

Online Event Scheduler

By

Ken Enginger

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

University of Cincinnati
College of Applied Science

March 2002

Online Event Scheduler


by

Ken Enginger

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

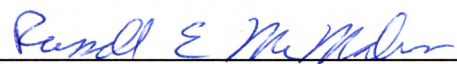
© Copyright 2002 Kenneth Enginger

This program is owned by *ec link*. The author grants to the Information Engineering Technology Program permission to reproduce and distribute copies of this document in whole or in part. The author does not give permission to reproduce and distribute copies of the source code in whole or in part.




Kenneth G Enginger Learner

3-7-2002
Date



Professor Russ McMahon Advisor

3/6/2002
Date



Professor Lawrence Gilligan Department Head

3-6-02
Date

Acknowledgements

There is a co-op company and a few people that I would like to give my gratitude for what they did to make this project possible. I like to thank *electronic commerce link* (*ec link*) for allowing me the opportunity to work in a professional setting giving me the chance to gain the knowledge that was needed to complete this project, and the time to complete this project for the company. I would also like to thank my co-workers at *ec link* John Stullenberger, Robert Brown, Tony Smikle and Thal LaCharity for giving me the guidance and help to complete this project. Without *ec link* and my co-workers at *ec link* this project would have never been completed and I would not know where I would be at this time.

Table of Contents

Section	Page
Acknowledgements	i
Table of Contents	ii
List of Illustrations	iii
Abstract	iv
1. Statement of the Problem	1
1.1 Definition of the Need	
1.2 Other Valid Needs	
1.3 Similar Programs	2
2. Review of the Literature	3
3. Description of Solution	4
3.1 User Profile	
3.2 Design Protocols	
3.2.1 Technical Aspects	
3.2.2 Description	
3.2.3 Invitee's Side	5
3.2.4 Administration Side	
3.2.5 Super Administration	6
4. Deliverables	7
5. Design and Development	8
5.1 Timeline	
5.2 Budget	9
6. Proof of Design	11
6.1 Database	
6.2 Color Schema	14
6.3 Administration Side	15
6.4 User Side	34
6.5 Features	40
7. Conclusions and Recommendations	44
8. References	46

List of Illustrations

Figure 1 Time Line Flow Chart	8
Figure 2 Budget Flow Chart	10
Figure 3 Microsoft SQL Database	13
Figure 4 Administration Flow Chart	16
Figure 5 Administration Login Page	17
Figure 6 Super Administrator Main Page	18
Figure 7 Normal Administrator Main Page	19
Figure 8 Add Event Set Page	20
Figure 9 View Event Set Page (Without Information)	21
Figure 10 View Event Set Page (With Information)	22
Figure 11 Email Event Sets	23
Figure 12 Add Event Step 1	26
Figure 13 Add Event Step 2	27
Figure 14 Required Field Check	28
Figure 15 Add Event Step 3	28
Figure 16 Add Event Step 4	29
Figure 17 Add Event Step 5	30
Figure 18 View Event (No Users)	31
Figure 19 View Event (With Users)	32
Figure 20 Invite User (Current User)	33
Figure 21 Invite User (New User)	34
Figure 22 User Flow Chart	35
Figure 23 Event Set Scheduler	36
Figure 24 Event Scheduler	37
Figure 25 Event Detail Description	38
Figure 26 Event Registration User Information	39
Figure 27 Event Confirmation	40

Abstract

Time and money are valuable assets that cannot be controlled easily in a busy work environment without software. Any program or item that would save time and money while making a job easier and cheaper is encouraged. The Internet helps everyone save time and money in connecting to parties without picking up a telephone. This is where my program "Online Event Scheduler" becomes a valuable asset to a company that holds many events that require conformation.

It is very time consuming to confirm reservations for events that include conferences, meetings, classes, parties, demonstrations, and many other events. This is because it can be hard to get in touch with someone, as well as find time to call him or her. If the call is long distance, then calling a few times to get a response can be expensive. For the person holding the event receiving the call might be time consuming and difficult keeping track of the numerous people invited. I have created a program that needs little administration to invite clientele, confirm reservations, keeping data about the invitees and managing emails to the invites.

The programming that was required to complete this project included: ASP, HTML, JavaScript, and VB Script. Along with the programming a SQL database was using to hold the data about the events and users. With this technology the event owner can relax when they know that the "Online Event Scheduler" is taking care of managing the reservations of their event.

1 Statement of the Problem

1.1 Definition of the Need

One of *ec link*'s partners, Jerry Felix, head of development and my reference, asked me if there was a way that I could help make this process easier with an online event scheduler. We outlined exactly what was needed and how to go about it. He gave me a lot of freedom in developing this project. This program will be used to create and edit an event set; within this event set the creation and editing of different events that invitees can sign up for will be done. The administration side will need to be able to send out information to these attendees if needed. Also, there needs to be a super administration side that can see all objects from all companies that are using this program. Jerry plans to use this when he conducts online conferences, conducts interviews, confirming internal meetings, for registration for online classes that we will provide and any other activity that would include scheduling.

This request from Jerry looks reasonable and the ideas are current, but the difficult part will be putting them together along with a complex database. The platform will be Web based using HTML and VB script code along with Microsoft SQL Sever v7.0 as the database. There will need to be some aspects of JavaScript also. The project will be large and therefore time consuming. However, within the time allowed it can be done. As a whole the program is reasonable.

1.2 Other Valid Needs

Ec link has a proposal on the table to Procter & Gamble for a registration system for surveys. If this is approved, the scheduler would be a step in the right direction for

this project. This is not important right now, but if it is approved, with a change in the look, the scheduler could be used with this project.

Another potential use is as an interview scheduler and event scheduler for any of our current customers or new customers. We want to sell it as a component to other companies for their human resources department for hiring. In addition we have a contract with Wright-Patterson Credit Union for their Web site. Jerry mentioned this project to them and they were interested in it. Once it is finished, our plan is to sell it to them to fulfill a need for Wright-Patterson Credit Union. *Ec link* is good at reusing programs to fill other needs.

1.3 Similar Programs

At the University of Cincinnati we use a program that is similar to this, but not as complex; this is the registration site. One logs in with their student number and PIN then registers for classes (events). My scheduler program will be easier to use because events will be listed for them to sign up for.

Online schedulers are common on the Internet. If one does a search for "online schedulers" one will find many links. A few examples are:

<http://www.usnews.com/usnews/edu/college/schedulr/schedulr.htm>,

<http://www.oe-pages.com/BIZ/Getrich/dcarey/appt.htm>,

http://www.webstation.com.au/online_appointmentdiaries.htm, and many more.

Obviously there is a need for this program because they are out there; at *ec link* Jerry states that there is a need for an online scheduler at *ec link* as well. When we need a Web program if it is reasonable we will build it ourselves; this is usually more cost effective.

In this situation Jerry assigned me to design and build the Event Scheduler.

2. Review of Literature

My references included two of the three partners at *ec link*: Jerry Felix and Bill Nadler. Jerry sat down with me and went through what he wanted from this program. We drew up sketches and diagram of what this site should look like. He told me what functionality he wanted from the scheduler. I was not told how to get the project done, but he did review it from time to time. If Jerry did not like something he would not hesitate to tell me what to change. Bill reviewed the project a few times and asked me to make a few changes and add some functionality. Like most projects the criteria changes throughout the project, but when the project was finished it was what they both wanted.

3. Description of Solution

3.1 User Profile

The users of this program will be cliental of *ec link*, any company that is interested in purchasing it, the public that is interested in taking an online class from *ec link*, the public for interviews, Wright-Patt Credit Union employees, Wright-Patt Credit Union customers and *ec link* employees. The level of IT experience of the users range for beginner Internet users to experienced IT professionals. This will cause the design to be simple allowing all users to benefit from this program. The reason for creating this program is for internal use, but there are already companies that are interested in buying it including Wright-Patt Credit Union.

3.2 Design Protocols

3.2.1 Technical Aspects

Throughout this program languages such as HTML along with ASP will be linked to a relational Microsoft SQL database using VBScript. It will be accessible from anywhere on the Internet. This program will have links to the database to retrieve, delete, update, or add information. There will be pages that will allow administrators to manipulate the data. JavaScript will use message boxes when the user adds or deletes data for confirmation. These checks will prompt the user with confirmation before any information is deleted or updated; this will prevent any accidental deletion or changes. The program will have character checking to prevent illegal character entry into the database. Along with the code behind the program, it will need to be visually appealing.

3.2.2 Description

The components to this project will be an invitee's side, an administration side and a super administration side (*ec link's* employees). The invitee's side will be simple to make it easy for people to sign up for the event that they desire. Through this site administrators will be able to manage the events, manage the people that are signed up for these events, and manage other administrators. The last part will be the super administration side that allows the super administrators to see all the companies using this program and manage any event if needed. These three components are the framework of this project.

3.2.3 Invitee's Side

On the invitee's side there will be a web address that will allow new or old users to sign up for an event if it is not full to capacity. Before they finish registering they will need to confirm their personal information, or, if they are new, they will need to input their personal information. Once this is done, they will be prompted to go to the main registration page. While they are going back to the main page they will be receiving an e-mail confirming their registration. This e-mail will have the web address that will take them back to this site in the future. This site needs to be simple because they are potential customers. The easier it is, the more likely they will attend the conference and be an *ec link* customer.

3.2.4 Administration Side

On the administration side there will be a login page that will authenticate the user. If authenticated, the user would be taken to a view event set page where they can add an event set, edit an event set, or view a list of events within an event set.

If they choose the **view events** they would go to a page that will allow them to add an event or to view an event. When they choose to **view event**, they will see that event's information. On this page they will be able to click buttons to edit the different parts of an event or to invite people. When they select the **invite** button they will be able to invite people that are currently in the database for that company or invite a number of people to this event. Once the administrator inputs the number of people they want to add, they will be prompted to input the invitees' user information. Once this is done e-mails will be sent out to each of these people; these emails will have a link to the invitee's page.

On the main page there will also be a place to add, edit, and view administrators and current user. On all the pages there will be a link to logout or to go to the main page, which will be the event set view page.

3.2.5 Super Administration

The super administration will be just like the administration but there will be a prompt to specify what company the data will be added to. It will also show all the event sets, events, administrations, and users from all the companies. This will allow administrators at *ec link* to change anything and to oversee the different companies' data.

4. Deliverables

The deliverables that are included in this project consist of 8 main priorities that have to be met to consider it complete to sell in the Internet market.

1. The first is the ability to e-mail users one at a time, or a whole group that is registered for an event or will be invited to register for an event. This will be done through ASP and *ec link's* e-mail component.
2. Second is the ability to change the look of the page through style sheets for different companies and event sets.
3. Third is a wizard type interface that is used when an administrator creates an event for an event set. The way this will be done is many pages will divide the input of information and allow the user to back up and change the data as needed.
4. Fourth, throughout the program, deletion checks will be used through JavaScript so wanted information does not get deleted.
5. Fifth, throughout the program illegal data checks will be used through ASP so illegal data is not added to the database.
6. Sixth, throughout the program required field checks will be used through ASP so the empty data will not get added to the database.
7. Seventh, authentication for the users on both the invitees' side and the administration side are needed to prevent unauthorized access.
8. Eighth priority will be on the administration and super administration side, which will be the lists of event sets, events, companies, administrators, and users along with the pertinent information of each of these objects depending on the administrator's permissions.

5. Design and Development
5.1 Timeline

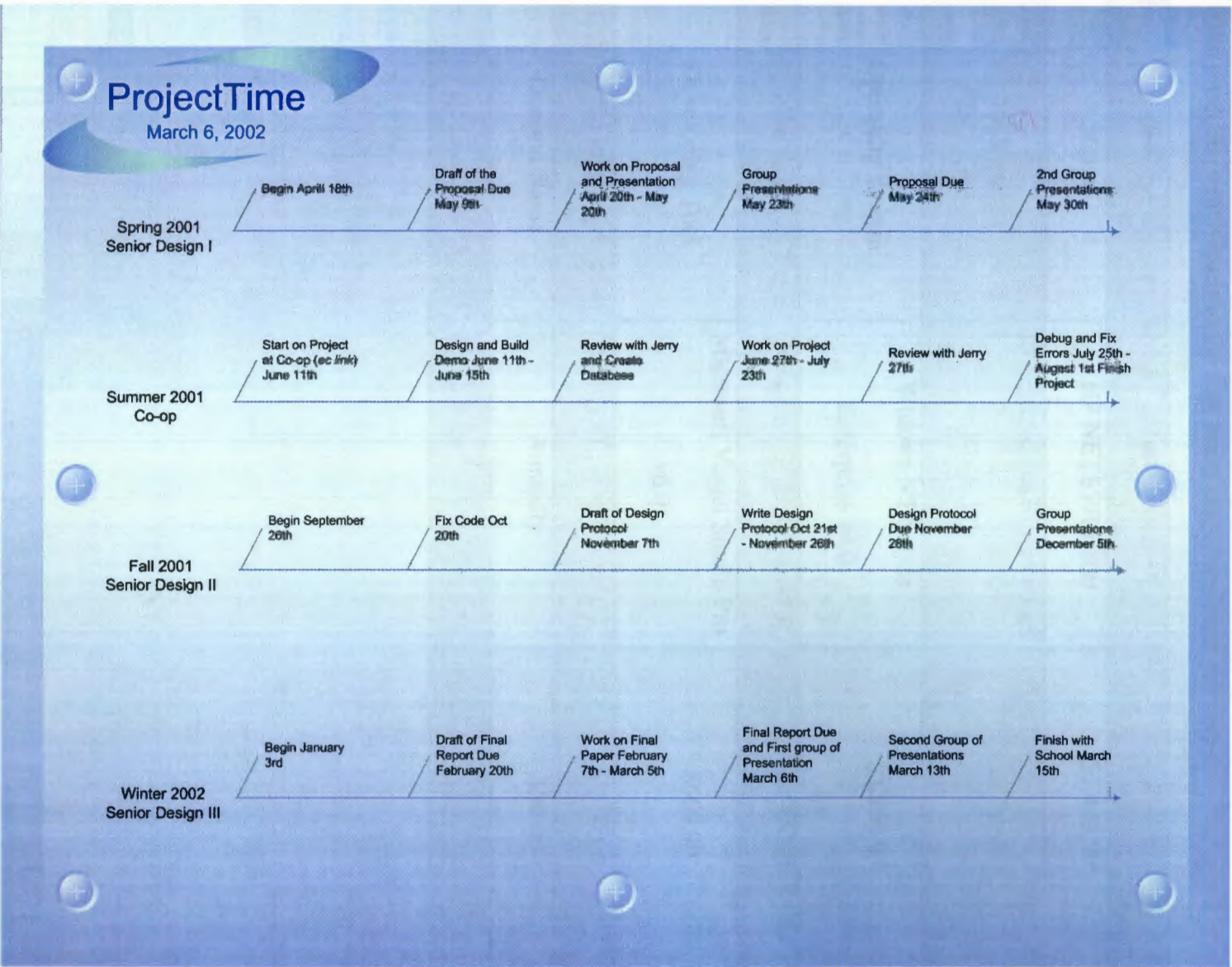
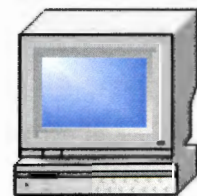


Figure 1 Time Line Flow Chart

5.2 Budget

These prices were found on <http://www.egghead.com>. *Ec link* is equipped with all of these products, which leads to the cost being only man-hours put into it.

