The Ruth & Ted Braun Awards
for Writing Excellence
at Saginaw Valley State University

Creating an E-Learning Environment

Dustin Kuhl and Larry Wascher

COLLEGE OF SCIENCE, ENGINEERING & TECHNOLOGY

Nominated by Dr. Farid Hallouche
Assistant Professor of Computer Science

Dustin Kuhl, from Sebewaing, Mich., is in his senior year, majoring in Computer Science and minoring in Computer Information Systems. After graduation he hopes to find a job as a programmer or database administrator and one day own his own business.

Larry Wascher, from Reese, Mich., is currently in his senior year at SVSU, majoring in computer science and minoring in chemistry. He hopes to find a career in network engineering and system administration. Good technical report writing skills have proven to be very useful in his degree fields to communicate information and data to others who may not be as apt to understand.

Introduction

The development of software programs presents unique challenges. This project presents research and evaluates options for choosing a programming language for the design of two types of educational messaging programs. The project had two main objectives:

1. Provide a tool for students and teachers to use in a classroom type setting, mostly to establish better communication between one another, as well as allow students and teachers to interact in different ways outside the classroom.
2. Provide ourselves with hands-on learning for real world applications.

The requirements were to create a simple instant messaging program that would allow people to communicate with one another over a small local network, and to design and create an educational web program allowing students to access a virtual classroom environment.

To begin, we compared a number of different programming languages, for the purpose of finding one suitable to write the network applications with. We also researched different educational messaging programs and virtual classroom environments to better understand how they work.

A local network computer messaging program can be a very useful tool for students in a classroom environment,

allowing them to communicate with each other. Rather than walking across a building or room, making a phone call, or sending an email message to communicate with others, students and teachers will be able to use the instant messaging program on their computer to type and send text messages to others using the same software. An interactive educational classroom program can also serve as a very useful tool in classroom settings. It would allow only valid students to log in and view important class information, as well as communicate with others about similar topics of interest. Students would be able to view virtual calendars, join forums, exchange files, take online quizzes, send emails, and view announcements on a secure virtual classroom environment.

The code for the instant messaging program was written using the Visual Basic 6 programming language, and the code for the virtual classroom environment was written using ASP.NET, XML, SQL Server, and VB.NET, both running on Microsoft Windows NT platforms. While both the messaging program and virtual classroom use TCP/IP to link to other computers, the virtual classroom program uses HTTP channels to allow users to connect over the internet. When the messaging server is initiated, it will establish a listening socket over the network, listening for any connection requests from clients. When client computers send out a connection request to the server, the server initiates the services between the two. Once the service is initiated, users will then be able to send simple text messages back and forth. When the educational web program is installed and running on the server, users will be able to connect to it from anywhere using the Internet. When the user initially connects, he/she will be authenticated before being allowed access to the class's private data. Once the user is logged in, he/she will have access to a number of different pages available for the different classes.

Research Methods and Information

We performed the following research to determine the programming languages, operating system, design, and compilation for a computer messaging system and virtual classroom environment:

1. To better understand how client/server applications, messaging programs, and virtual classroom services operated and were developed, we used information obtained from classroom lectures, read through chapters of different programming books, and used the Internet to find and review material.

2. To better understand the architecture of messaging and E-learning programs, we performed research on some of the current instant messaging and chat programs available, as well as reviewed some of the educational software we currently use, such as the Blackboard program.

3. We compared several programming languages: C, C++, Visual Basic, VB.NET, C#, Java, PHP, MS SQL, ASP, ASP.NET, HTML, and XML to help choose a language to write both the messaging and E-Learning programs. Comparisons were made mostly by looking at sample code obtained from the Microsoft Developer Network Library (MSDN), textbook examples, Internet examples, classroom textbooks, and lectures.

