Master of Science in Computing at the Institute of Technology, Sligo.

Michael Hannon

April 2007

Project Title:

To design, develop, implement and test an intranet for a third level satellite campus.

Submitted in part-fulfillment of requirements for:

M. Sc. in Computing

Submitted to:

Mr. Greg Doyle, Project Supervisor.
Dept. of Computing and Networking,
Institute of Technology, Carlow.

Submission Date:

16th April 2007

Submitted by:

Michael Hannon, Assistant Registrar,
Galway-Mayo Institute of Technology,
Dublin Road, Galway.

Awarding Institution:

Institute of Technology,
Sligo.

Acknowledgements

A special word of thanks to my wife Regina and son Adrian for their support over the past number of years.

A sincere thanks to my supervisor Greg Doyle at the Institute of Technology, Carlow, and to Terry Young at the Institute of Technology, Sligo.

CONTENTS

Chapter 1 INTRODUCTION

1.1 General 4

1.2 Background 4

1.3 Rationale 5

1.4 Objectives 6

1.5 Timeframe 7

1.6 Report Structure 8

Chapter 2. THEORY AND INTRODUCTION TO THE INTRANET

2.1 What is an Intranet? 9

2.2 Why build an intranet? 12

2.3 Intranet Protocols 13

2.4 Intranets and Security 14

2.5 Security for Microsoft IIS 15

2.6 Designing an intranet 16

2.7 Measuring the success of an intranet 17

Chapter 3. TECHNOLOGIES USED

3.1 Internet Information Server (IIS) 19

3.2 HTML 19

3.3 Scripting Languages 20

3.4 ASP 21

3.5 SQL 23

3.6 Open Database Connectivity (ODBC) 25

3.7 ActiveX Data Objects (ADO) 26

3.8 DSN 27

3.8 VMWARE 27

3.9 ASP Upload 27

Chapter 4. DATABASE DESIGN

4.1 Analysis of Requirements 28

4.2 Databases 31

4.3 Microsoft Access 2000 31

4.4 Development of the system Database 32

4.5 Creating the Database 34

4.6 Relationship of Tables 37

Chapter 5. DESIGN and IMPLEMENTATION OF THE SYSTEM 39

5.1 Linking The System 41

5.2 Building the System 43

5.3 The Intranet Site 47

5.4 The Administration Section 51

5.5 Securing the application 56

5.6 How it all works 59

Chapter 6. DISCUSSIONS and CONCLUSIONS 79

Bibliography 83

APPENDIX 86

List of Figures

Fig. 01 *Timeplan*

Fig. 02 *Diagramatic Representation of how ASP works*

Fig. 03 *MS Access launch screen*

Fig. 04 *MS Access Design View*

Fig. 05 *MS Access add data*

Fig. 06 *Admin table in design view*

Fig. 07 *Admin table populated*

Fig. 08 *Relationships between the tables*

Fig. 09 *System overview*

Fig. 10 *Database overview*

Fig. 11 *The administration section overview*

Fig. 12 *ODBC Data Source Administrator dialogue box*

Fig. 13 *Create New Data Source dialogue box*

Fig. 14 *ODBC MS Setup dialogue box*

Fig. 15 *Example of includes file*

Fig. 16 *ranet site*

Fig. 17 *Overview of administration section*

Fig. 18 *ASP pages are the only web service extensions allowed.*

Fig. 19 *Default permission on the website is set to 'read only'*

Fig. 20 *NTFS permissions allowed*

Fig. 21 *Homepage design*

Fig. 22 *Course Notes*

Fig. 23 *Courses table from database*

Fig. 24 *Stages in a course*

Fig. 25 *Modules in a stage*

Fig. 26 *Lectures linked to a module*

Fig. 27 *Search page*

Fig. 28 *Login to admin section*

Fig. 29 *The Super Administrator Menu*

Fig. 30 *Updating courses admin menu*

Fig. 31 *New course form*

Fig. 32 *Linking stages to courses*

Fig. 33 *Confirmation that course was added to database*

Fig. 34 *Select courses to edit*

Fig. 35 *Edit course details form*

Fig. 36 *Live site to show extra stage was added*

Fig. 37 *Add new module option from admin menu*

Fig. 38 *Form to add a new module*

Fig. 39 *Linking a module to a lecturer*

Fig. 40 *Confirmation that module was added to the database*

Fig. 41 *Menu option to associate module with a course*

Fig. 42 *List of avaiable courses dynamically populated from database*

Fig. 43 *Drop down list of available modules*

Fig. 44 *Linking the module to one or more years*

Fig. 45 *Confirmation that the module was associated with a course*

Fig. 46 *Menu option to ad notes to a module*

Fig. 47 *Drop down list of modules to add notes to*

Fig. 48 *Editorial rights allowed to notes linked to a module*

Fig. 49 *Form to add new notes*

Fig. 50 *Confirmation that notes were added to the database*

Fig. 51 *Menu option to change module name and associated lecturer*

Fig. 52 *Form to change module name and associated lecturer*

Fig. 53 *Menu option to add a new course to the database*

Fig. 54 *Form for new course details*

Fig. 55 *New course appears on the live site*

Fig. 56 *Menu option to delete existing courses from the database*

Fig. 57 *Form to select course to be deleted*

Fig. 58 *Menu option to edit an existing course*

Fig. 59 *Form to edit an existing course*

Fig. 60 *Menu option to delete an existing module*

Fig. 61 *List of modules from dynamically populated drop down list*

Fig. 62 *Confirmation that module has been deleted*

