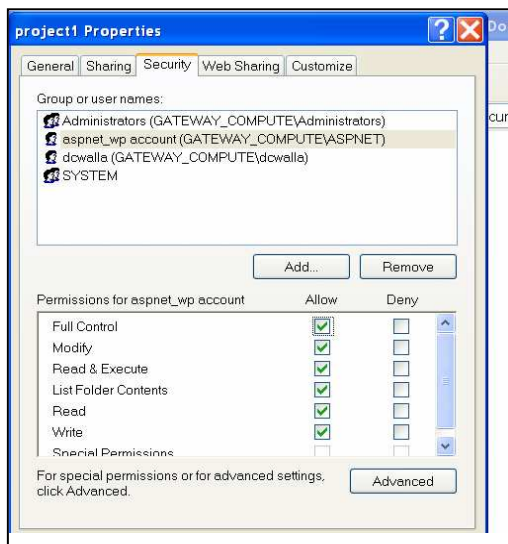


Figure 5. Security Permissions

It is important to note that setting the permissions for full control and access increases the risk of unwanted access if the student's computer is attached to the

Internet. While the students were taking the class, the instructor recommended against connecting to the Internet with these settings. Most students choose to solve this problem by dedicating their laptops for the course and reversing the settings after the semester. This avoided any problems with security on the Internet. For students that could not do this, the network administrator for the department setup four computers that had IIS and Visual Studio.Net installed but were not connected to the Internet. According to the students, this arrangement was successful.

2.3. IIS and Visual Studio.NET Setup

Under the Control Panel of the start menu on Windows XP desktop, there should be an administrative tools folder or icon. If the student selects this folder, they should be able to locate the IIS tool. By selecting the IIS tool, the student can create an application on their web server to test each of their projects during the semester. Figure 5 illustrates the web server folders in a tree structure. The Default Web Site is listed within this tree structure.

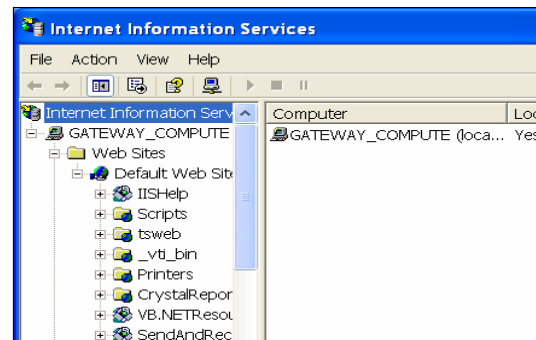


Figure 6. IIS Administrative

Now, the student can create the project application from this panel by right clicking the Default Web Site node and choosing "New" from the drop-down menu. New refers to the Term Virtual Directory. By setting the appropriate permissions, the student can create an application on the web server for their project. Once students select this option, the Virtual Directory Creation Wizard will guide the student through the process of creating the application. Figure 7 demonstrates the initialization of the wizard.



Figure 7. IIS Virtual Directory

At this point, the wizard will prompt the student for the name of the project and the physical directory where the project will reside. In our case, the instructor provided the students with this information. Finally, students will need to set the access permissions for their application on the screen illustrated by Figure 8. Notice that Read, Run Scripts, and Browse are selected.

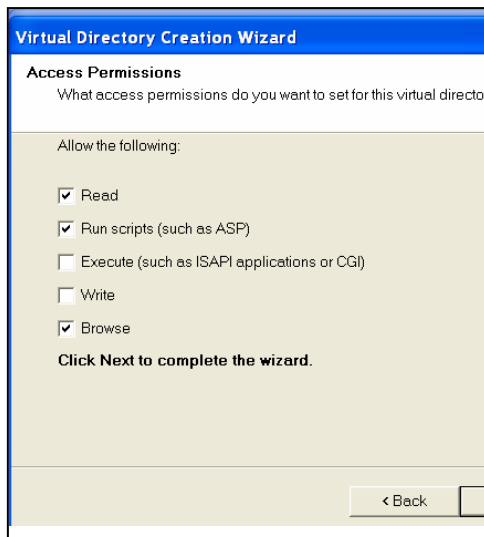


Figure 8. Access Permission

Now that the Web Server is setup to handle the project application, students will need to setup Visual Studio.Net to access this application. Within Visual Studio.Net, students can set access to their applications by selecting Tool from the menu and