Product	Description	Price
Server	HP NETSVR E800	\$1,439.99
Monitor	LG Electronics Studioworks 775N 17"	\$159.99
Operating System	Windows NT Server Enterprise v4.0	\$3,039.99
Database	Microsoft SQL Server v7.0	\$1,699.99
Developing Tool	Microsoft Visual Studio Pro v6.0	\$669.99
High-speed Internet Access	Yearly (T1 From Level 3)	\$1500.00
Domain Name	Annually	\$30.00
	Total	\$8539.95



Monitor
LG Electronics
Studioworks
775N 17" \$159.99



Server
HP NETSVR
E800
\$1,439.99

Domain Name
Annually \$30.00



Workstations

High-speed
Internet Access
Yearly (T1
From Level 3)
\$1500.00

Microsoft
SQL Server v7.0
\$1,699.99

Operating System
Windows NT Server
Enterprise v4.0
\$3,039.99

Developing Tool
Microsoft Visual Studio
Pro v6.0 \$669.99

\$1,439.99	Server	
\$159.99	Monitor	
\$3,039.99	Operating System	
\$1,699.99	Database	
\$669.99	Developing Tool	
\$1500.00	High-speed Internet	
\$30.00	Domain Name	
		Cost Provided by ec link
\$8539.95	Total	

Figure 2 Budget Flow Chart

6. Proof of Design

A project's architecture is a very important step to take into consideration when a project manager plans his or her program. The Five parts of the architecture of this project is client code, server code, color schema, features and Microsoft SQL database. Throughout this project these components have change on weekly bases to fit the plans of the partners of *ec link*. Like many projects during the process of programming the way one gets for point A to point B might change a many times. This is because the programmer may think of a better way to complete his or her task while the developing is taking place and or the management might change his or her needs.

6.1 Database

The database that was chosen for this program was Microsoft SQL because is the size limitation is many times larger that Access and because of the capabilities it has over the Internet. About every page was link to the database in some form. Here is an example of a connection and retrieval of the database:

```
Set dbConn = server.createobject("adodb.connection")
dbConn.Open "DSN=schedule;UID=sa;Password=;"
sqlLoginQuery = "SELECT * FROM admin WHERE
admin_name=" & dbSafe(session("adminname")) & ""
Set dbadmin = dbConn.Execute(sqlLoginQuery)
if dbadmin("admin_super")=1 then
    sqlLoginQuery = "SELECT * FROM users"
    Set dbuser = dbConn.Execute(sqlLoginQuery)
Else
    sqlLoginQuery = "SELECT * FROM users where
    admin_company=" & session("admincompany") & ""
    Set dbuser = dbConn.Execute(sqlLoginQuery)
```

end if

The table and their description are of the following:

1. Admin - this table holds the administrators name, id, company, password and permissions. Linked to the company table through the admin_company to company_id, many-to-one relationship. Linked to the eventsets table through the admin_id to eventset_owner, many-to-one relationship.
2. Company - this table holds the company name, id, abbreviation, if they paid, and their e-mail. It is linked to the style_sheet table through the style_sheet_id to company_style, many-to-one relationship.
3. Style_sheet - the table holds the id, name, company id and event set id
4. EventSets - this table holds the event set name, id, abbreviation, owner, company, style sheet and if it is possible for a user to sign up for more than a one event in this event set. It is linked to the events table through eventset_abv, one-to-many relationship.
5. Events - this table holds the event name, id, abbreviation, event set id, time, owner, type, size, quantity, date, description, time zone, company, and title. It is linked to the eventmbr table through event_id, one-to-many relationship.
6. EventMbr - this table holds the member id, event id and user id. It is linked to the users table through user_id, many-to-one relationship.
7. Users - this table holds the user name, company phone, e-mail, title, and administrator's company. It is linked to the extra_field content table through the content_user_id, one-to-many relationship.
8. Extra_field_content - this table holds the user id and the content. It is linked to the extra_fields table through extra_field_id, one-to-many relationship.

9. Extra_fields - this table holds extra field name, comments, event id and id. It is linked to the events table through event_id, many-to-one relationship.

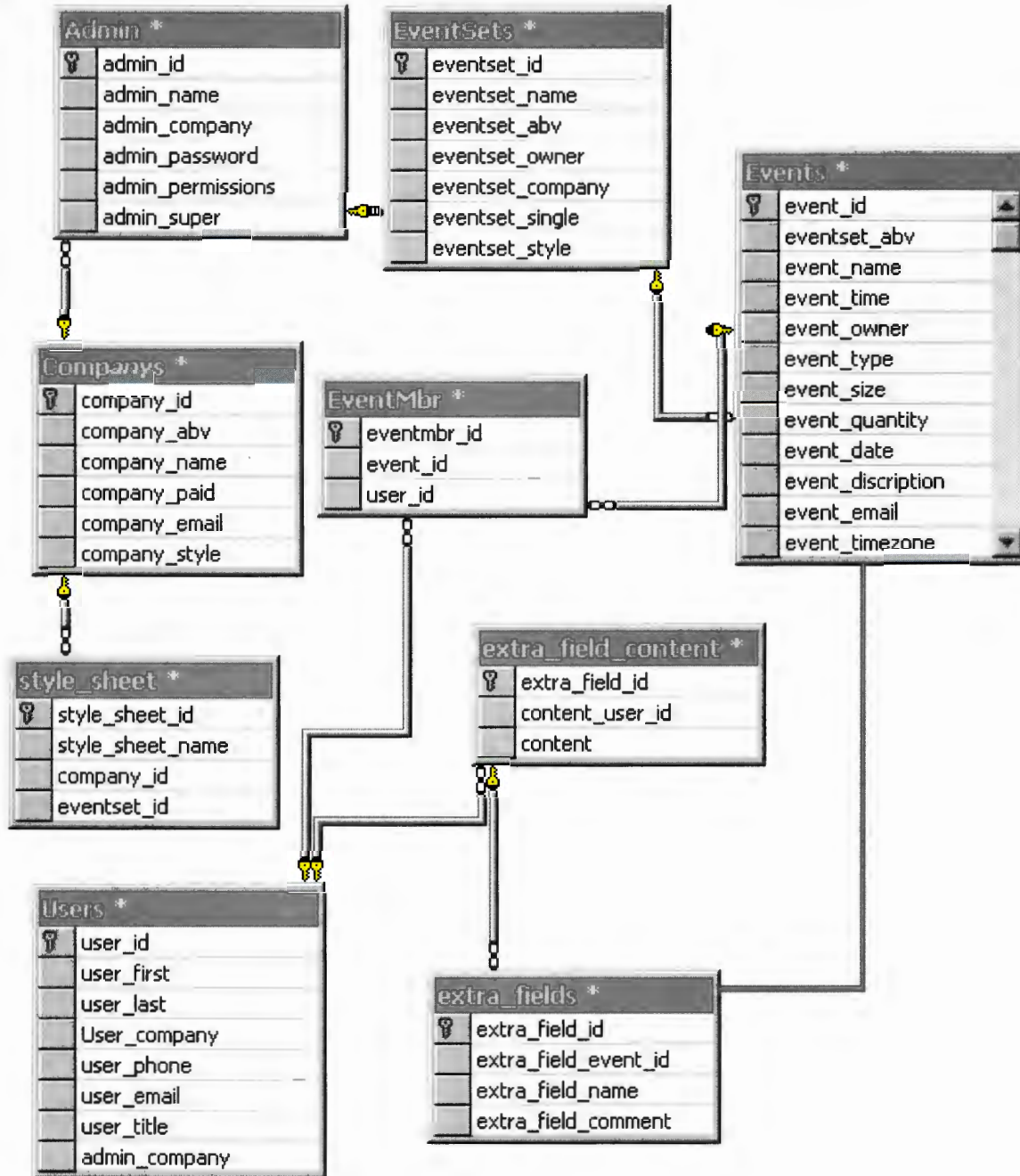


Figure 3 Microsoft SQL Database

6.2 Color Schema

Throughout this program the color schema will be done using style sheets that are chosen by the administrator. When a company is added to the site the super administrator will choose a style sheet that will be default for the administration side for that company. Then when an administrator for that company creates an event set he or she will then choose what style sheet he or she prefer for the invitee's side. These style sheets controls the background color, font color, and font style. Once this has been done, a cookie will be placed on the administrator's computer once he or she has logged in. Here is the coding for the cookie:

```
Response.Cookies ("Stylesheet")("CompStyle")=style
Response.Cookies ("Stylesheet").Expires = Date + 30
if request("style")="" then
    style = Request.Cookies ("Stylesheet")("Compstyle")
else
    style=request("style")
end if
```

The reason this is done is when the administrator goes to the login page the correct style sheet is shown; the login page is the same for all users. Then on the client side the style sheet will be assigned to the active server pages through code depending on what event set that the invitee is registering for. Here is the code for applying the style sheet:

```
sqlLoginQuery = "SELECT * FROM admin WHERE
    admin_name=" & dbsafe(session("adminname")) & ""
Set dbadmin = dbConn.Execute(sqlLoginQuery)
sqlLoginQuery = "SELECT * from companys WHERE
    company_id=" & dbadmin("admin_company") & ""
Set dbcompany = dbConn.Execute(sqlLoginQuery)
```

```

sqlLoginQuery = "SELECT * FROM style_sheet WHERE
                style_sheet_id=" &dbcompany("company_style")&""
Set dbstyle = dbConn.Execute(sqlLoginQuery)
                style=" ../stylesheets/" +dbstyle("style_sheet_name")+ ".css"
<link href=<%=style%> rel="stylesheet">

```

Example: on the *ec link* home page the colors that are used match the web site and the same for the Wight-Patt site the colors match their site as well. These style sheets are link to the ASP through code and the database; the database stores the style sheets name for the event sets and companies.

6.3 Administration Side

Now that the first two pieces of the architecture are in place they need to be tied to the project through ASP and HTML pages. The coding that was used on these pages includes HTML, VB Script and JavaScript. The reason that both VB Script and JavaScript were used is JavaScript is included on the client side while the VB Script is included on the server side. VB Script is used to connect to the Microsoft SQL database and to access the style sheets while JavaScript and VB Script are used in navigation from page to page. JavaScript allow the code to pass values over within the dynamic links and buttons to the next page. VB Script link pages together and passes data from page to page not through the links but through forms.

The next piece of the architecture that is required is the administration side of the program. This side is the backbone to the project. This is where the administrator creates the event set, events, and invites the users. Over 95 percent of the files in this program are located on the administration site. Here is the flow of how this site works:

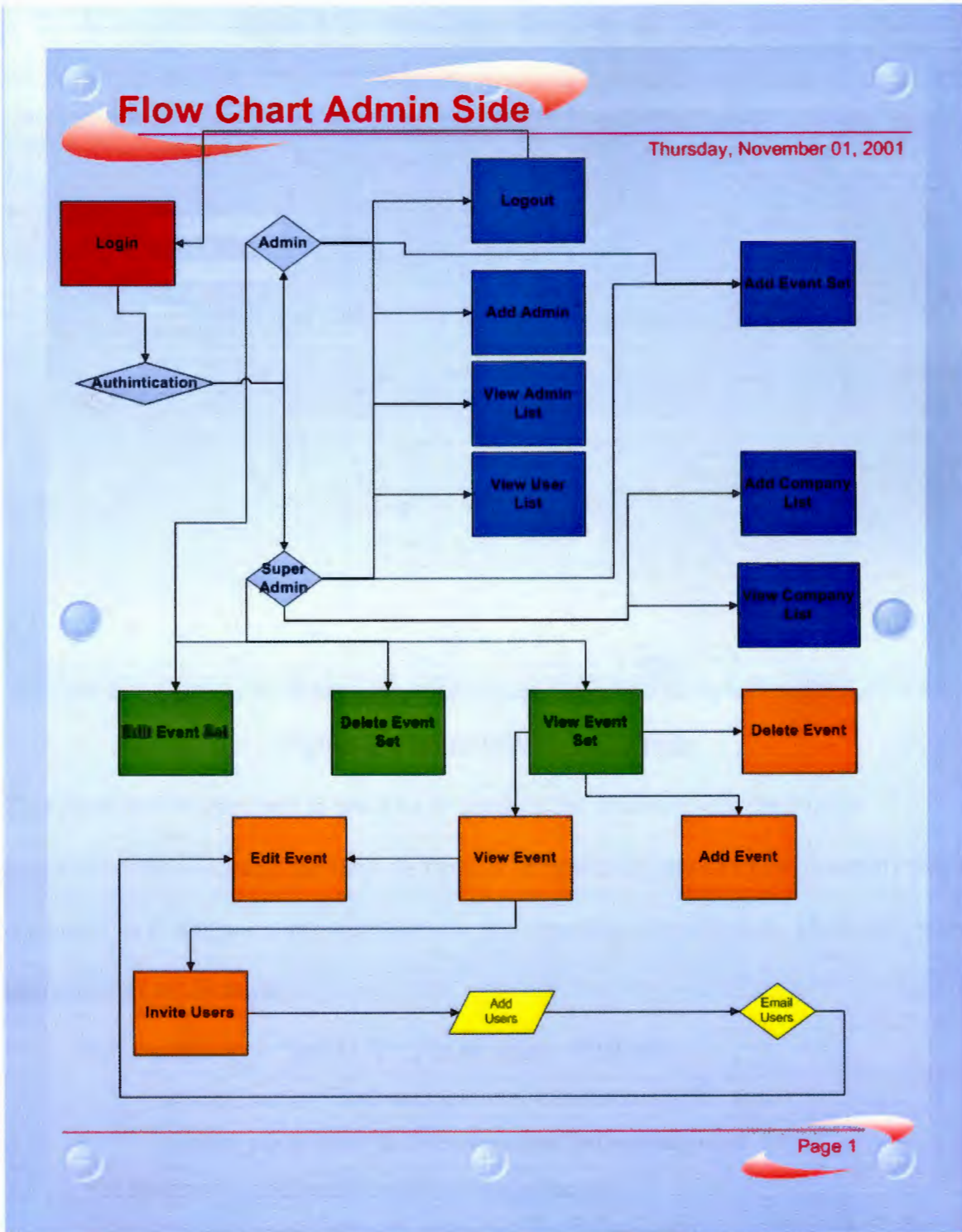


Figure 4 Administration Flow Chart

The first page of the administration and the super administration is the login page.