Fig. 63 *Admin table showing administrators and super-administrators*

Fig. 64 *Drop down list of all tables in the database*

Fig. 65 *Module Notes table*

Fig. 66 *Lecturer Account Maintenance menu*

Fig. 67 *Form to change lecturer details / rights*

Fig. 68 *Confirmation that lecturer details have been updated*

Fig. 69 *Menu option to add a new lecturer*

Fig. 70 *Form for details of new lecturer*

Fig. 71 *Form fields validated using client-side javascript*

Fig. 72 *Assigning a lecturer rights to a module*

Fig. 73 *Lecturer / administrator logon menu*

Fig. 74 *View of a lecturer's rights to module notes*

Fig. 75 *True / false button controls display of notes on live site*

Fig. 76 *Live site showing Lecture 1 notes disabled*

Fig. 77 *Form to link a module with a lecturer*

Fig. 78 *Admin menu showing two modules linked to a lecturer*

Chapter 1 INTRODUCTION

1.1 General

The Internet is a worldwide network of computers joined together by telecommunication lines. It is a communications medium that allows users to send and receive electronic information between computers linked to the network.

The intranet, in comparison, is a private computer network based on the communication standards of the Internet. It is a smaller version of the Internet setup so that only members of an organization can access it.

This project aims to setup an intranet for the satellite campus of the Galway-Mayo Institute of Technology (GMIT), located in Letterfrack.

1.2 Background

This M. Sc. in Computing is part of the 'Training of Trainers' programme. Since 1990, dedicated funding has been provided to higher education institutions in Ireland to undertake training of trainers.

'The Department has indicated to the Institutes of Technology that staff development and training should be viewed as a planned and continuous learning process aimed at maintaining and enhancing the knowledge, skills, attitudes and effectiveness of all staff to assist them in meeting institutional needs and goals.'[1]

A minimum of nine programmes were offered over the period ranging from degree to masters award. The delivery was structured by a specific Institute taking responsibility for a particular programme. The Institute of Technology in Sligo has responsibility for the M. Sc. in Computing.

60% of this programme was delivered though taught modules and the remaining 40% is attributed to a project. A choice of modules were offered from which candidates

select any four. The modules were offered at low activity periods during the academic year to minimise lecturers out-of-class time. The periods were generally early September, during the Christmas and Easter holidays. In general a candidate would only be able to study one module per year, but depending on the delivery schedule two modules were sometimes possible. Each module consisted of three weeks of intensive class contact at the above times. Assessment consisted of an assignment at the end of each week followed by a module examination in May/June.

The modules selected, and successfully completed by the author are as follows:

- WWW Development
- Networks
- Human Computer Interaction (HCI)
- Database Technology.

1.3 Rationale / Hypothesis

The Galway-Mayo Institute of Technology has a student body of approximately 9000 students. It is a multi-campus institute with campuses at Castlebar, Cluan Mhuire, Mountbellew and Letterfrack. All programmes at the Letterfrack campus are linked to furniture, ranging from design and technology to education.

Letterfrack is located remotely 50 miles West of Galway. While it shares a number of computer services with the main campus it does not have it's own intranet. This is a service/facility the staff have been requesting over a period due to the amount of electronic communication that takes place on this campus between staff and students, between management and staff, and between staff. A number of staff share information with students on a student shared drive on the network. Other staff members have set up their own personal websites. Staff also use a staff shared drive on the network for accessing key documents.

The main college intranet is poorly supported and maintained. Staff find it impossible to have content uploaded. Content management is centrally controlled. The design of the site is regarded as poor. It is generally accepted that the staff at the Letterfrack

campus interact much more with their students and colleagues via email/shared drives and web sites than the staff on the main campus.
An intranet, specifically designed for the Letterfrack campus would provide a structure to the information flow, encourage all staff members to use it, properly managed it would keep all information up-to-date and provide a one-stop portal for students and staff to access all information in relation to programmes and other activities of relevance.