choosing Options from the submenu. Figure 9 reveals the various parameters that affect the performance and appearance of Visual Studio. Under Project and Solutions, the student can identify the physical folder location for the project. This folder should be the same folder identified in the Virtual Directory Creation Wizard. It is important that Visual Studio point to the correct application folder. The instructor must emphasize the importance of this connection. Every project created during the course will need to establish this connection correctly. This allows Visual Studio to simulate both sides of the client server application by performing as both the client and the server during the execution of the web form and the execution of the code behind the file.

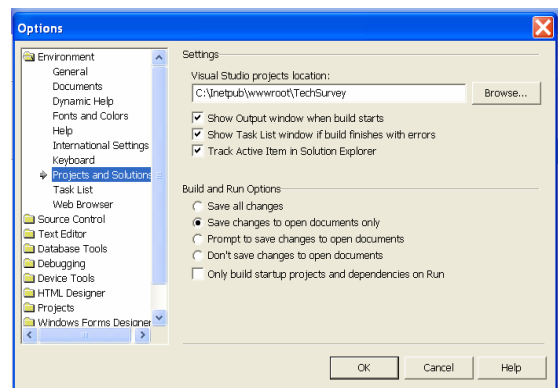


Figure 9. Visual Studio

3. Course Content

The client-server assignments in the course require students' computers to simultaneously perform as both a client computer and a web server. This dual role demands that students work with software that can be used to both develop the client-side applications and execute the server-side operations. Figure 10 illustrates this relationship.

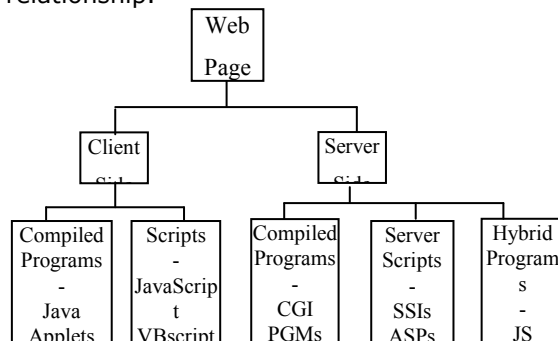


Figure 10. Client/Server-Side

Visual Studio.NET software is uniquely suited for the demands of this course and was used as the integrated development environment (IDE) to develop, test, and debug the ASP.NET applications. Figure 2 illustrates the integration of all three components necessary as the foundation for the ASP.NET applications. The core content of this topic course is illustrated by Table 1. The client side web component was divided into two areas of focus: (1) Introduction to HTML/Form Processing, and (2) Client-side Scripting using Java Script. The focus of the latter included instruction on the logic and syntax of the scripting language, data validation, cookies, and debugging techniques. The textbook used for this course was Morrison and Morrison's (2003), *Database Driven Websites* which provided detailed explanations for the setup and applications for the .NET assignments and examples.