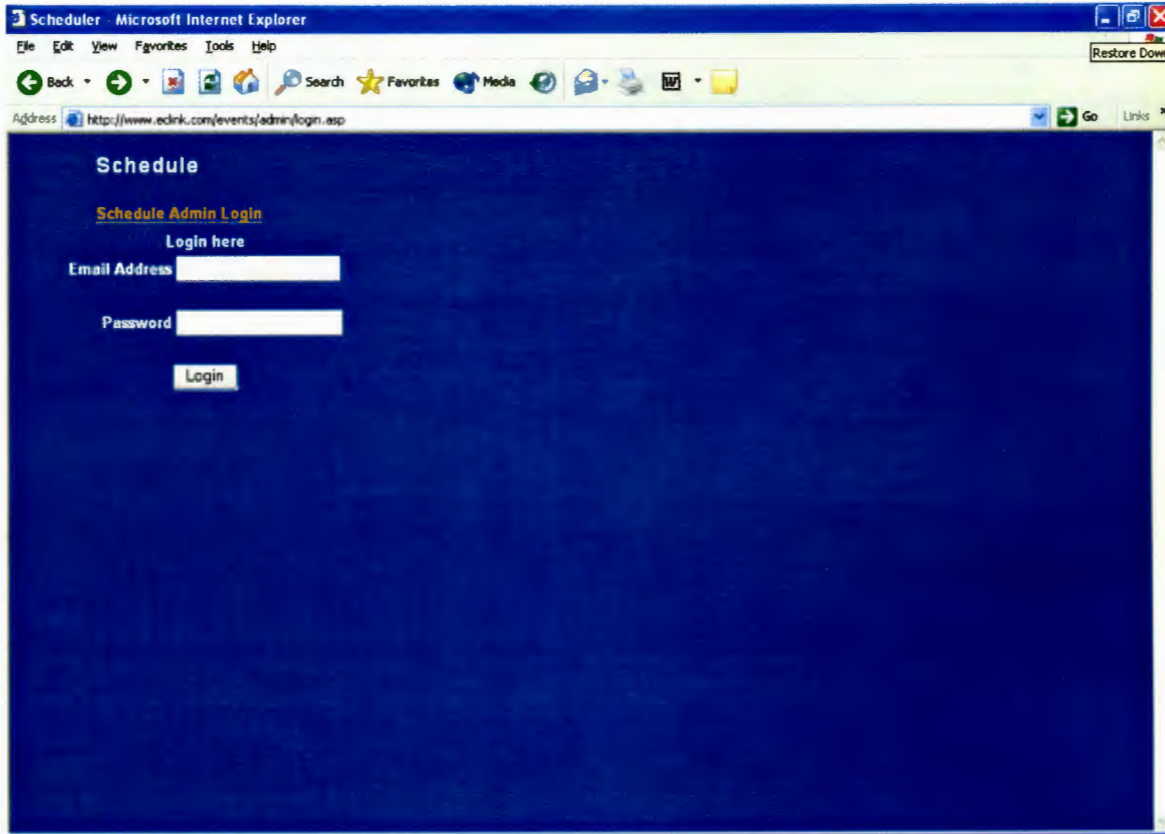


Figure 5 Administration Login Page

This page verifies the users to see if he or she is in the database with the correct password, what company that they are from, if he or she can see all of the company event set, and if he or she is a super administrator or a company administrator. Here is the code that does this verification:

```
sqlLoginQuery = "SELECT * FROM admin WHERE  
    admin_name='"&dbsafe(session("adminname"))&'" and  
    admin_password='"&dbsafe(session("adminpassword"))&'"  
Set dbadmin = dbConn.Execute(sqlLoginQuery)  
session("admincompany")=dbadmin("admin_company")  
If dbadmin("admin_permissions")=0 then  
    sqlLoginQuery = "SELECT * FROM EventSets WHERE  
        Eventset_owner='"&dbsafe(dbadmin("admin_id"))&'"  
    Set dbeventset = dbConn.Execute(sqlLoginQuery)
```

else

if dbadmin("admin_super")=1 then

sqlLoginQuery = "SELECT * FROM EventSets"

Set dbeventset = dbConn.Execute(sqlLoginQuery)

else

sqlLoginQuery = "SELECT * FROM EventSets where

eventset_company="&dbadmin("admin_company")&"

Set dbeventset = dbConn.Execute(sqlLoginQuery)

end if

end if

The next page that the administrator will advance to is the Admin Main page.

This is the page the supper admin will see:

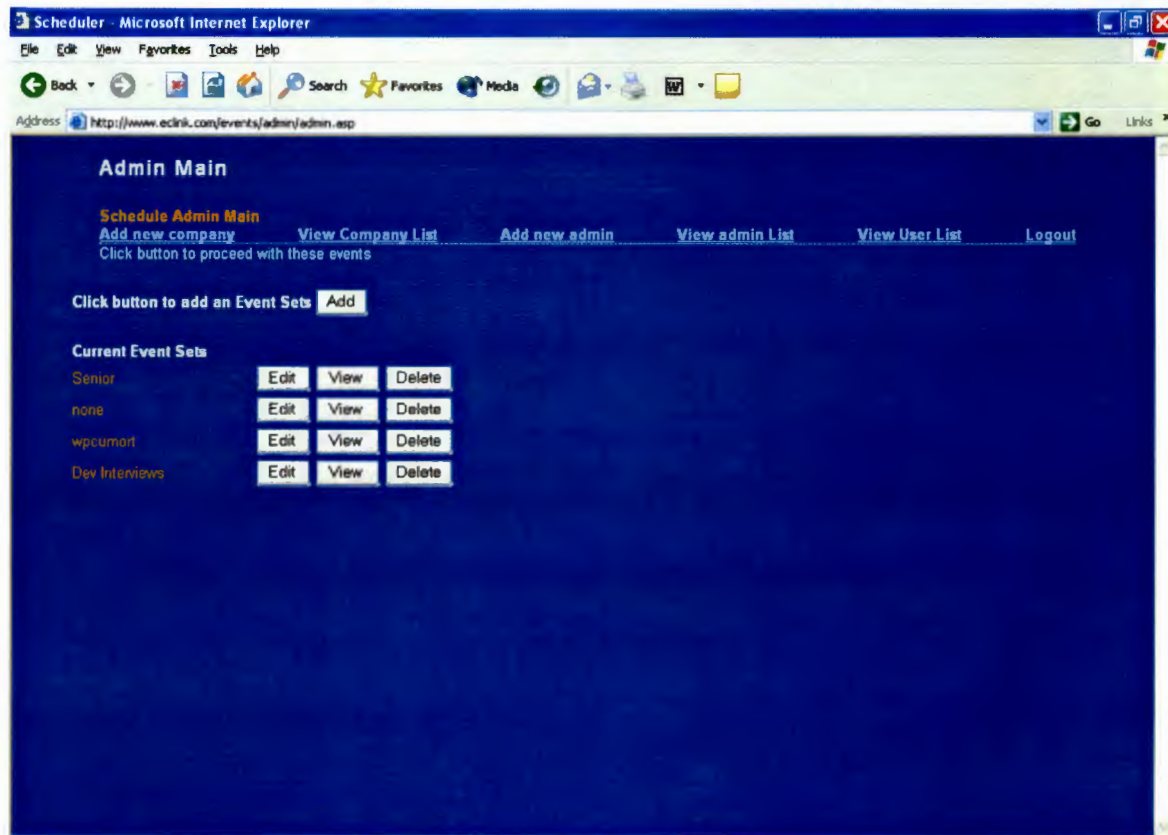


Figure 6 Super Administrator Main Page

This page is where the super administrator can go to the pages where he or she can add new companies, view all companies, add new admin, view all admin, view the user lists, add event sets, edit event sets, view event sets, delete event sets and log out.

The normal administrator main page is a little different. This is the page that a normal administrator will see:

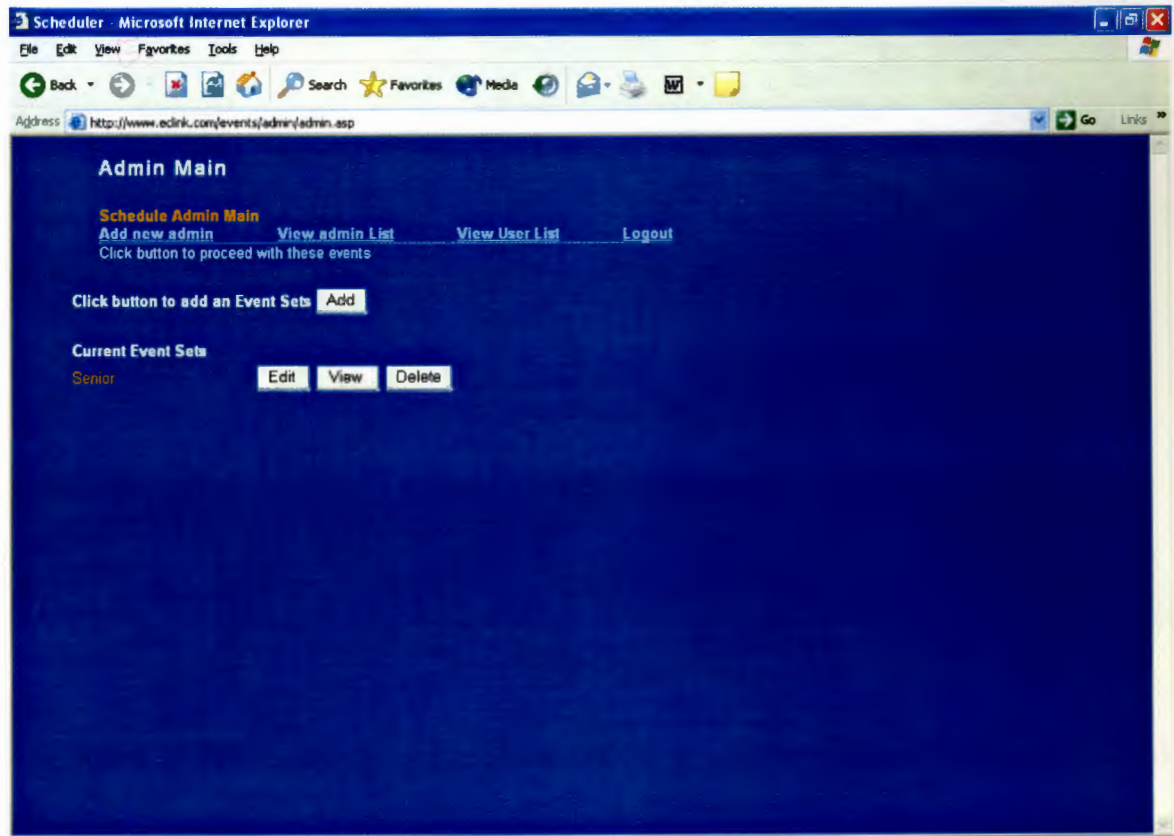


Figure 7 Normal Administrator Main Page

This page is very similar to the super administrator main page without the view company link and the add company link.