The rationale for setting up a Letterfrack Intranet is to:

- Support this M. Sc. in Computing with a live and worthwhile project.
- Provide for a more detailed information flow than what is possible on the college intranet and using email.
- Provide a means of managing modularisation and other curricular developments such as a common first year curriculum resulting in several lecturers delivering the same subject.
- Keep information up-to-date.

1.4 Objectives

1 To develop a project to demonstrate competency in the taught modules undertaken as part of this M. Sc. in Computing.

2 To set-up a key resource and communications platform for the campus.

3 To allow the intranet sites content to be easily stored and managed in a database.

4 To design and develop an easy-to-use interface to allow editors to add, remove or modify content.

5 To enable staff and students to find and use documents and web pages.

6 To control access to the site and address security issues.

7 To carry out a review of the site using heuristics.

1.5 Timeframe

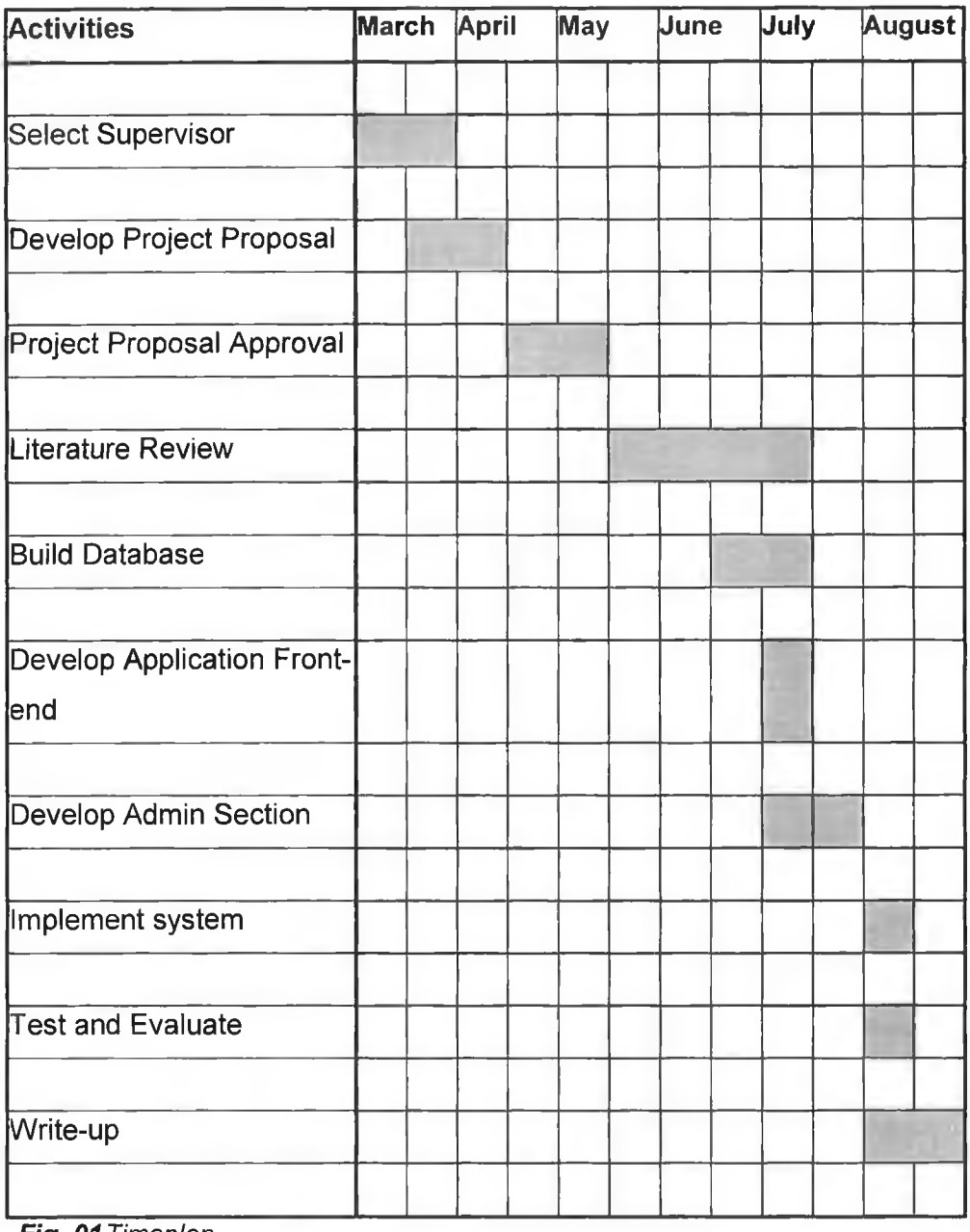


Fig. 01 *Timeplan*

All dates refer to 2006.