4. We used previous Visual Basic classroom textbooks and CS401 networking books for the foundation to build the messaging program during the fall 2003 semester. We used the Internet, CS402 classroom lecture materials, textbooks, and other reference books for the E-Learning design project during the winter 2004 semester. These projects combined provide an excellent tool for students and teachers to use in a classroom setting.

5. We experienced some technical problems in advancing the design of our messaging program, and spent some time trying to convert the code to Visual Basic.Net. During that time we started researching ASP.NET for the creation of our educational web programs. ASP.NET, the next generation of Active Server Pages from Microsoft, is designed to support the development of dynamic and data-driven web applications and web services in a more simplified and visual manner.

Some functionalities of ASP.NET include application and web service development, custom controls, data access, security, deployment, error handling, and support for inherited features from other web-related namespaces written in the .NET Framework Class Library. ASP.NET allows users to take an existing server control object, such as the Calendar, DataGrid, or TextBox, and create a new class derived from its base class. Automatically, without having to write a single line of code, inheritance can give a new server control over all the functionality provided in the objects base class. Therefore, to add any new functionality, we only needed to write the code necessary for whatever extra we wanted the control to do, providing a fully strong, usable, Web control similar to the original base control. ASP.NET provides a framework for building fully compiled Web applications. It uses programming languages such as VB.NET, C#, and JScript.Net to increase flexibility and security of its web applications. ASP.NET Web forms have many strengths. Web forms can run on any browser, and provide enhanced functionality for Internet Explorer by using Dynamic HTML. Any .NET compatible language can be used for designing ASP.NET Web applications. ASP.NET code executes inside the Common Language Runtime (CLR), which offers the application a wide selection of class libraries, safety features, and reusable components.

Results

This section presents the findings of our research and design. We cover the need for creating a messaging program and educational web program, initial research done,

programming language selection criteria, development, and design of the programs.

1. TCP/IP Client/Server Program

Both of us were new to the aspects of networking. After comparing several options, we both took interest in the creation of a messaging program as well as an educational web program. After deciding the projects to be created, it took weeks of learning the new network material before we would have a starting point for the two projects. We first learned how the server computer on the network acts as the "central" computer, providing services to all client computers connected to it. The server establishes a listening socket which allows it to listen over the network for incoming connection requests. Connection requests are sent out by the client to a specific IP address where the server resides. The client computer must also use the same port for which the desired application is accessible through the server. Once the connection request is received by the server, the server decides whether or not to accept the request. If the request is accepted, the client and server can interact using the messaging application.

2. Study of Current Messaging and E-Learning Systems Used Today

A number of popular computer messaging and E-Learning systems are used today. Messaging programs allow people to send text messages back and forth over the Internet, while E-Learning systems allow users to log into virtual classroom environments. We first studied how AOL Instant Messenger (AIM) worked. The AIM program requires you to first connect to their website *www.aol.com* and download the messaging software. After downloading and installing the software, you are prompted with a login screen where a username and password are requested. In order to use the messaging software, an active Internet connection must first be established. Screen shots of the main login screen and messaging screen are shown below for both AOL Instant Messenger and the messaging program we developed, which is very similar to the AIM program. (See Figure 1.)

After the information is entered, a messaging screen pops up and allows you to type the username of the person you want to send the message to, and the message to be sent. Both messaging systems use TCP/IP. The main message server is connected to via the Internet. Each person who uses the software to connect to the message server is a client. We created our messaging program to function like the AIM and Yahoo programs, where clients are able to connect to the server computer, and send text messages back and forth to each other. (See Figure 2.)