ASP.Net Web Development		
I. HTML	I. Web Server	I. ASP.Net
A. Tags	A. IIS	A. VB.Net
B. Tables	B. Physical	B. Language Structure
C. Hyper-Links	C. Virtual	C. Server Controls
D. Forms	D. Applications	D. WebForm Cycle
II. Client-side	II. Visual Studio	E. Data
A. JAVA-Scripts	A. Options Pgm/Sol	Validation
B. Methods	B. Creating Projects	II. DB Process
C. Function	C. Develop Process	A. Connect
D. Language Structure	D. Test/Debug	B. Adaptor
E. Data		C. Display
Validation		D. Methods
F. Cookies		E. Properties

Table 1. .Net Core Content

The middle section of the course focused on the environment encompassing: 1) Operating System - Windows XP Professional; 2) Web Server - IIS; and 3) Visual Studio. It was imperative that the student understood how these three components worked together to generate,

test, and debug ASP.NET applications. One of the most daunting aspects of the course was properly identifying the causes of confusing error messages so that students could correctly identify the location of their coding errors. For example, if the error generated a message that denied the application update access to a database, the student would need to determine if the cause was the operating system file privilege on the directory holding the database or if the error was caused the privilege in the database structure itself. Understanding the role of the operating system, the files within the server directory, and the privileges assigned to each gives students the ability to systematically locate and correct the error.

The last section of the course content involves more advanced ASP.NET application building. It centered on the implementation of a language (in our case, VB.NET), the server controls that contains the various server controls, the web form processing cycle, the structure of the code behind file relative to the cycle, and finally the processing of a web server database using a .NET application.

The setup and control of projects for the development ASP.NET applications is an integrated process of balancing the operating system, the Microsoft's IIS web server, and Visual Studio.NET. If one of these components is not set correctly, the student will often receive server errors that are difficult to diagnosis. Approximately, one half of the execution errors that occur during testing the web forms within Visual Studio were the result of a broken link in this process. After a period of time, students were able to trace the link path from the operating system to the IIS web server to Visual Studio.NET. And once students discovered where the broken link occurred, they were generally able to correct it.

Having students work with the nuts and bolts of web forms and visualizing the pseudo-conversation between the web server and the client helped them to understand the logical construction of an integrated web application.

3.1. Assignments

The assignments during the semester were design to expose the students to the

different levels of client-side and server-side processing required to develop ASP.NET applications. The first two assignments focused on the basic development of HTML pages using tags for navigation, style, and content. The second assignment involved basic form processing and utilized data input components. For this assignment, students were required to use client-side scripting (JavaScript) to validate user input. Creating user defined procedures in the heading section with data validating and error reporting was an essential requirement for this assignment. The last four assignments focused on the development of web forms using code-behind files to illustrate the post-back logic of ASP.NET. Each assignment evolved a little deeper into the logic cycle of the ASP.NET using Visual Basic as the language of implementation.

The use of server controls and rich server controls which resembled the basic form components of an HTML web page with a wider degree of functionality was the main concept that the students used to integrate both client and sever sides into a practical application. The students were also required to use the validation controls embedded within ASP.NET to scrub their data. Often, special customizing logic was required to adapt to a practical application. The final assignment focused on the use of a relational database to retrieve data and display it on a web form. The focal point of this assignment was the four step process of connecting to the database, generating a data adapter (using the SQL Select statement to retrieve data), creating/filling a Data Set, and using a DataGrid control to display the data logically on a web form. The overall objective of this assignment was to utilize the major components of ASP.NET to produce a "real world" web site.

4. SURVEY

One of the objectives of this course was to provide students with the real world skills they would need to compete in the job market. However, another, equally important objective was to deliver a course in a way that the students would feel was enjoyable and worthwhile. To assess this second objective, upon completion of the course, we surveyed the students to measure their reaction to: (1) the course content, (2) the tools (hardware/ software),

(3) the instructor, (4) the job skills taught, and (5) the interaction of these components. Twelve questions were listed on the evaluation form rating each of these areas on a scale of 1 to 5 with a rating of 1 as the highest positive score and 5 as the lowest rating. This scale was chosen because the students were very familiar with this format from previous course evaluations within the university. Figure 11 illustrates the relationship between the questions and the 5 areas of measurement (see Appendix A for the evaluation form).