1.6 Report Structure

Chapter 1 gives an introduction and deals with the background and objectives of this project.

Chapter 2 deals with the theory of intranets and the technologies used in this project.

Chapter 3 deals with the technologies used in the project.

Chapter 4 concentrates on the database design.

Chapter 5 is concerned with the design and implementation of the system.

Chapter 6 discusses conclusions and also outlines possible further work in this area.

Chapter 2. THEORY AND INTRODUCTION TO THE INTRANET

2.1 What is an Intranet?

An intranet is a private network, unlike the internet. It is built using the internet's TCP/IP protocols for communications. TCP/IP protocols are capable of running on many different hardware platforms and cabling systems. What makes an intranet is the software protocols, the hardware protocols are not as significant.

Intranets can function alongside other local networking technology. In several organizations existing 'legacy systems' are being integrated into an intranet [Bangar, 2003]. This is possible through a variety of tools. Common Gateway Interface (CGI) scripting is sometimes used as a means to access legacy databases from an intranet.

According to Gonzalez [1997], communication over computer networks first occurred in the 1960's with electronic mail on time-sharing computers. People communicated by sending messages using the same mainframe computers through dumb terminals connected to the mainframe or through dial-up telephone lines. At that time networked computers allowed people connect with each other like the telephone, but computers facilitated the transfer of digital data such as text and computer code.

One major drawback in the early development of the web was that most computer languages could not communicate with each other. Spoull and Kiesler [1993] document the difficulty with the following observation:
"Difficulties are: the researcher cannot get access to data; if he can he cannot red them, he does not know how good they are; and if he finds them good, he cannot merge them with other data."

Two trends led to the development of the internet: the development of the internet protocols, and the development of computer-aided reading of electronic documents. In relation to the latter, one of the most important developments was the creation of SGML, a system for describing digital document types.

Charles Goldfarb invented SGML in 1974 as a system for formalizing the structure of documents. SGML provided a vendor neutral, formal, international standard for information interchange.
The creation of hypertext was another milestone. The idea behind hypertext is that it provides a way to link information together so that it makes sense to the person using the system. Vannevar Bush is credited with developing what is now known as hypertext in 1945. Ted Nelson devised the word hypertext to describe what Bush had developed. Van Dam and Nelson added to the hypertext system in 1968 by developing WYSIWYG (what you see is what you get).

The father of the internet, Tim Berners-Lee wrote in 1990 that he envisioned a system that would enable people to:

- connect to various electronic systems;
- view an online employee telephone book,
- construct a "Personal Skills Inventory", listing projects and areas of expertise;
- access documents linked by categories and projects, keywords and authors;
- read newsgroups and create links from one newsgroup to another;
- share key learnings at the conclusion of projects;
- search the system for information; and
- generate mailing lists to keep people informed of changes.

What Berners-Lee described in 1990 is virtually everything associated with intranets today. He wrote that he wished to create a system that was 'future-proof', meaning that it was portable, or supported many platforms. The system would have universal access and the links could map to any existing database structure.

The word 'intranet' can have different meanings to different people. To some it is the physical medium used to transport data. To others it is a set of web pages with content. And to others it is a virtual place where a number of applications can be combined to provide improved communications, work flow and transaction processing within organizations.
All of the above definitions are correct. However, intranets are dependent upon the collection of web protocols and standards that unify the languages used to deposit and retrieve information from a web server.

Three standards define the web: URLs, HTTP, and HTML. These standards are responsible for locating, accessing, and displaying information.

URLs (Universal Resource Locators) are a system to specify the location of an internet server. HTTP is a network protocol for transferring information and for enabling hypertext. HTML is the language that is understood by web browsers.

Computers on a network pass data to one another because they use another protocol called Transmission Control Protocol/Internet Protocol (TCP/IP). All computers on an internet and on intranets communicate using TCP/IP.

Intranets are becoming popular as a result of the exponential growth of the internet. Vast numbers of people are using the internet for communicating with the outside world, for gathering information and for doing business, be it ecommerce or otherwise. People now recognise that the components that work so well on the internet could work equally well internally, hence the growth in intranets.
Using TCP/IP intranets can be accessed remotely, such as from home or while traveling. Connecting to an intranet in this fashion is similar to connecting to the internet, except that you are connecting to a private network and not the world wide web.