Figure 1: AOL Instant Messenger Login

CS 401 Messaging Program Login

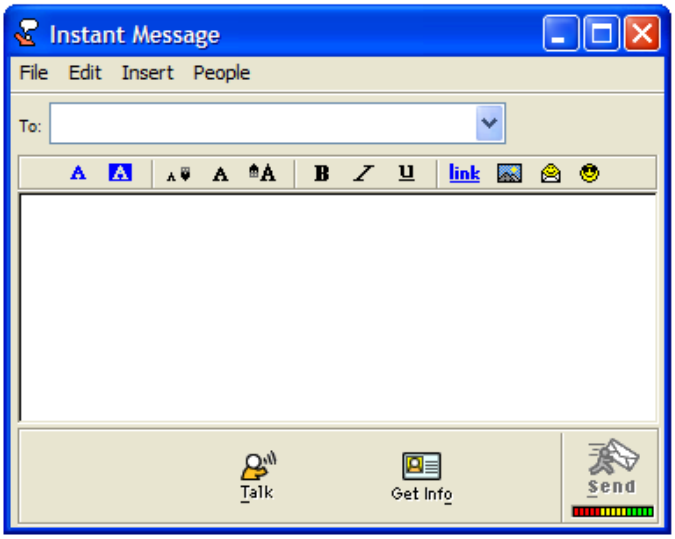
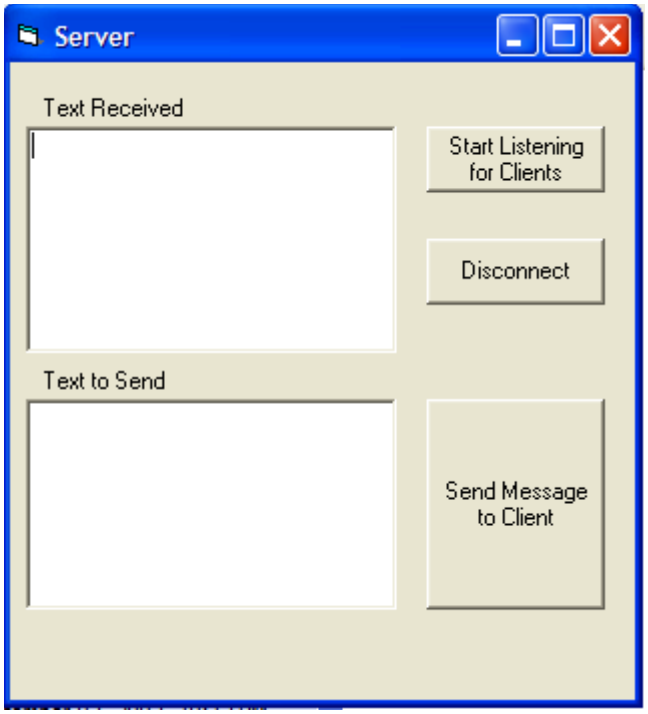
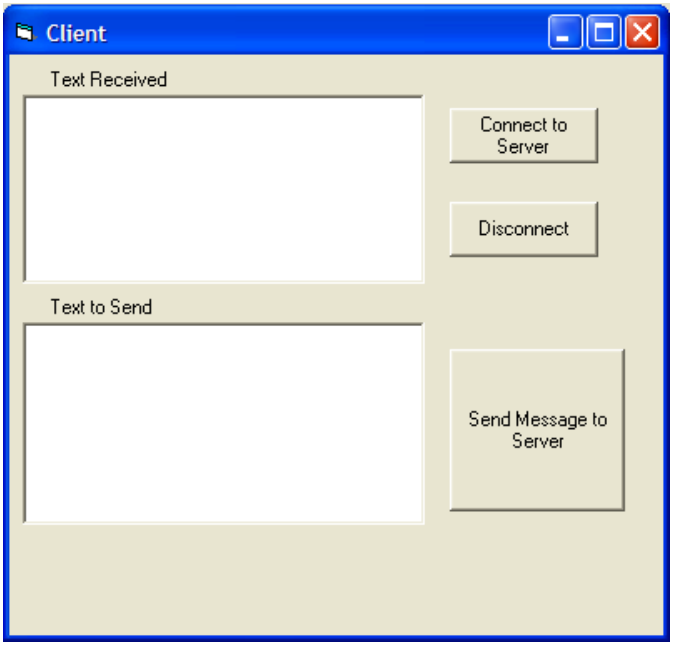