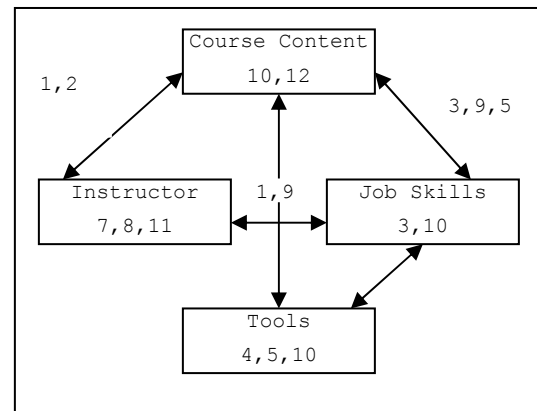


Figure 11. Questions/Measurement

As illustrated by Figure 11, each question was analyzed according to its major impact on one or more of the focus areas as well as the interaction between the areas. For example, Question 1 asks the student to rate the concepts and skills taught in the class. This question was determined to reflect most on the instructor's interaction with the course content and the job skills taught areas. Thus, Question 1 is reflected in the diagram with double arrows indicating the interaction between instructor and course content, and between instructor and job skills. On the other hand, Question 7 asks the student to rate the instructor's knowledge of the course material. This question was determine to mostly impact the instructor's capabilities and was placed in the instructor's box on the diagram. One by one, each question was categorized in this fashion.

4.1. Survey Results

In general, the data suggests that students enjoyed the course; felt the instructor was knowledgeable and competent, and that the class provided them with needed job skills. Figure 12 shows the average scores and standard deviation for each of the focus areas as well as the interaction between each area. Figure 12 shows an overall average score of 1.61 with a standard deviation of 0.71. The instructor focus area scored the highest positive rating with an average of 1.49 with a standard deviation of 0.68 and the instructor's interaction with course content and skills taught were also rated relatively high. The data suggests that the instructor's knowledge of the course material, organization, planning and presentation of the material was an effective component for the class.

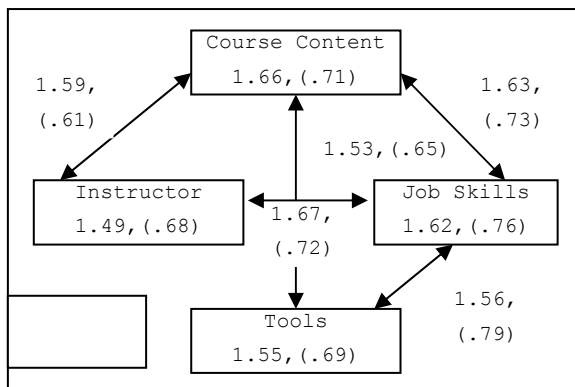


Figure 12. Average/ (SD) Scores

The students gave the tools (hardware/ software) the second highest score with a mean of 1.55 and a standard deviation of 0.69. The data suggest that students felt that the .NET technologies along with the IDE Visual Studio were effective in teaching students the basic foundations of the .Net framework and programming principles. The rating for the interaction between the tools, the job skills, and the course content further supports the idea that the tools effectively complemented the course content and the job skills being taught in the course. In addition, the scores for Job Skills were relatively high with an

average of 1.62 and a standard deviation of 0.76.

The data suggests that the students felt that the course material and presentation of the material helped them to develop the job skills required for the client/server environment on the Internet. Also, the average score and standard deviation for the interaction between course content and tools indicated that both factors significantly helped them develop and refine their computer and logic skills necessary to effectively application on the Internet. Finally, it appears that the student felt confident that they could successfully apply the concepts to an entry-level job position focusing on developing e-commerce applications.

5. CONCLUSION

With the increasing importance of the Internet to communication, commerce, and entertainment fields, academic institutions should incorporate Web based client/server technologies, traditional mainframe technologies, and bridging software that allow for the integration of these two technologies into their curriculum. We believe that computer science curriculums should accommodate both the traditional mainframe as well as the Internet client/server environments to best serve the information technology field and the graduates entering this field. Bridging technology such as Microsoft's .NET framework and the IDE Visual Studio can play an important role in helping students close the gap between client/server and mainframe technology by focusing on the integration of these two components within the spectrum of applications that often encompass each approach for a more dynamic and cost effective use of an organization's computing resources.