An intranet includes web sites which contains web pages that are hyperlinked. The home page indicates what is available on the site. The intranet can also include links to email applications, databases, and files stored on legacy systems. The advantage of an intranet is that the web browser provides a common interface to a variety of other applications.

The main difference between an intranet and the internet is down to security systems. An organization's intranet is protected by firewalls – hardware and software configured to allow only certain people access the intranet for particular purposes. Firewall technology allows people on an intranet to use internet resources, but blocks outsiders from getting into the intranet.

Intranets can be used for anything that an existing network is used for and possibly more. It is very easy to publish information on the web, such as organization procedures and policies. Databases can be developed with easy to build front-ends that use the web and programming languages such as ASP.

2.2 Why build an intranet?

According to Teelan [2003], if you believe an intranet is important to your business, there are a few key points to consider. An intranet supports and encourages three different types of information: official, project or group, and personal. All have value to communication within the organization. He advises that you recognise and manage these different uses instead of trying to eliminate some. This can be done he says by using policies, architectures and education.

It is vital that the quality of official information delivered over the intranet is managed. Quality is a management issue and the intranet is a tool that can assist with this. It is also important to identify the people and roles involved in the development and publication of content. Policies need to be developed in relation to information access and editorial control. The intranet infrastructure must support diversity to cater for all groups within the organization.

According to Bangar [2003], intranets allow people to work together more easily and more effectively. Groupware software allows people to collaborate on projects; to share information; to do videoconferencing; and to establish secure procedures for production work.
The ease with which information can be shared, and with which people can communicate with one another will continue to drive the building of intranets.

Some organizations/companies build intranets and measure the return on investment (ROI). The intranet is treated as a business support and as such it must deliver measurable performance and remain accountable to the investment. If an intranet site is not being measured, it risks failing the needs and demands of employees and management [Finding ROI, 2005]. This report goes on to say that measuring the

precise value of an intranet is almost impossible as it is, at best an imperfect science. Nonetheless, many organizations are measuring the ROI of their intranet investment. The author would argue that it is the less tangible, inherent value of an intranet that is most difficult to measure but potentially this yields the most valuable benefits.

Intranets have a number of technical advantages. Berlind (1996) lists them as:

- intranets allow system managers to leave all the data on the system where they exist. They do not have to convert data to another system.
- The web browser minimises the number of interface mechanisms that people have to know how to use.
- The amount of training required for intranet participation corresponds to the nature and purpose of participation. People who use the intranet only to look at documents need very little assistance.

In relation to a third level campus, like GMIT Letterfrack, some of the key drivers for building an intranet include knowledge management and content management. Content management may be described as the process by which content is created, managed and published to the website. Software solutions are available for this purpose but in-house solutions, like this project are adequate for small organizations starting off on this route.
The advantages of having a central repository of official documentation for this or any organization are obvious. Staff benefit by having their official course notes always available to students. Similarly students benefit by being able to track lectures and access course notes when convenient. Management benefits by having official policies and procedures available to all concerned and by having access to an effective communications tool for both staff and students.

2.3 Intranet Protocols

An intranet is different to any other kind of private network as it is based on TCP/IP: the same protocols that apply to the internet. TCP/IP refers to two protocols that work together to deliver data. They are the Transmission Control Protocol (TCP) and the Internet Protocol (IP). When information is sent across an intranet, the data is broken

down into small packets. The packets are sent independently through a series of switches called routers. When all the packets arrive at their destination, they are combined into their original configuration. The Transmission Control Protocol breaks the data into packets and combines them at the receiving end. The Internet Protocol deals with the routing of the data and is responsible for the packets arriving at their correct destination.

Data sent within an intranet is required to be broken into packets of less than 1,500 characters. TCP breaks the data into packets and adds a checksum to the packet. The checksum is based on the precise amount of data in the packet.
The packet and checksum is placed in IP wrappers. The wrappers have the address on the intranet or Internet where the data is to be sent. The packets travel between networks by intranet routers. The routers determine the most efficient pathway for each packet to travel.

When the packets arrive at their destination, TCP calculates and compares the checksum for each packet. If the checksums do not match, TCP discards the packet and requests that the original packet be resent.
When all the non-corrupt packets are received, TCP assembles them into their original form. Packets are reassembled using the header information in the packet.

2.4 Intranets and Security

An intranet is open to attack from people intent on destruction. The open nature of the Internet and TCP/IP protocols expose an organization to attack. A variety of security measures are required, including hardware and software combinations to offset this threat.