Figure 2: Messaging Program Designed in CS401

Server Program



Client Program



3. Program Language Selection Criteria

Choosing a programming language played a major role in the development of the messaging and E-Learning program. To choose a language to write both the Internet applications in, we compared several programming languages, including C++, Visual Basic 6.0, VB.NET, C#, Java, XML, HTML, MS SQL, and C, by looking at sample code obtained from the Microsoft Developer Network (MSDN) Library, textbook examples, and Internet examples. We found that Visual Basic 6.0 would work best for creating the messaging program since it is an easier language to understand. Visual Basic 6.0 contains a Winsock control which eliminates the need for specific socket programming required by the other programming languages. Visual Basic was also a more user-friendly programming language which made it easier to understand. We decided to write the educational web program in ASP.Net, Visual Basic.NET, HTML, XML, and MS SQL, but primarily focused on learning new aspects of networking as well as increasing our skills and knowledge by getting hands-on involvement with new programming languages.

4. Program Language Functions

Using our previously gained knowledge of Visual Basic, VB.NET, ASP.NET, XML, and HTML, we were able to choose from many different functionalities for both projects. The two tables below contain some of the main methods used for both projects. (See Figure 3 and Figure 4.)

Figure 3: Visual Basic Member Table for CS401 Messaging Program

Function Name	Purpose of Function
RemoteHost	Defines the IP address of the server
RemotePort	Declares the port number on the client machine for the messaging program
Connect	Connects the client to the server
State	Returns the state of the connection
sckConnected	Gives status of the socket
GetData	Waits for incoming data from other computer
SendData	Sends data to other computer
LocalPort	Identifies port on the server machine for messaging program
Listen	Creates a listening socket
sckListening	Identifies socket state used to listen for client requests

Figure 4: Members and Methods Used in CS401 and CS402 Applications

Function Name	Purpose	Language
Dim connectionString As String = "server='(local)'; trusted_connection=true; database='User402' "	Used to make a new local SQL server database connection to 'User402'	VB.NET
SELECT [UInfo].[Name], [UInfo].[Password] FROM [UInfo] WHERE (([UInfo].[Name] = @"& _ "Name) AND ([UInfo].[Password] = @Password))	Used to select which database on the SQL server to access, and validate the user's login id and password	VB.NET
FormsAuthentication. RedirectFromLoginPage(UserName.Text, false)	Used to authenticate user login on the main page, and create a cookie to validate the user for the session	VB.NET
<%@ Page Language="vb" %>	Specifies that Visual Basic code is contained in the page	ASP.NET
<form id="UserControl1Usage" method="post" runat="server">	Specifies user control, and the server it runs over	ASP.NET
Server.MapPath(" ")	Used to map a particular web directory	VB.NET
<?xml version="1.0" encoding="UTF-8" ?>	XML tag, used to specify UTF encoding for the specified XML text	XML
dataAdapter.Fill(data Set)	Used to fill a DataGrid web control with data obtained from the SQL Server	VB.NET
Dim myMailServer As System.Web.Mail.SmtpMail	Used for sending Smtp mail	VB.NET

5. Visual Basic Messaging Program

We first created a server program. The program defines the local port number to be 1544 for the client program connection. The server is then activated by pushing the button labeled "Start Listening for Clients," which creates and establishes a listening socket that listens over the network for incoming connection requests from the clients. The programs send and receive text to a buffer where it is then modified and transferred to the txtReceive text box on the screen. After several times of testing and debugging the server, we found it worked and compiled successfully.