Our end of semester survey suggests that the students enjoyed the course and felt that the course material and presentation of the material help them develop the job skills required for the client/server environment on the Internet. In addition to the survey, we feel that the class and the technologies used were a success because students were able to effectively apply the concepts taught in the class to practical applications and projects.

In this paper, we have discussed ways to incorporate ASP.NET into the classroom and given step-by-step instructions for setting up both the client and the server side software for academic use. Throughout the paper, we have attempted to point out potential pitfalls and alert the reader to the most common sources of difficulties our students encountered. In addition, we have outlined our course content and class assignments with the hope that this may be helpful to other academics wishing provide a similar course.

6. REFERENCES

- Balasubramanian, T. (2004). Living with the COBOL Legacy. *The Economic Times*, March 25, 2004, Retrieved January 19, 2005 from www.acucobol.com/company/press/articles/article_61.php
- Durward, K., Hayes, D. H., & Matto, N. (2004). Federating Mainframe Data. *DB2 Magazine*, 9, Retrieved November 8, 2004 from www.db2mag.com/showArticle.jhtml?articleID=17602326
- Krill, P. (2004). IBM Looks to Modernize COBOL. *InfoWorld*, Retrieved May 19, 2005 from www.infoworld.com/article/04/05/11/HNcobol_1.html
- Koma, B. (2003). The Incredible Shrinking Legacy Workforce. *Optimize*, August 2003, issue 22 Retrieved November 8, 2004 from www.optimize.com/article/?printableArticle=true &articleId=17700946
- Martens, China. (2005) Dear IT Graduate, Just One Word: Mainframes, *ComputerWorld*, IDG News Service, Retrieved September 3, 2005 from: www.computerworld.com/hardware/mainframs/story/0,10801,1036
- Murach, M. (2005) The Future of COBOL. The COBOL Center, Retrieved Fri May 20 2005 from <http://www.infogoal.com/cbd/cbdz040.htm>
- Morrison, M. & Morrison, J. (2003). *Database-Driven Websites* (2nd ed.). Boston: Course Technology.
- Rohde, L. (2004), Microsoft Tries to Cozy Up to Mainframe. *ComputerWorld*, IDG News Service, Retrieved December 4, 2004 from: www.computerworld.com/printthis/2004/0,4814,95400,00.html
- Thibodeau, Patrick. (2005) Shortage of Mainframe Skills May Give IT EXECs Gray Hairs, *ComputerWorld*, IDG News Service, Retrieved September 3, 2005 from: www.computerworld.com/hardware/mainframs/story/0,10801,1035
- United States Department of Commerce. (2004) Washington, D.C., Retail E-Commerce Sales.

Appendix A

Course Survey

For items 1-9 respond:

A = Strongly Agree
B = Agree
C = Neutral
D = Disagree
E = Strongly Disagree

For items 10-12 respond:

A = Excellent
B = Good
C = Acceptable
D = Poor
E = Unacceptable

1. Concepts and skills were effectively taught in this course.
2. The class periods contributed to learning the course material.
3. The assignments helped you understand and apply the concepts and skills being taught.
4. The assignments were accessible using my computer and software outside the classroom.
5. I was able to work effectively on assignments using my computer system outside the classroom.
6. The course content and the computer system were integrated effectively for a better learning environment.
7. The instructor had an excellent knowledge of the course material.
8. The use of class time showed effective planning and organization by the instructor.
9. I am confident that I will be successful in applying the concepts taught in this course to an entry-level job position.
10. How would you rate the material presented in this course?
11. How would you rate the instructor overall?
12. How would you rate the course content overall?