A firewall is the main line of defense. This is a hardware – software combination that controls the type of services to run on an intranet. A proxy server can be used in building a firewall. A proxy server allows system administrators to track all traffic going in and out of the intranet.

Server based virus checking is used to scan files entering the intranet to ensure that they are virus free.

Authentication systems are a key aspect of intranet security schemes. Authentication systems ensure that anyone trying to logon to an intranet or any of it's resources is the person with permission to do so. User names and passwords are the typical authentication systems used.

People can be barred from getting objectionable material on an intranet through the use of server-side site blocking software.
Another way to ensure that unacceptable data does not get into the intranet is to use a filtering router. A filtering router examines the IP address and header information in every packet coming in to the intranet and only allows those packets that have addresses or other data that the system administrator has decided should be allowed.

2.5 Security for Microsoft IIS

Microsoft Internet Information Services (IIS) version 6 is included with the Windows Server 2003 operating system. IIS allows administrators to host Web and FTP sites on a Windows server. This is the operating system the proposed intranet will be operating on. This section deals with the security features of IIS version 6.

The Internet Information Services (IIS) Manager in Windows server 2003 allows IIS specific security configuration so that the maximum level of security can be implemented on the web server. This includes controlling authentication, and defining web site permissions and securing communication channels. The extra security features of IIS 6 can be configured to prevent one web application from affecting other web applications running on the same IIS 6 server.

FTP service security must be increased by configuring IIS if an FTP server is implemented. This involves limiting authentication to anonymous access and configuring the FTP folder structure so as to reduce attacks on the disk system.

Additional security features of IIS 6 include:

- The default installation locks down the service to limit the server to hosting static HTML.
- Privilege requirements are reduced
- Automatic health warning. All work processes are monitored to detect failed processes.
- Application isolation implementation. It is possible to isolate each web application from others running on the server.
- Security of HTTP.sys improved. HTTP.sys is moved to kernel mode from user mode, which results in better performance and a higher level of security for web-based applications.
- ASP.net security supported.
- Default settings secured. All options are not installed by default. An interface is provided to manage which applications are supported by the web server.
- Improved protection against attacks.

2.6 Designing an intranet

According to Neilsen (2005), intranet homepages have become very similar in their basic layout. He goes on to say that intranets that look the same can nonetheless differ drastically in usability due to different features and content.

The standard features of an intranet homepage include:

- Top horizontal bar, generally used for the logo and site navigation.
- Left column used to locate a menu of detailed options on the site.
- Middle area to host the most important content on the page.
- Right column which may contain a set of links or some noticeboard type information.

This particular intranet design incorporates these features from a standard template available in Macromedia Dreamweaver. The design of an intranet site Neilsen argues

is not as important as an Internet site as only the employees of an organization see it and they are all basically for the same purpose, namely to share internal information.

According to Ward (2006), the five winning characteristics of leading intranet sites are:

1. Engage. Find out what the users want and incorporate the findings into the site plan.
2. Standards. Develop guidelines and standards for using the intranet.
3. Simplify. Keep the design of the site clean and simple, without too mush animation and multimedia.
4. Measure. Measure the site using a set of Critical Success Indicators (CSIs). These measure both quantitative and qualitative performance.
5. Promote. Promote the intranet to users and motivate them to use it.

The author has attempted to incorporate these characteristics into the design but would argue that it is an iterative process and takes time to perfect.

2.7 Measuring the success of an intranet

"The right metric for intranet success is consistent use over time." (Gonzalez, 1997) The intranet for this project was created with the users in mind (staff and students). A measure of it's success will be, as Gonzalez points out: how many will be using it six months after it is launched. A heuristic evaluation could be carried out to determine the usability of the site. This is normally done in controlled environments but can also be carried out on everyday users.

It generally involves one or more people looking through a list of intranet pages. As each person finds problems they record them and categorise it as major or minor. At the end all the results are collated and the combined list of problems can be used in the redesign of the web site.

Not all people agree that heuristic inspections are worthwhile. According to Molich (2004), Heuristic inspections are cheap, simple to explain, and deceptively simple to

execute. He goes on to say that he doesn't use this method very often and does not recommend it as it is based solely on opinions.

The intranet should become the first point of contact for all queries, such as downloading travel authorization and claim forms, examination templates, external examiner database, continuous assessment schedules, list of project supervisors and their students, exam papers, lecture notes and lots more. As people use the site more the number of phone calls and email queries should diminish.

Chapter 3. TECHNOLOGIES USED

3.1 Internet Information Server (IIS)

IIS is a free server software package from Microsoft that provides a means of adding web connectivity to Windows based systems.