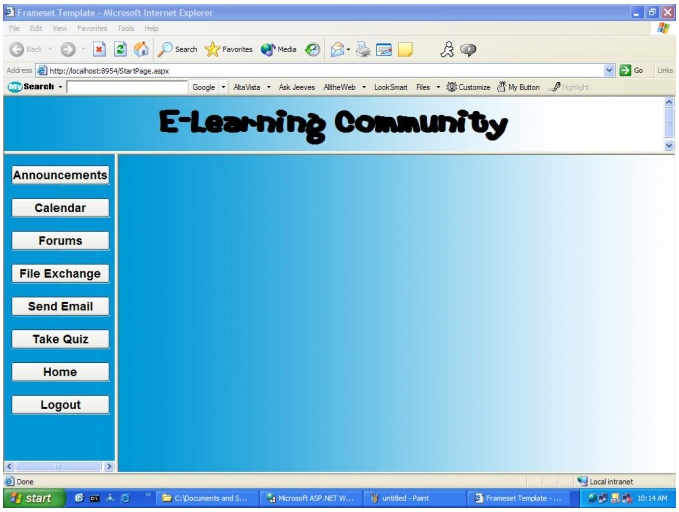
The client program was designed to interact with the server by allowing the users to send text messages back and forth to each other. It first prompts with an input box for the users to type their name. After the users' names are entered, the messaging program appears. When the "Connect to Server" button is pressed, a second input message box appears, requesting the IP address of the server. In order for the programs to work correctly, the server must first be running and listening for clients before the client tries to connect. When the "Connect to Server" button is pressed on the client program, it sends a connection request to the server program through port 1544. The server then has the option whether or not to accept the request.

6. Design of the E-Learning Program

ASP.NET, based on its predecessor ASP, has many advantages over it. First, it uses server side scripting, just as ASP did, but it has been somewhat redesigned, giving it enhanced performance. It supports WYSIWYG editing, as most of the configuration files are text-based, administration is easier, no server reset is needed to replace running/compiled code, and it has better processing management and security. Not all of these will be a factor in our webpage, but it does show how powerful ASP.NET is.

ASP.NET has support for three different programming languages, C#, VB.NET, and JScript. These can be used along with the ASP code to build different kinds of applications for web pages. The homepage to our E-Learning application is shown below. (See Figure 5.)

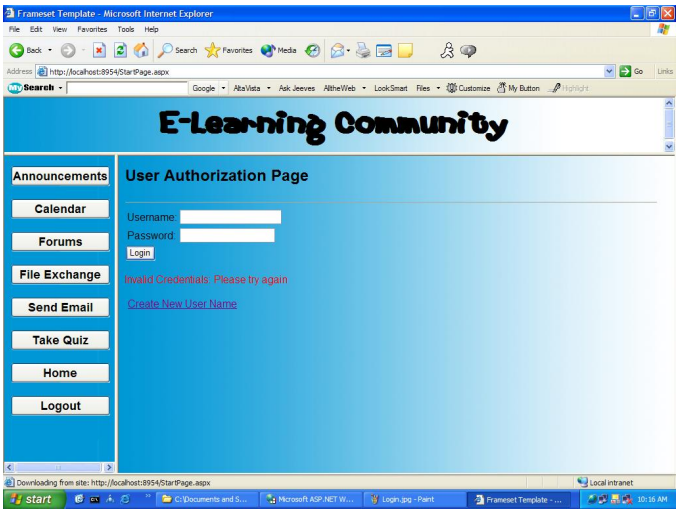
Figure 5: E-Learning Home Page



The forum uses a connection with an SQL server which holds the login information, forum posts, user information, and recent announcements. The website will mainly be used for students and staff to post messages to talk, show information about people registered, get assignments, and check grades. From the main forums page, students and teachers will be able to directly post comments to the SQL server database by following the "Submit" link.