Now that the administrator is logged in he or she will need to add an event set so he or she can add events under this event set. Here is the page that this can be done on:

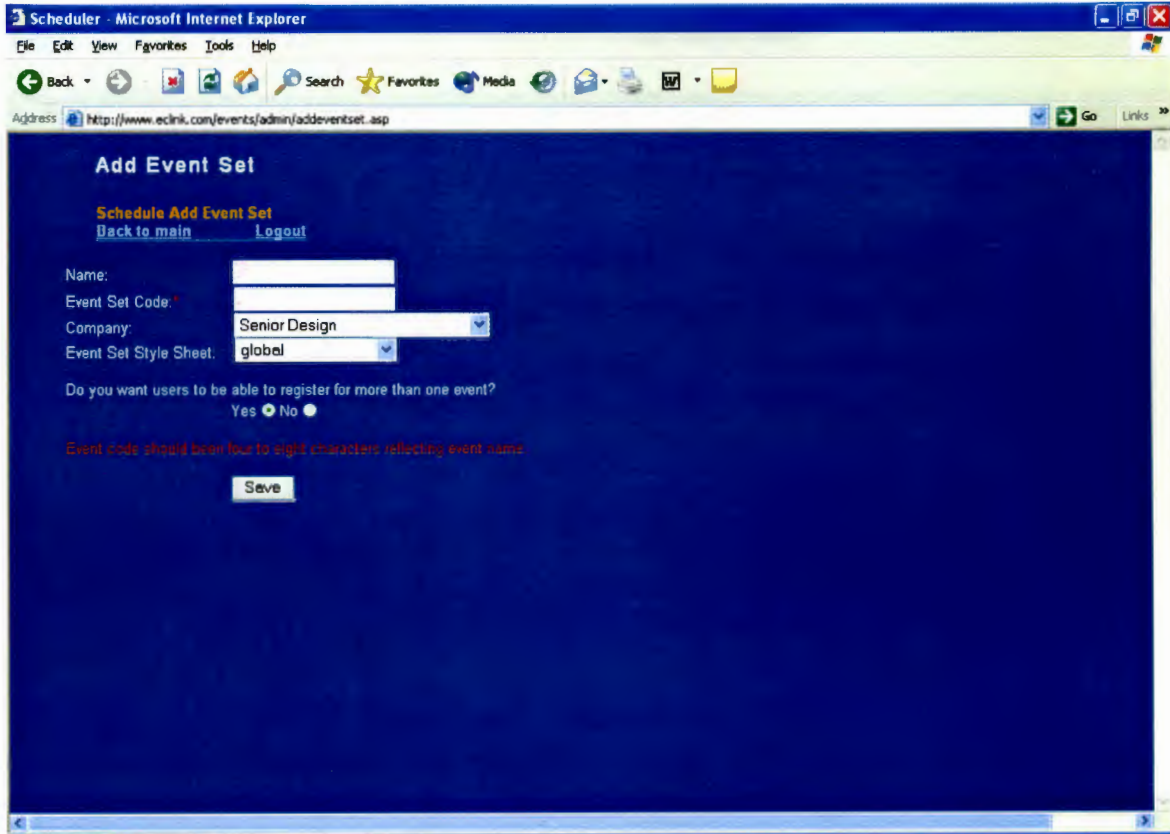


Figure 8 Add Event Set Page

The administrator is required to input an event name, event set code (Event code should be four to eight characters reflecting event name), the company (not an option if the are a normal administrator), event set style sheet, and if the invitee can sign up for more than one event. This information will be added to the database at this time.

The next step in event creation is adding an event; this is done when the administrator clicks of the view event set on the Admin Main page. Here is the page that the View Event Button takes the Administrator:

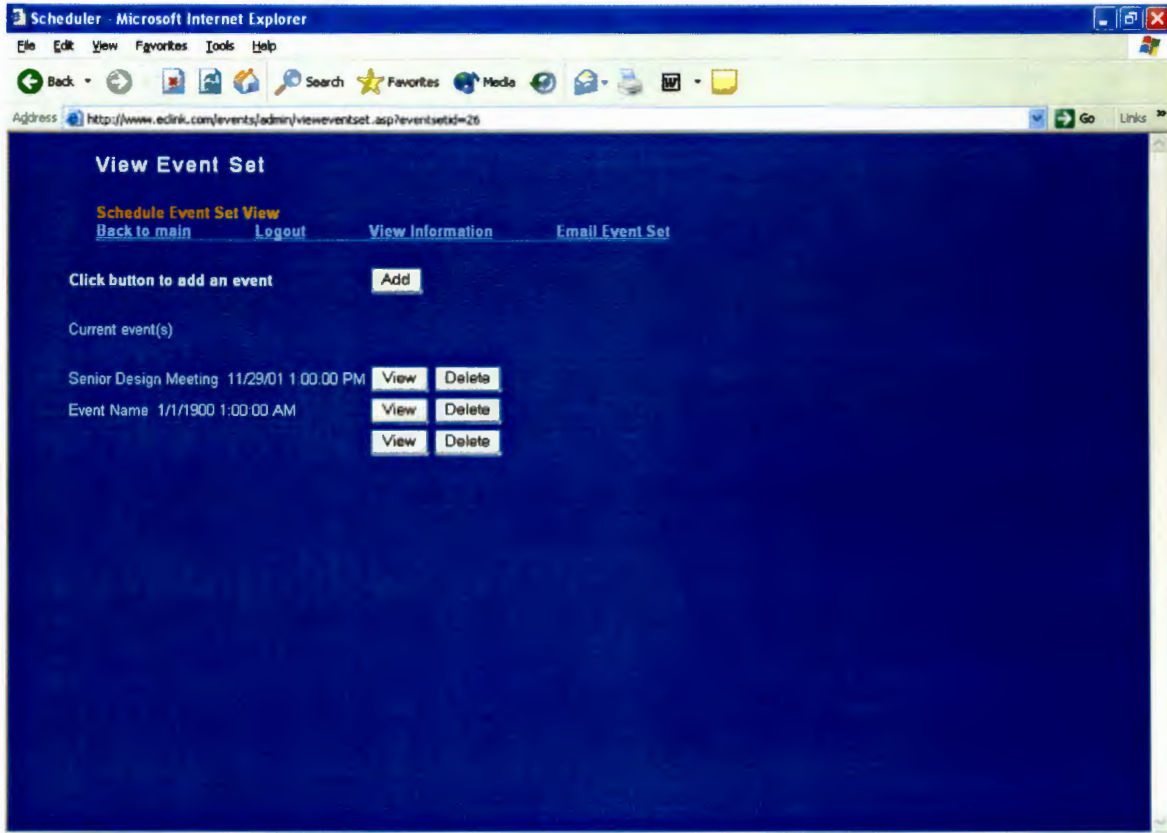


Figure 9 View Event Set Page (Without Information)

On this page the administrator can go back to the main page, logout, view more information, email event set, add an event, view event detail, and delete an event.

This is the View Events Set page after the administrator clicks on the View Information. It now shows the current people that have registered for all events in this event set, their email address and the additional filed that were added by the Administrator.

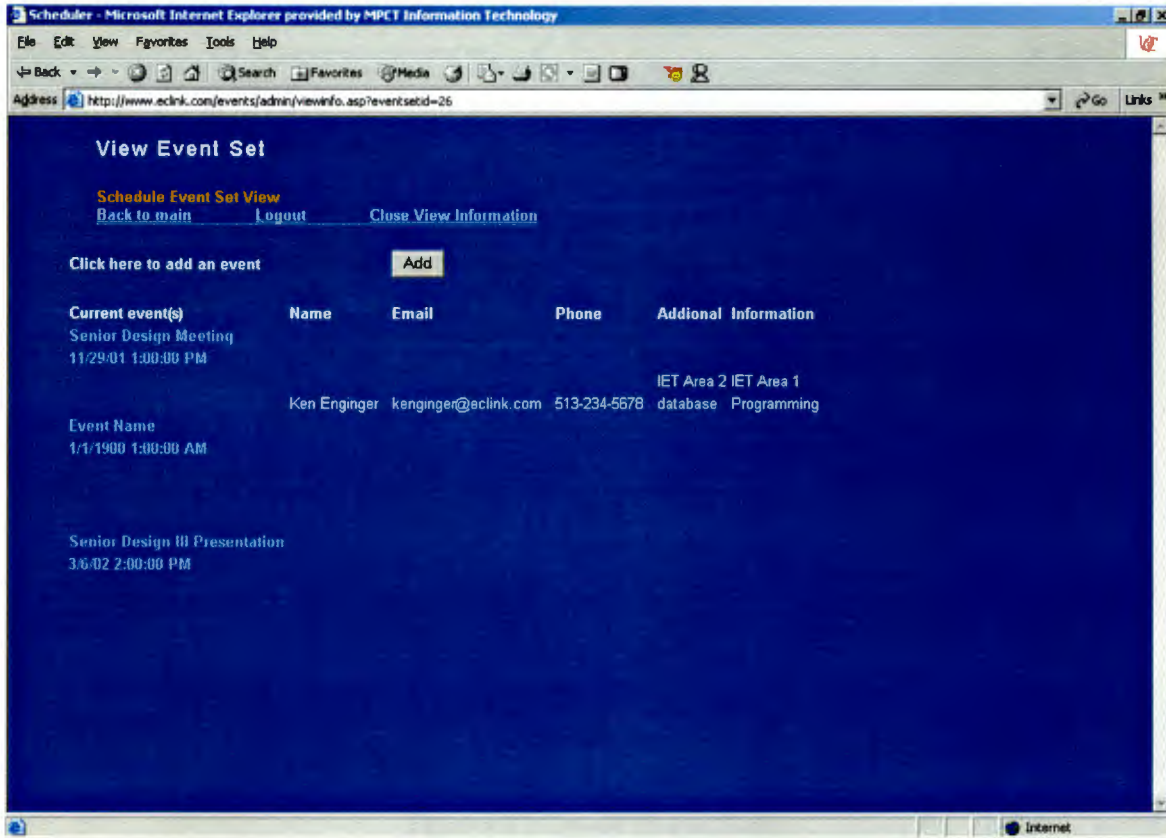


Figure 10 View Event Set Page (With Information)

If the administrator closes the view information page then he or she will then have the option to email event set again. This link will take you to a page where the administrator can email the users in one event set or all of the users in every on in the event set.

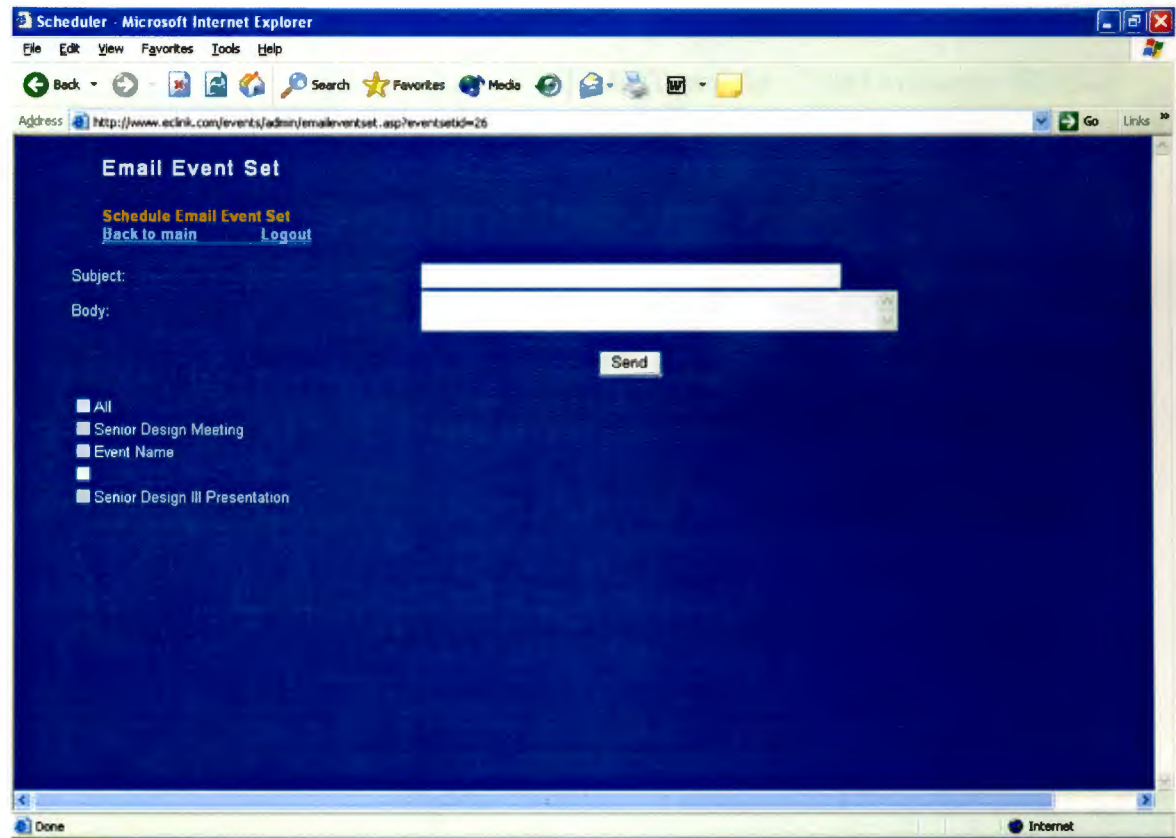


Figure 11 Email Event Sets

Here is the code that goes through the database and finds each user that is in the event check to receive the email and sent them this email

```
sqlLoginQuery = "SELECT * FROM eventsets WHERE  
                eventset_id='"&request("eventset_id")&'"  
Set dbeventset = dbConn.Execute(sqlLoginQuery)  
sqlLoginQuery = "SELECT * FROM companys WHERE  
                company_id='"&dbeventset("eventset_company")&'"  
Set dbcompany = dbConn.Execute(sqlLoginQuery)  
company=dbcompany("company_email")
```

```

if request("all")=1 then
    sqlLoginQuery = "SELECT event_id FROM events WHERE
        eventset_abv=""&dbeventset("eventset_abv")&""
    Set dbevent = dbConn.Execute(sqlLoginQuery)
    do while not dbevent.eof
        sqlLoginQuery = "SELECT user_id FROM eventmbr WHERE
            event_id=""&dbevent("event_id")&""
        Set dbeventmbr = dbConn.Execute(sqlLoginQuery)
        do while not dbeventmbr.eof
            sqlLoginQuery = "SELECT user_email FROM users
                WHERE user_id=""&dbeventmbr("user_id")&""
            Set dbuser = dbConn.Execute(sqlLoginQuery)
            email=dbuser("user_email")
            Set objMail = Server.CreateObject("ecMail.SMTP")
            strBody = request("body")
            strsubject = request("subject")
            objMail.Host    = "smtp1.eclink.com"
            objMail.From    = company
            objMail.SendTo  = email
            objMail.Subject = strsubject
            IP = request.servervariables("Server_Name")
            IP = IP&Request.ServerVariables("Path_Info")
            eventset_abv=request("eventset_abv")
            URL="http://" & IP
            URL=replace(URL,"sendmaileventset.asp","events.asp")
            URL=replace(URL,"/admin","")
            URL=URL &"?"& "e="& dbeventset("eventset_abv")
                &"&"& "email="&email
            objMail.Body = strBody & vbCrLf & vbCrLf & vbCrLf &
                URL
            err = objMail.Send
        end while
    end while
end if