When looking at software to interact with an ACCESS database and IIS, it was found that Active Server Pages (ASP) would be the right choice, as ASP was developed by Microsoft with the aim of running alongside its web server IIS.

3.2 HTML

Hypertext Markup Language is the fundamental technology for controlling the structure and appearance of web pages. It is a way to format a page so that a browser knows how to display the information.

HTML is made up of tags, which are interpreted by the browser to generate the web page.

The tags are enclosed in the greater than < and less than > signs. E.g. if you want the text to be bold, you would enclose it within <B> and </B> tags.

HTML allows Tables, Forms, Images and many other items to be viewed in a browser. HTML on its own is limited in its capacity, i.e. it is static.

HTML is the main format used on the world wide web. Even though it is not a true programming language, HTML has increased in power over the years. It is regarded as a subset of XML, which is a strict language. The main advantage of XML is it allows developers design their own data format using their own terms and requirements.

HTML is so flexible that many browsers and Web applications have added their own functionality to the base HTML protocol. However, this comes with increased security risks and as a consequence efforts are being made to replace HTML with a more standardised markup language called XHTML.

While it is generally accepted that it will take many years to replace HTML with XHTML, many webmasters are embracing the new protocol and it is considered by many as the next version of HTML (HTML 5.0).

Once the page has been made in HTML it will always be displayed in the same manner. But when used with other technologies such as scripting languages it becomes dynamic and generates very interactive web pages.

3.3 Scripting Languages

Scripting languages are used in conjunction with HTML to provide Dynamic HTML. Examples of scripting languages would be JavaScript, VBScript and Perl.
There are two types of Scripting Languages Client Side and Server Side.
The desktop browser of the user who is visiting the web site executes client side scripts. The server executes server side scripts before the page is passed to the clients' browser.

Client side scripts are defined as follows:

```
<SCRIPT LANGUAGE = javascript>
```

Server side scripts can be defined in two ways:

- <SCRIPT></SCRIPT> tags with the RUNAT=Server attribute.

E.g.

```
<SCRIPT LANGUAGE = Vbscript  RUNAT=Server>
```

- Embed lines between <% %> tags

This method supports Vbscript by default.

JavaScript is the main scripting language used in this project. It is a prototype-based scripting language with a syntax based loosely on C. JavaScript has no input or output constructs of its own, it relies on a host environment into which it is embedded.

The majot use of web-based JavaScript is to write functions that are embedded in or included from HTML pages and interact with the Document Object Model (DOM) of the page to perform tasks which are not possible in HTML alone. Examples of such tasks include:

- Changing images as the mouse cursor moves over them.
- Validation of values in web form input to ensure that they will be accepted before they are submitted to the server.

JavaScript is interpreted, loosely-typed and when run at the client side it may be hosted in varying environments, applications, implementations and versions. This means the programmer has to take extra care to make sure the code executes as expected in as broad a range of environments as possible.

JavaScript has had its fair share of security issues due to the fact that it is a language running arbitrary server-provided code on a client computer.

3.4 ASP

Active Server Pages.
This is a server-side scripting facility that Microsoft supplies with all its web servers, which enables the creation of truly dynamic web pages.
An ASP page is an HTML page that includes scripts, which are processed on the web server before the page is sent to a user.
The scripting language of choice for ASP is VBScript.
Active Server Pages is not a programming language. It is more accurately referred to as a type of programming platform where programmes reside. Any language that the server understands can be used to build the Active Server Pages. The most popular language used for ASPs is VBScript.

ASP works as follows:
When you request an ASP page through your browser the web server that the ASP page is sitting on goes through the page looking for ASP code and executes it. After the ASP code has run it is stripped out and a pure HTML page is sent to the browser.

Advantages of ASP includes:

- ASP/VBScript is an easy language to learn.
- Debugging is easy in ASP as programmers will be able to see the result of the change immediately instead of waiting for compilation to finish.
- ASP components offer several benefits to programmers such as: speed, security, modularity and extensibility. ASP components execute much faster than standard interpreted ASP codes since they are compiled.
- Compiled ASP components are stored in binary DLL format which hides their source code, compared to standard text ASP codes.
- ASP provides database connectivity with many RDBMS (Relational Database Management Systems) such as MicroSoft SQL Server as long as the ODBC (Open Database Connectivity) driver for thr RDBMS is available. This facility is required to support data-driven web applications which is one of the features that makes ASP so functional and appealing in the first instance.

A disadvantage of ASP is that it runs mostly under Windows platforms which means it is not really a good choice if cross-platform compatibility is required.