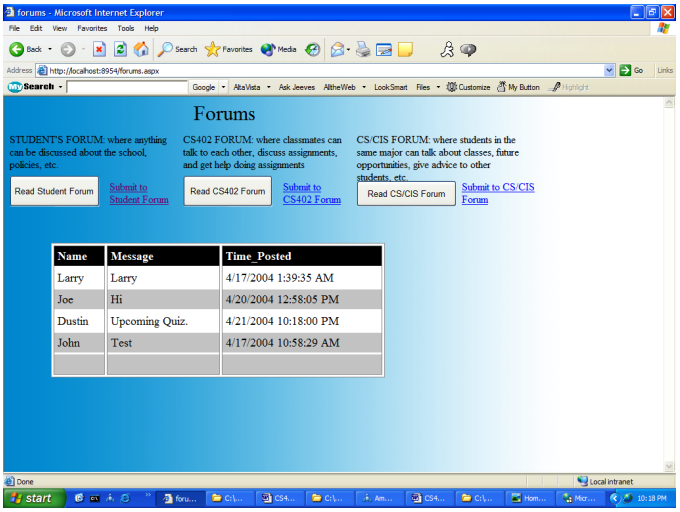
Before creating the main homepage, we created a login page, which would allow only authenticated users to access the site. Once the users type their username and password, the text strings are encrypted and then authenticated with the names and passwords stored on the SQL Server database titled "UInfo." In the event that a login fails, a link appears to a registration page, where the user may register a new user name and password to gain access to this site. To allow such security handling, changes were made to the web.config file, to only allow certain users access to certain pages. Below is a screen shot of the main login page after a login has failed. (See Figure 6.)

Figure 6: User Authentication Page



The main goal of this project was to design a forum where people can talk to each other about different topics, and a place where they can exchange ideas and/or thoughts. Figure 7 is a view of what the Forums Page looks like.

Figure 7: Student Forums Page

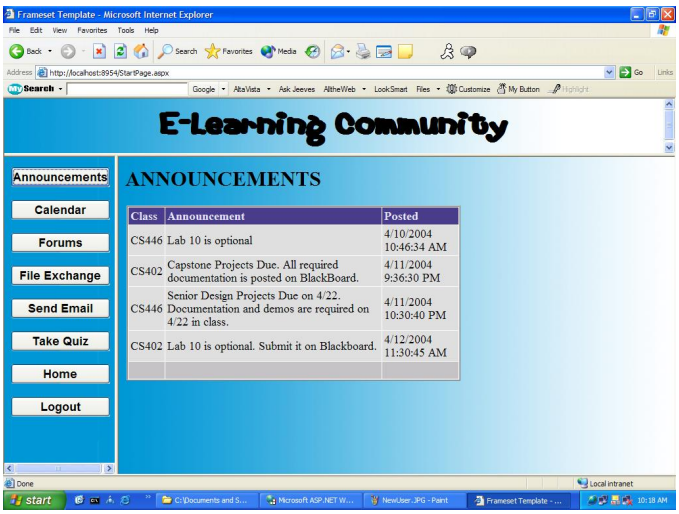


The following forums are used:

- Students forum: anything can be discussed about the school, policies, etc., a free range of topic discussions.
- CS402 forums: classmates can talk to each other, discuss assignments, and get help doing assignments.
- CS/CIS forums: students in the same majors can talk about classes, future opportunities, jobs, and give advice to other students on class scheduling, etc.

We included an Announcements Page, somewhat similar to the Forums Page, but allowing only the teacher or database administrator access to add or delete announcements. In the Announcements Page, students can access news on upcoming quizzes, tests, homework, and other late-breaking course information. The announcements themselves are called from a database file stored on the SQL server. The Announcements Page is shown below. (See Figure 8.)

Figure 8: Announcements Page



The next useful page we created is a File Exchange page. This page designates a folder to be used as a container for the files that will be available only to students and teachers. Once in the File Exchange page, all files that are available for download are displayed in a DataGrid. In the Data Grid, other information is also stored, such as filename, when it was last written, and the file size. The filename provides a link to download the file directly to the user's computer. On the other hand, if a user has a file to upload to the server, he/she can click on the link to upload a file, and be redirected to the upload page. Once at the upload page, the user can click the "Browse" button to browse his/her computer for the file to be uploaded. Below are pictures of both the File Exchange page as well as the Upload Page. (See Figure 9 and Figure 10.)