```

```

        Set objMail = Nothing
        dbeventmbr.movenext
    loop
dbevent.movenext
loop
else
for each uid in request("users")
    sqlLoginQuery = "SELECT user_email FROM users WHERE
        user_id=" & uid & ""
    Set dbuser = dbConn.Execute(sqlLoginQuery)
    email=dbuser("user_email")
    Set objMail = Server.CreateObject("ecMail.SMTP")
    strBody = request("body")
    strsubject = request("subject")
    objMail.Host    = "smtp1.eclink.com"
    objMail.From    = company
    objMail.SendTo  = email
    objMail.Subject = strsubject
    IP = request.servervariables("Server_Name")
    IP = IP & Request.ServerVariables("Path_Info")
    eventset_abv=request("eventset_abv")
    URL="http://" & IP
    URL=replace(URL,"sendemaileventset.asp","events.asp")
    URL=replace(URL,"/admin","")
    URL=URL & "?" & "e=" & dbeventset("eventset_abv") & "&" &
        "email=" & email
    objMail.Body = strBody & vbCrLf & vbCrLf & vbCrLf & URL
    err = objMail.Send
    Set objMail = Nothing
next
end if

```

Response.Redirect

```
("vieweventset.asp?eventsetid=" & request("eventset_id") & "") %>
```

The next step is for the Administrator to add an event; they will need to click on the **add** button. This button will take the administrator through a wizard where he or she will step through the add event fields. Here is the first step through the wizard:

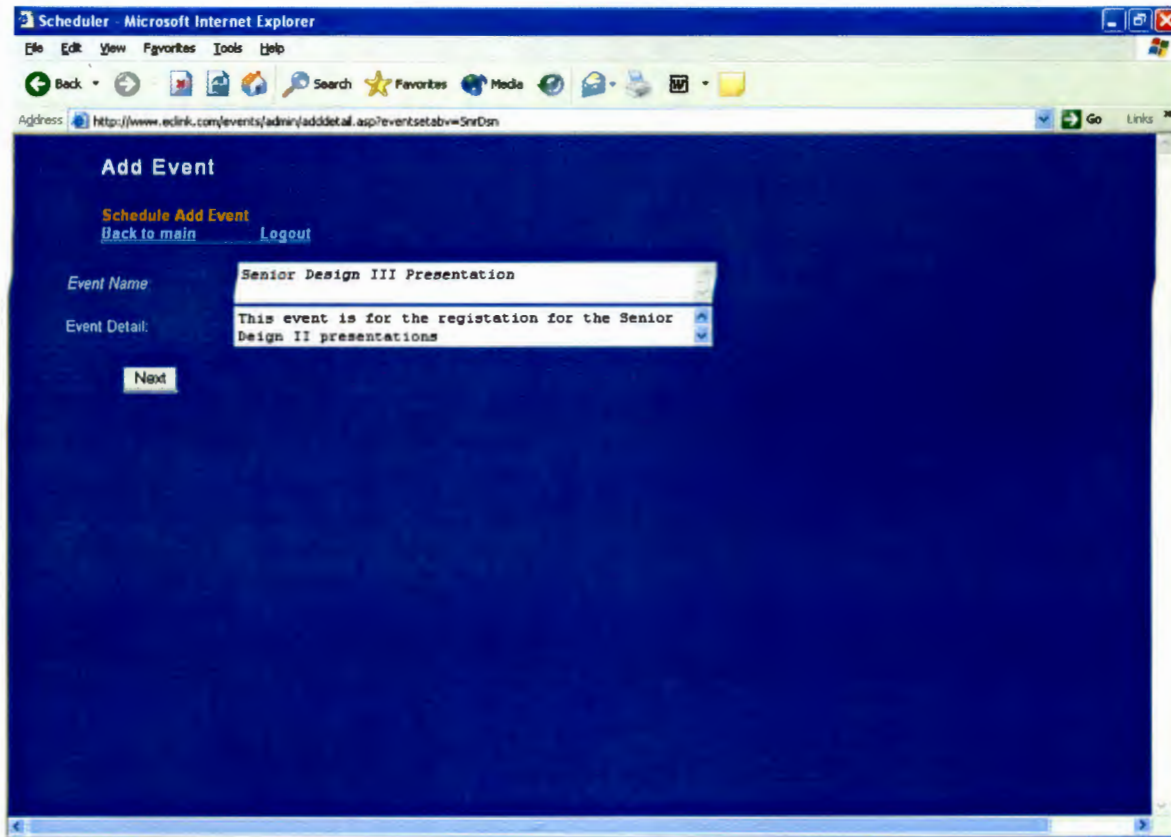


Figure 12 Add Event Step 1

Step 1 is where the event name and detail is inputted and inserted into the database.

The next step in the event adding is to input the capacity, the time zone, the start time/date, the ending time/date, and the number of additional fields. These additional fields are for event specific fields that are not already asked for. An example of one of these fields might be major.

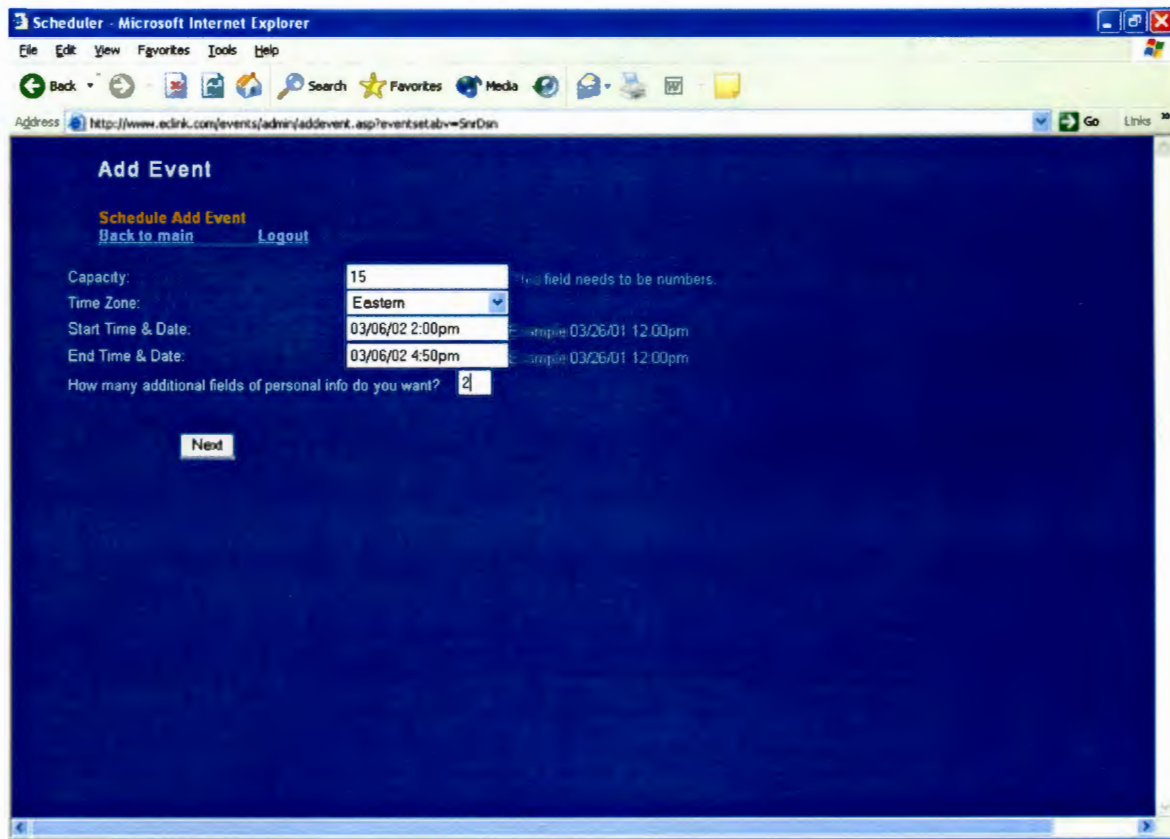


Figure 13 Add Event Step 2

If the proper information is not added to the input boxes then the administrator will get this popup box.



Figure 14 Required Field Check

The next step is where the additional field name and Comments are added.

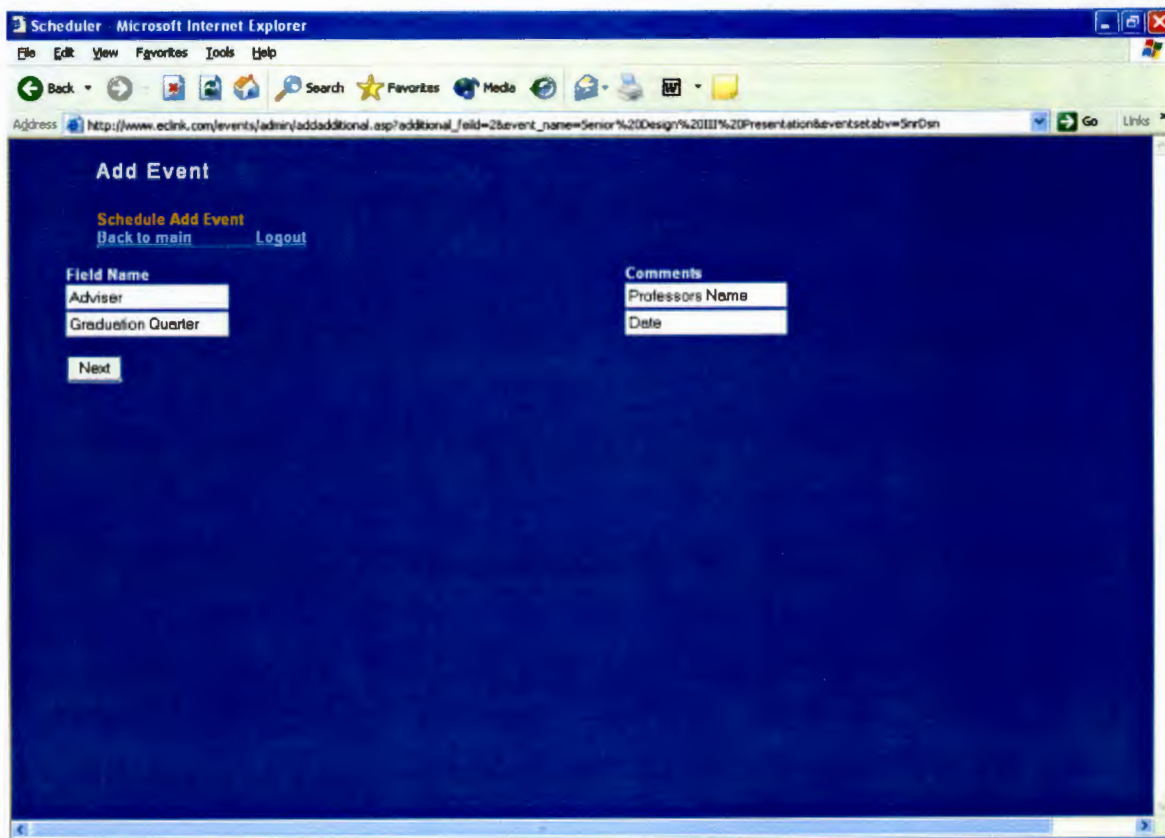


Figure 15 Add Event Step 3

The next step is where the email information is entered. There is default information given for both the Invitee and Confirmation emails. In the Invitee email this message will give the invitee the URL for the Invitee side of the Scheduler. In the Confirmation email the registered user will see the event information, which is in brackets in the text box.

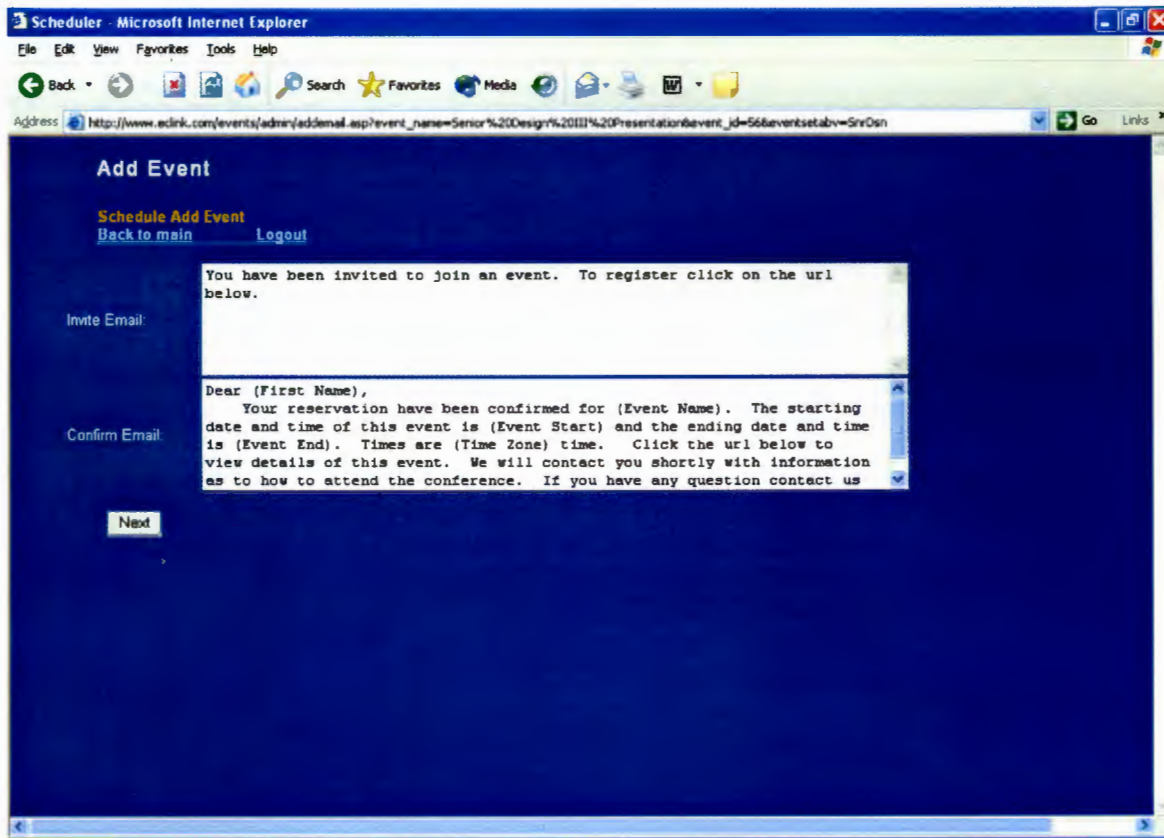


Figure 16 Add Event Step 4

The last step in the event creation is the Event Description; this information is an in-depth description of the event and its details or what ever the administrator wants to add. There is also a place to input how many, if any, duplications of this event the administrator will want. This feature was not in the original plans but Jerry wanted to it add for events that are the same but at different times. An example of one of these events are interviews.