Figure 9: File Exchange Page Conclusion

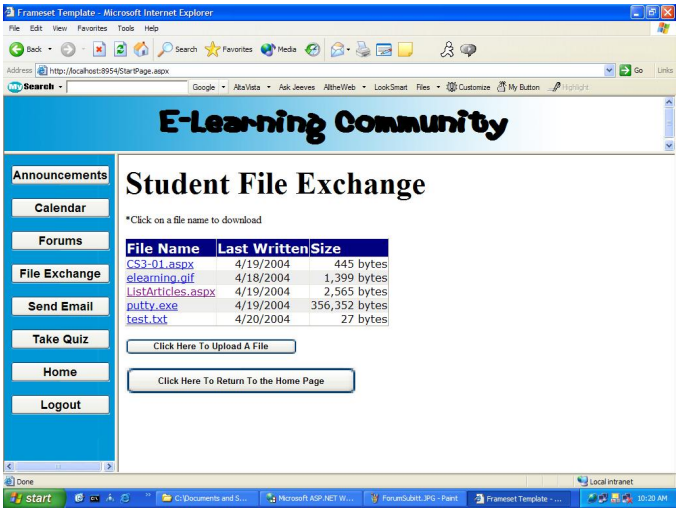
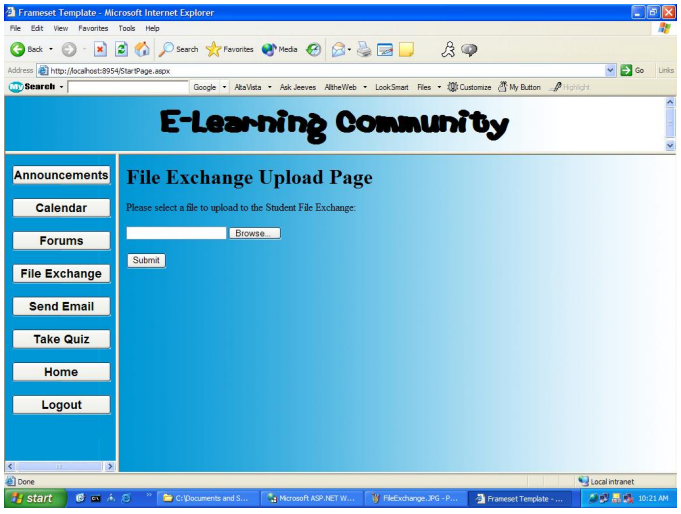


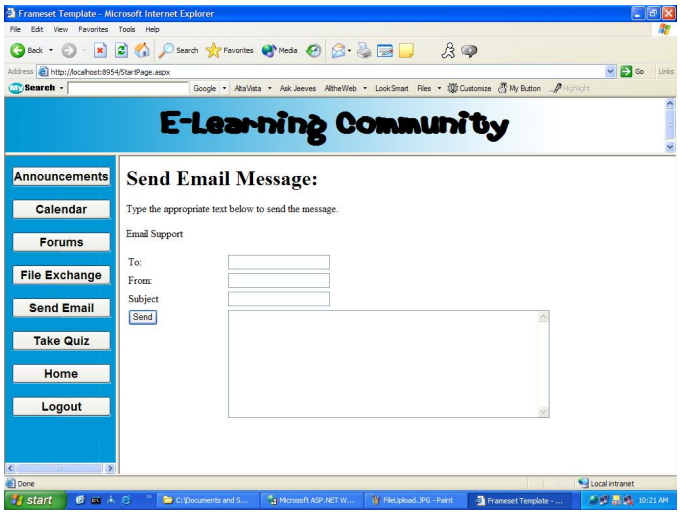
Figure 10: File Upload Page



A quiz page was created and added to the web project to allow students to take practice quizzes posted by the instructor. The instructor can add as many questions as he/she likes, and can also make it possible to record each student's score with a small amount of editing the code. The quiz questions are currently stored in an XML file, which serves basically as a data file for the quiz. Once the students have completed the quiz questions, a results page is displayed, showing all the questions which were asked, and providing all the correct responses.