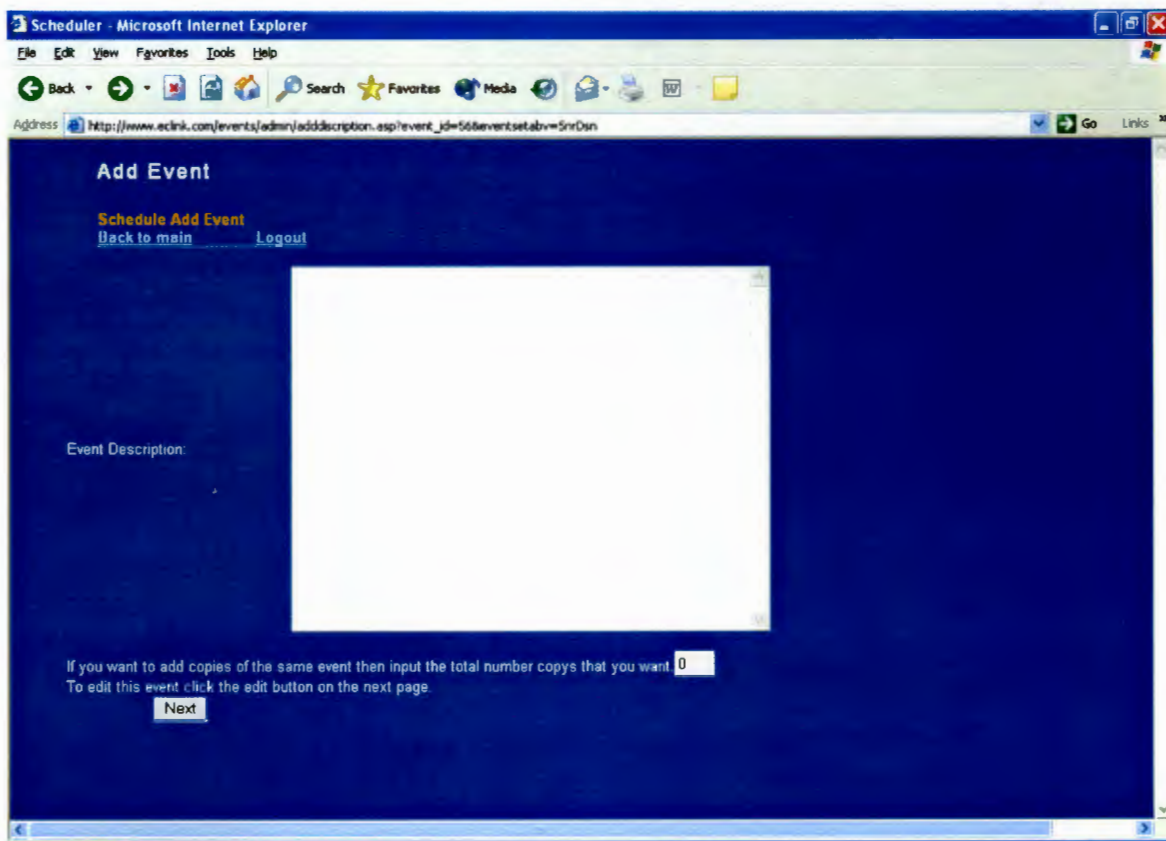


Figure 17 Add Event Step 5

The event has now been created and all the information has been inserted into the database. The Administrator will be at the View Event Set page; here he or she will see the event that he or she created. He or she will need to click on the view button to view his or her event.

At this time the administrator will be taken to the View Event page where he or she will see the information for his or her event. At this page the administrator can go back to the main page, logout, email user if they exist, edit all sections of the event and invite users.

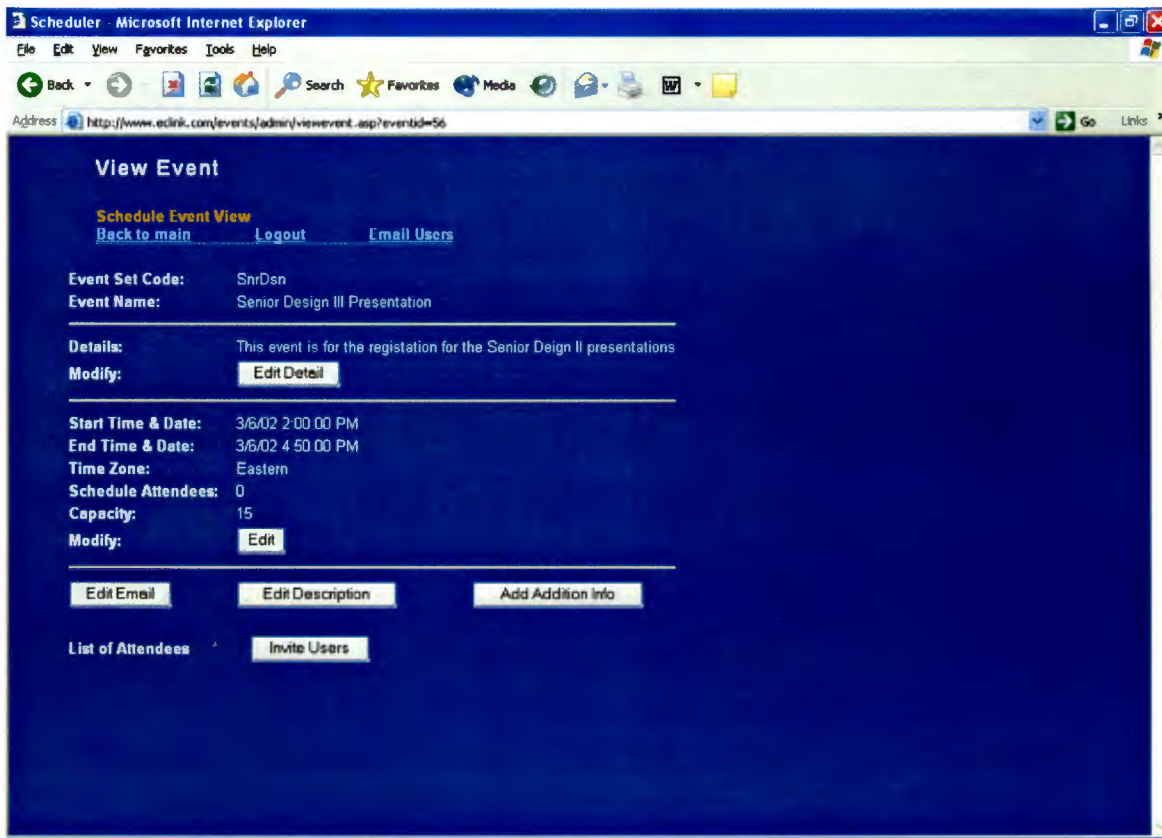


Figure 18 View Event (No Users)

If the event has users, the administrator can cancel these reservation is they desire; if he or she does this them the user will receive an email telling them that they are on longer registered for this event.

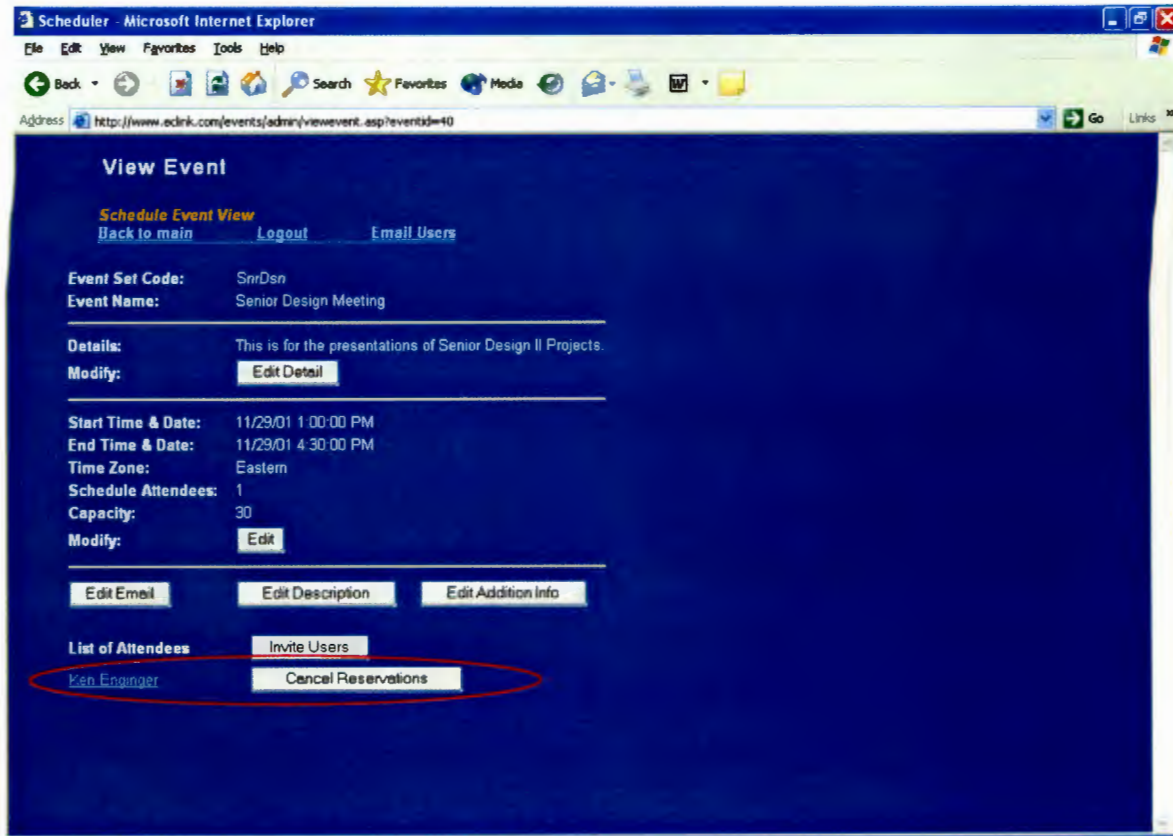


Figure 19 View Event (With Users)

Now that an event has been created the administrator can now invite uses; to do this they need to click on the **invite user** button.

Here the administrator can invite existing users or choose how many new user to invite to this event. Under the menu the administrator will how many spaces are filled if any. If the event is full the administrator will not be able to add any more people.

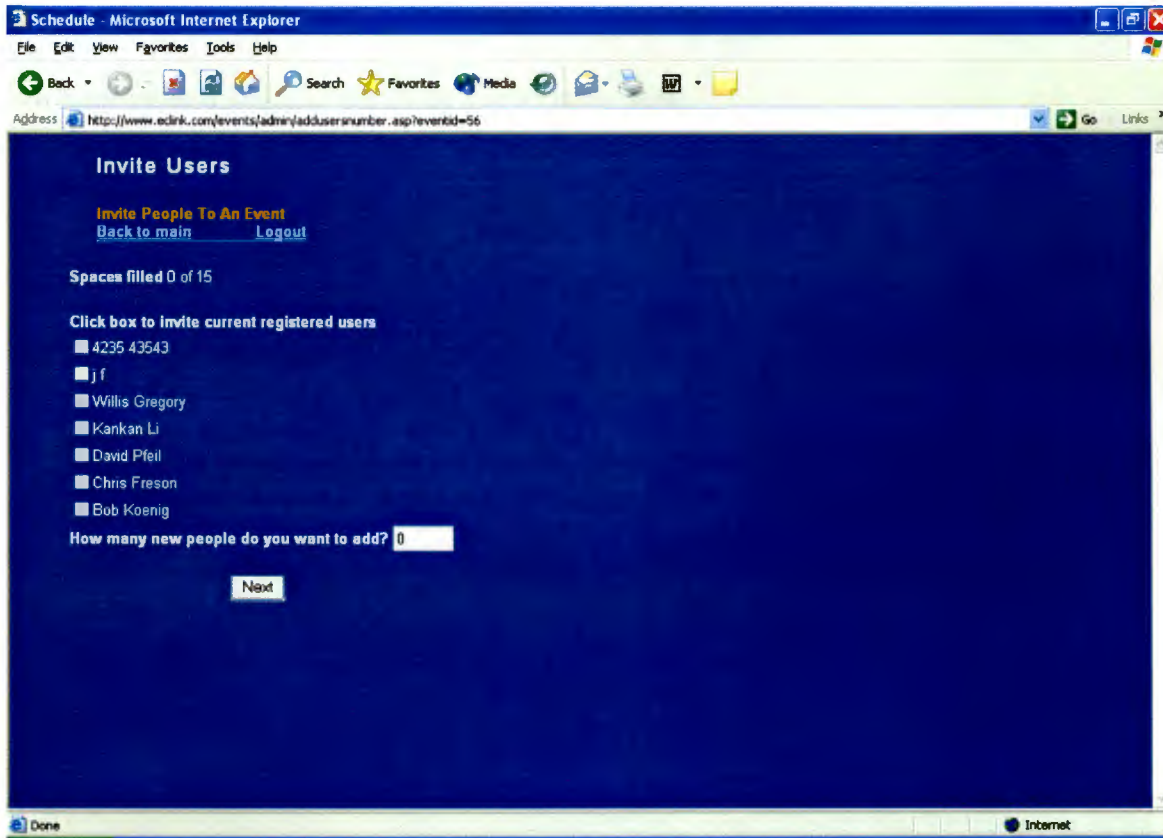


Figure 20 Invite Users (Current Users)

On this page the administrator will fill in the text box that include Email (required), first name, last name, and phone number. Once the administrator clicks the **invite** button the new invitees and the old invitees will get an email with the event information and a link to the event registration page.

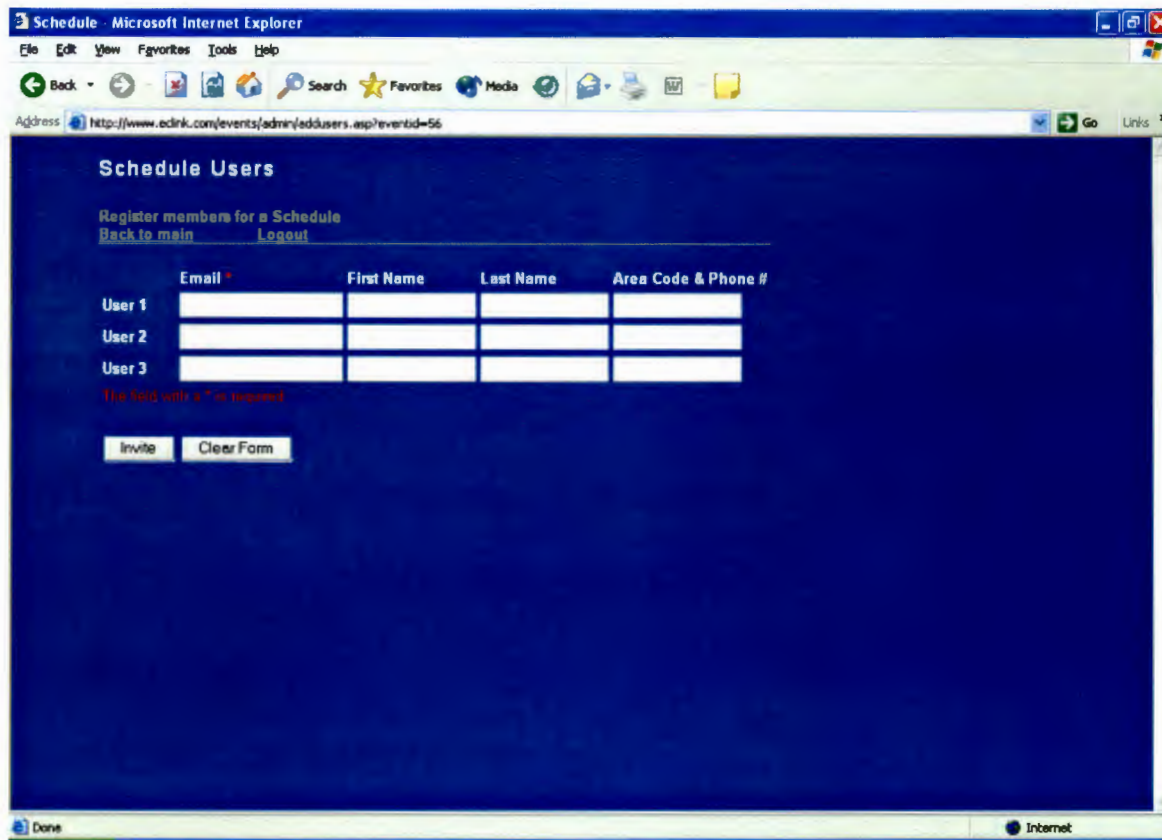


Figure 21 Invite Users (New Users)

6.4 User Side

Now that an event has been made it is time for the invitees to register for their desired event or events. Like the administration side the user will be connection to the database, but there might not be authenticated. As mention before the invitee can get to the scheduler from a URL in an email or they can get to it from the *ec link* and Wright-Patt web sites. The user side like the administration side is simple and straightforward.

This allows the user to register for an event and cancel for an event in minutes. Here is the flow chart of the user side:

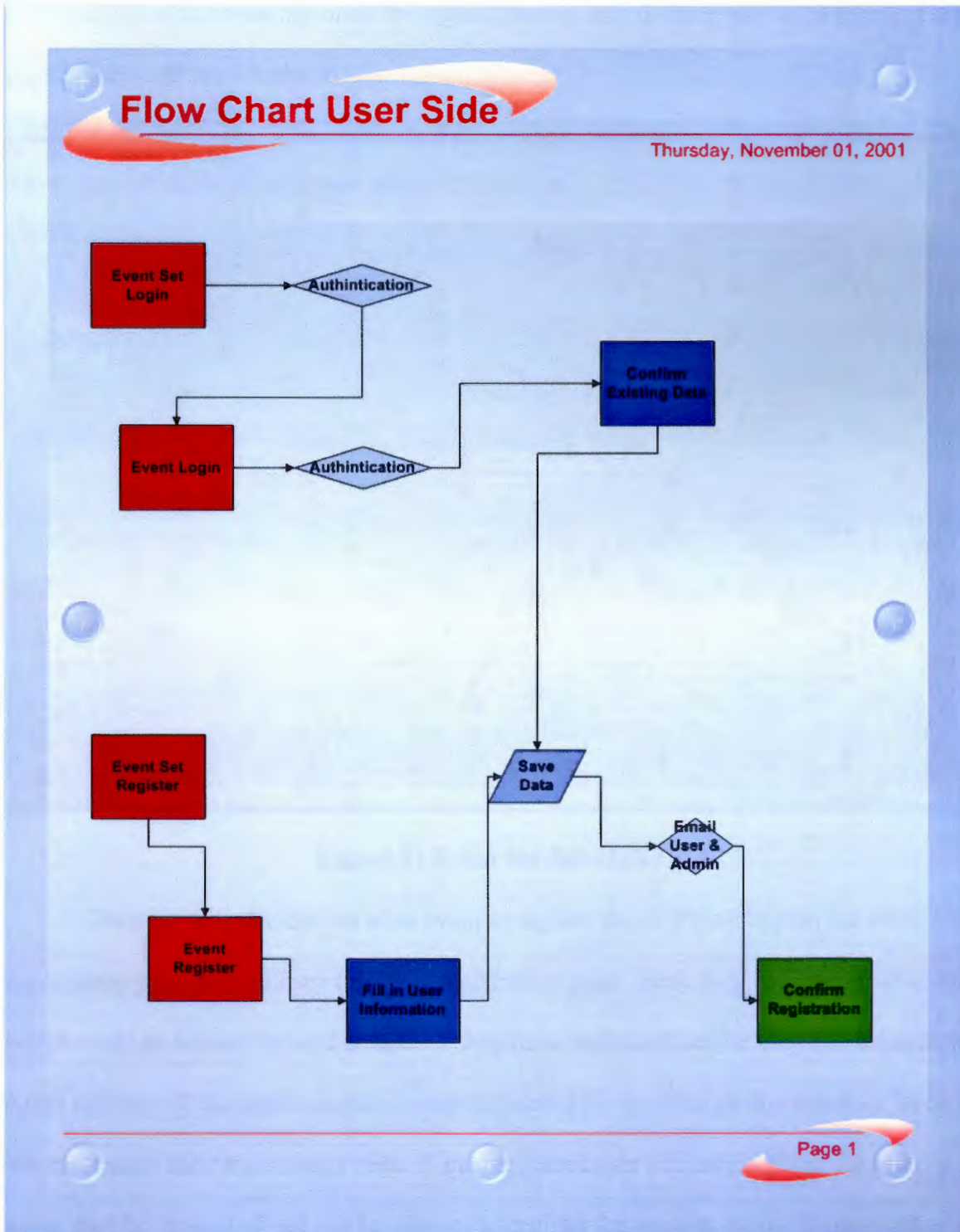


Figure 22 User Flow Chart

The user has two different options when they get to the Event Set or Event Scheduler. He or she can either input their email address if he or she has registered for an event before or he or she can click the register button next to the event set or event. This page is the event set scheduler:

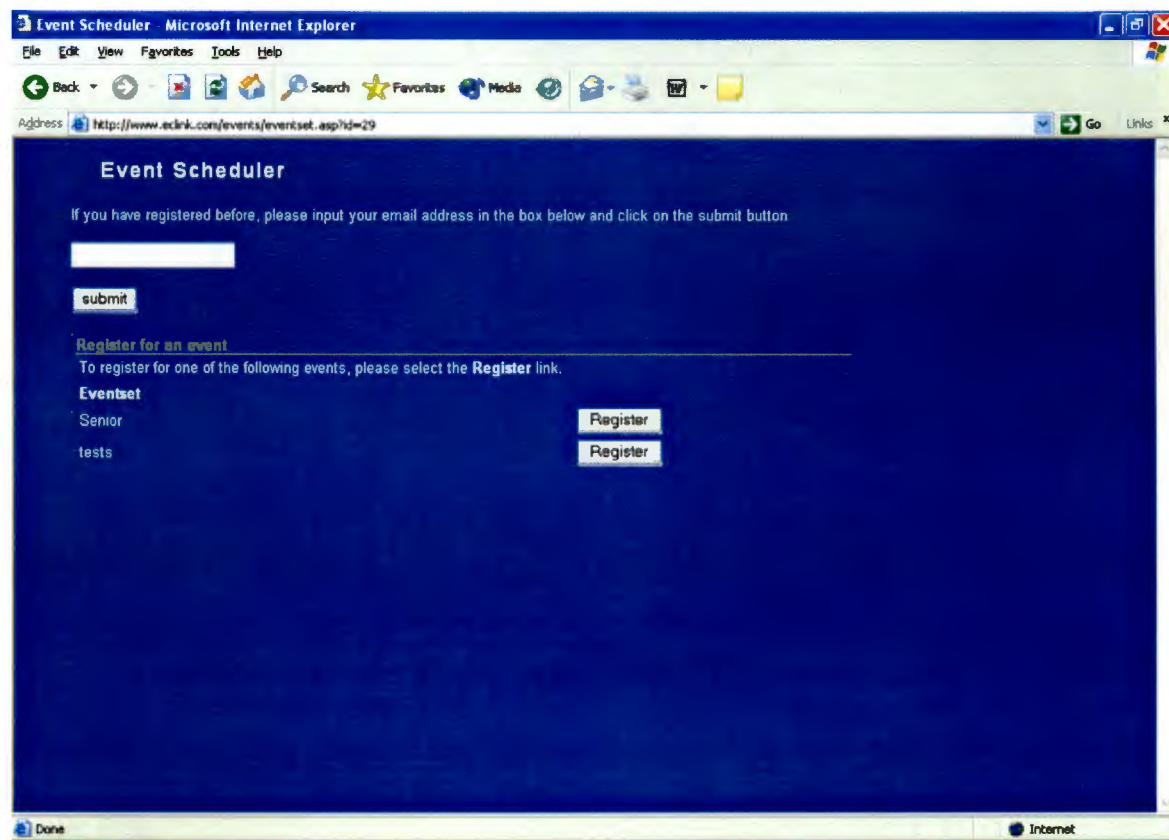


Figure 23 Event Set Scheduler

Once the user decided on what event to register for or if they bypass the event set registration page they will see the event registration page. Here they will need to decide which event to register for and or again if they have registered before they can input there email address. If the invitee is has already registered for an event in this event set he or she can cancel their registration here. If the registered user can only register for one event then he or she will see not be allowed to register for another event. If the event is

full in place of the register button the text **full** will be displaced.

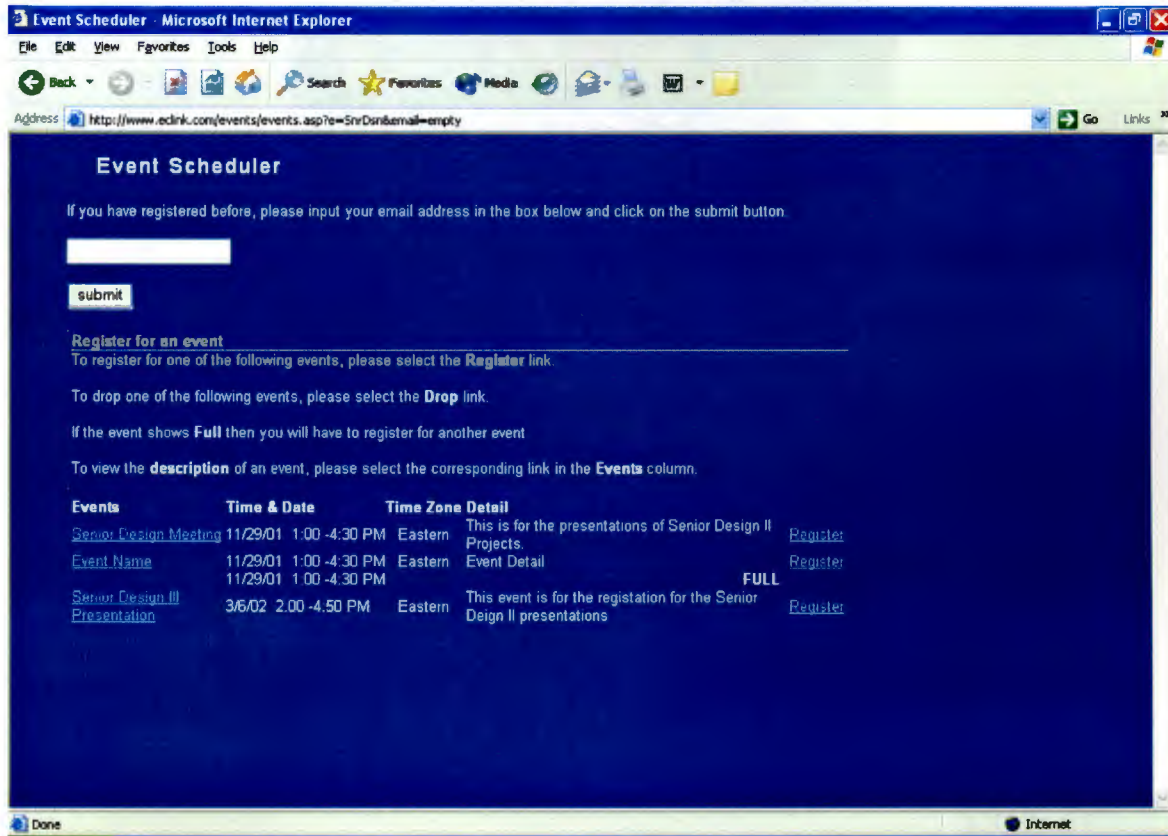


Figure 24 Event Scheduler

If the invitee wants to get more information about the event he or she can click on the **event title**.