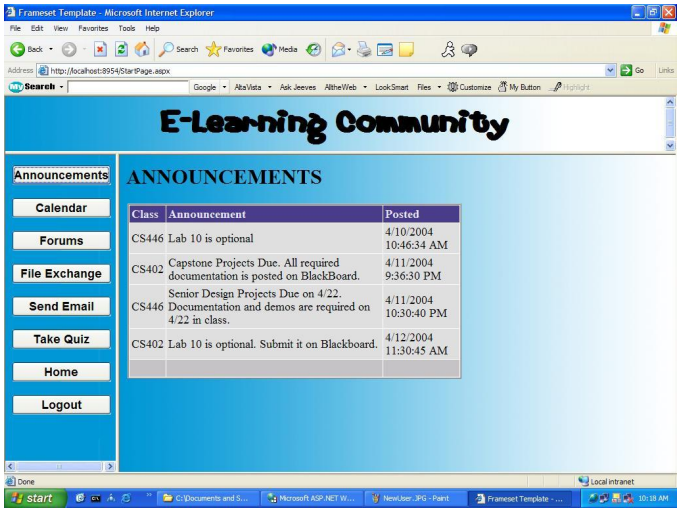
An email page was created to allow users to send instant SMTP email messages from the web site, rather than have to log off and log in to their email. This feature makes it very convenient for students to email other students or teachers with information relating to their classes. Below is a picture of the email page. (See Figure 11.)

Figure 11: Student Email Page



The final page we included is a calendar page, to make easy date access available. When students are checking their announcements, they may need to check due dates of projects, or what day quizzes are scheduled. Once the user is directed to the calendar page, a full size calendar is displayed, in which the user can page back and forth through the months and the years to display the dates he/she is looking for. Below is a picture of the calendar page. (See Figure 12.)

Figure 12: Calendar Page



The messaging and E-Learning system programs will be useful for students and faculty to effectively communicate with each other from the convenience of their own homes. The program will address some of the inconveniences students face, such as interrupting class time to ask questions, and support communicating with other students, thus allowing them to access a virtual classroom. The messaging program works over a local network, whereas the E-Learning program is accessible from anywhere once it has been established and run on a secure web server. Students will be able to post messages in forums as well as check for grades, assignments, announcements, lecture materials, dates, and file downloads. Overall, the messaging program and web program have been a great opportunity for learning the concepts and applications of networking.

Bibliography

Ahmad, S. (n.d.). Using the Winsock Control in Client/Server Applications. Retrieved November 8, 2003, from http://www.15seconds.com/issue/010820.htm

Crouch, M. (2002). *ASP.NET and VB.NET Web Programming*. New York: Addison-Wesley.

Microsoft. (2003). The ASP.NET Web Matrix Project (Reloaded!). Retrieved April 8, 2004, from http://www.asp.net/webmatrix/default.aspx?tabIndex=4&tabId=46

Schneider, D. (1999). *An Introduction to Programming Using Visual Basic 6.0.* New Jersey: Prentice Hall.

Snader, J. (2000). *Effective TCP/IP Programming.* New York: Addison-Wesley.

Sybex Inc. (1999). *Visual Basic 6 Complete.* California: Sybex Inc.

Tutorials and Examples. (2004). ASP.NET Tutorials, Components, Links. Retrieved March 25, 2004 from http://www.123aspx.com/directory.aspx?dir=69