Figure 25 Event Detail Description

Now that the invitees have chosen his or her event he or she will need to input his or her information in the text boxes. If the invitee is a returning user he or she will see his or her information in the text boxes. The returning user will not see his or her information in the additional text boxes that is different for this event. Below these text boxes is a description of the event that he or she is registering for. The invitee will then need to click on the **complete registration** button on the bottom of the page. They will be prompted to confirm their information; this is done when the invitee will need to click the **ok** button. If all of the required information is not inputted they the user will be told what are missing. Once all of the required fields are filled, then the user will need to click the **complete registration** button again. If their email address is already used they will be taken back to the registration page. All of the fields except for the extra fields and

the email address will have the information that was inputted already. This will not affect the returning user.

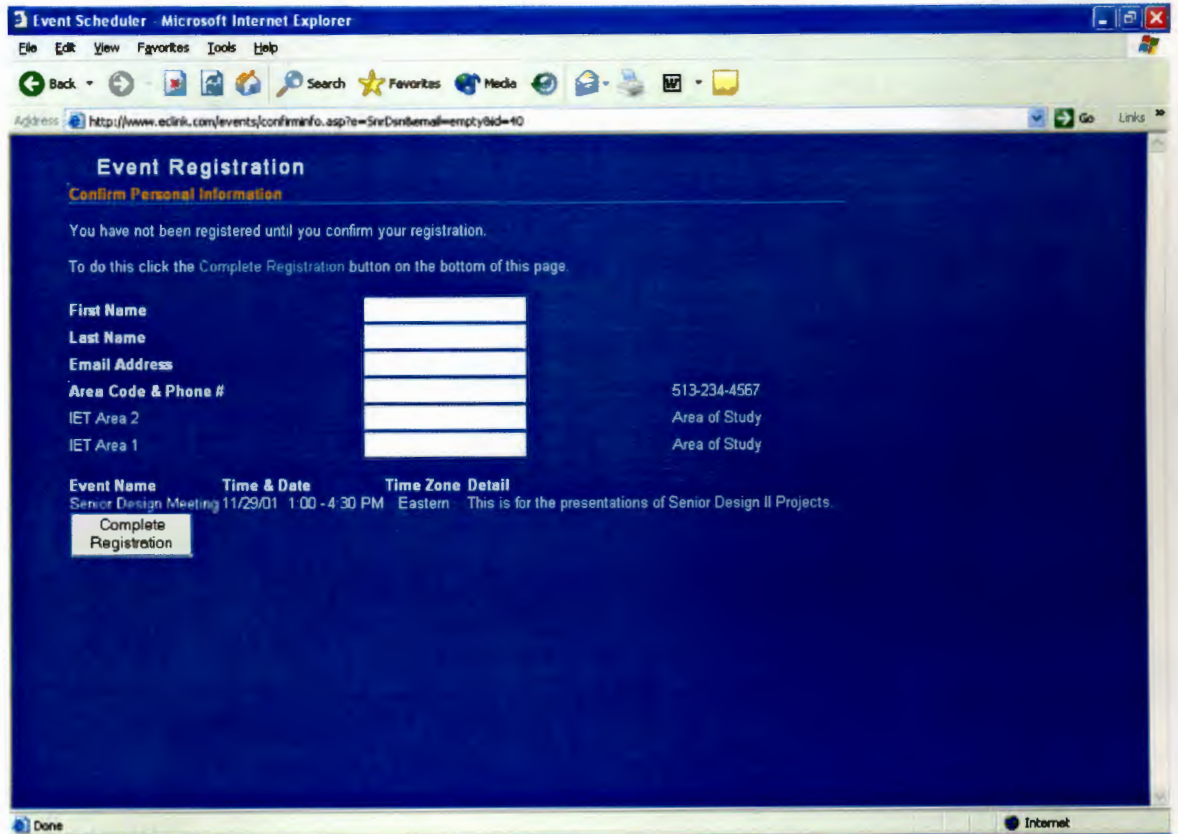


Figure 26 Event Registration User Information

The final page of the invitee side is when the user and the administrator get an email confirming their reservation. The invitee's email will have the event information along with the URL link to back to the event registration page. The administration's email will have the user's information along with the event that they have registered for. Here the invitee can either go back to the event set registration or the event registration page. As mention before the user can come back here to drop this event or register for another event if allowed.

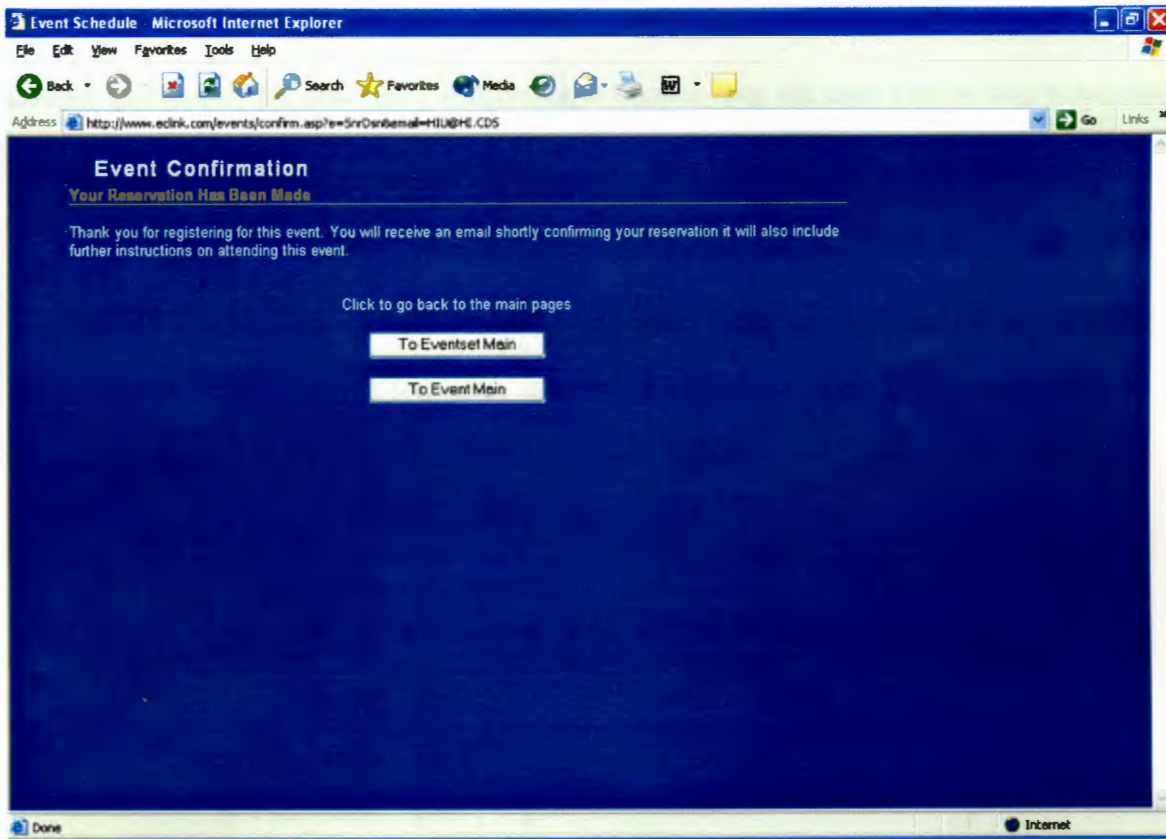


Figure 27 Event Confirmation

6.5 Features

On many of the pages there are pieces of code that are used to allow the pages to flow smoothly, to prevent error, and prevent illegal data in the database. This code included a database safe function, confirmation of data before it is committed to the database, and confirmation of deletion and data insertions. These bits of code are programmed in either VB Script or JavaScript.

The first of these and most used is the database safe function:

```
Function DBsafe( strDB )
```

```
    If Not VarType( strDB ) = vbString Then DBsafe = strDB : Exit Function
```

```
    DBsafe = Replace( strDB, """, """)
```

```
End Function
```

This function will replace double quotes with single quotes; this prevents any illegal double quotes into the database. If this would happen then the data will be corrupted and not be retrieved correctly.

The next bit of code is a script used to check for required data before inserted into the database; it is called easyform.js. There are bites of code that I used to check to see if the are in the correct form or required. This is used to check to see if the time/date is in the correct form:

```
function isDate(s) {  
    s = replace(s, "/", "");  
    s = replace(s, "-", "");  
    s = replace(s, ":", "");  
    s = replace(s, "AM", "");  
    s = replace(s, "PM", "");  
    s = replace(s, "am", "");  
    s = replace(s, "pm", "");  
    s = replace(s, " ", "");  
    return isNumeric(s);  
}
```

This is used to check to see if the phone number is in the correct form:

```
function isPhone (value)  
    {if (value.length > 10){value = numericize(value);}  
    //Check if first digit is 0 or 1. Invalid phone number.  
    if ((value.charAt(0) == '0') || (value.charAt(0) == '1')) {return false;}  
    //Should be 3 digit area code + 7 digit phone number, as a 10 digit string  
        if (value.length >= 10){if (isNumeric(value)){return true;}  
        else {return false; }}  
    else{return false; }return true;}
```

This is used to check to see if the email is in the correct form:

```
function isEmail (value){ var i,ii; var j; var k,kk; var jj; var len;
```

```

// Check valid email
// Must have a "@" and a "." to be valid.
// Must have at least 1 character before "@"
// Must have at least 1 character after "@" and before "."
// Must have at least 2 characters after "."
if (value.length >0){
    i=value.indexOf("@");
    ii=value.indexOf("@",i+1);
    j=value.indexOf(".",i);
    k=value.indexOf(",");
    kk=value.indexOf(" ");
    jj=value.lastIndexOf(".")+1;
    len=value.length;
if ((i>0) && (j>(1+1)) && (k===-1) && (ii===-1) && (kk===-1) &&
    (len-jj >=2) && (len-jj <=3)) {}
else {
    return false;
}}
return true;
}

```

The last of these functions is the switch to connect the text boxes to the functions; if the specific text if passed over the a textbox will popup with the returned text:

```

switch(arg) {
    case "req":
    case "required":
        if (elementValue == "")
            return " is a required field.";
        break;
    case "date":
        if (!isDate(elementValue))

```

```

        return " must be a valid date.";
        break;
    case "email":
        if (!isEmail(elementValue))
            return " must be a valid email address.";
        break;
    case "phone":
        if (!isPhone(elementValue))
            return " must be a valid phone number.";
        break;

```

The last check is confirmation of deletion and data insertions:

```

<input type=button value=" Complete&#13;&#10;Registration" onClick="if
(confirm('Do you want to register for this event?')==true) {if
(validateForm('frmRegister')) { document.frmRegister.submit(); }}">

```

This text displays a confirmation for the invitee to ok or cancel his or her reservations.

With completion of the last bits of code all of the deliverables were met and the project has been completed.

7. Conclusions and Recommendations

As this project concluded I felt very strongly in that it was completed to the best of my skills, fulfilling the need of Jerry Felix, and meeting all of the deliverables that I gave this project. I have learned more than I thought was capable in the few months that I completed this program. With the knowledge of web programming, database and project management I feel that I can repeat this work in the professional world.

The skills that I used throughout this project include web-programming, database, and project management. The programming that I used was VB Script, JavaScript, and HTML; this was done using Visual Interdev. The database that I used was Microsoft SQL; this was because of the great size capabilities and easy connections from the web. I learned how to work from scratch to get a project done in the time allowed. I can use these skills to complete a project in a professional environment.

This project was given to me from Jerry Felix, a partner at *ec link*; he gave me the criteria verbally and in writing. He gave me the freedom to do it whatever way I wanted, but it had to meet his goal. Throughout the project Jerry gave me some changes; these changes were recommendations that took place throughout the project. These changes made it a little tough, but with the help of my fellow co-workers I overcame them.

If any more recommendations needed to be added then they would have taken place before the project was completed. Changes cannot be made without the customer's permissions, now that the project has been sold and in use. The only recommendations that I would consider is a page to create style sheets and an example or image showing

8. References

1. Butler, Jason. "ASP Data Access for Beginners". Http:// www.15seconds.com
2. Chipman, Mary and Gunderloy Mike. *SQL Server 7 In Record Time*. Sybex, 1999.
3. Fedown, Alex; Francois, Brain; Harrison, Richard; Homer, Alex; Murphy, Shawn; Smith, Robert; Sussman, David; and Wood, Stephen. *Active Server Pages 2.0*. Olton: Wrox Press Ltd, 2000.
4. Jones, Mike. *Active Server Developer's Journal (2000)*. Element K Press, 2000.
5. Moten, Lewis. "SQL Preparation Functions". Http://www.asp101.com
6. Reselman, Bob. *Active Server Pages 3.0 By Example*. Que, 2000.
7. Shaw, James. "I will cover your ASP" Http://www.coveryourasp.com
8. Siegers, Pieter. "ADO and SQL Error Trapping in ASP". Http://www.asptoday.com
September 21, 2000
9. Sledge et all, Orryn and Spenik, Mark. *Microsoft SQL Server 7 DBA Survival guide*. Sams, 1999.
10. Walther, Stephen. *Active Server Pages Unleashed*. Sams, 1998.