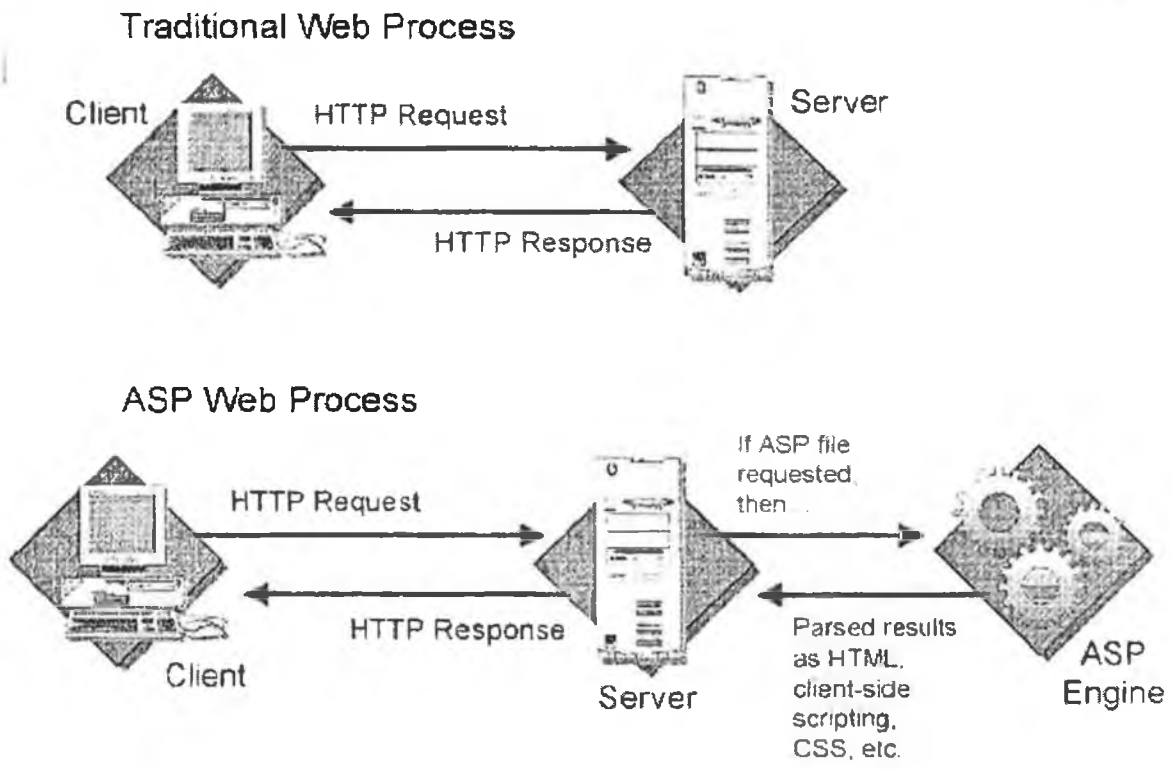


Fig. 02 *Diagramatic Representation of how ASP works*

3.5 SQL

Structured Query Language.

SQL database is the most widely used in the present computing environment. Data is stored in a very structured format that provides high levels of functinality. SQL databases are more robust, secure and have better performance than older database technologies.

SQL is used to create, maintain and query relatinal databases. It uses regular English words for many of its commands, which makes it easy to use and understand. SQL is ofetn embedded within other programming languages. The main difference between SQL and standard programming languages is that SQL is declarative. That means you specify the type of data required from the database and the BDBMS is responsible for retreiving it.

A table is the basic structure of the relational model of an SQL database, consisting of rows and columns. Data definition includes declaring the name of each table to be included in the database, the names and types of all columns for each table, constrains on the values in columns, and the granting of priviliges for table manipulation to prospective users.

SQL data manipulation operations are invoked through a general query specification. The language includes all arithmetic operations, predicates for comparison and string matching, universal and existential quantifiers, summary operations for maximum/minimum or count/sum. Transaction management is achieved through COMMIT and ROLLBACK statements.

Access control is provided by GRANT and REVOKE statements. Each user must be granted the privilege to access a specific table or view using a specific statement.

Referential integrity, CHECK constraint and DEFAULT clauses is offered through the SQL Integrity Enhancement facility. Referential integrity allows specification of primary and foreign keys with the requirement that no foreign key row may be inserted or updated unless a matching primary key row exists. Check clauses allow specification of inter-column constraints to be maintained. Default clauses provide default optional values for missing data.

Advantages of SQL databases include:

- High Speed. SQL queries retreive large amounts of records from a database speedily. SQL databases are also more efficient at handling large volumes of data and processing it quickly compared to non-SQL databases.
- Security. With the SQL database all the data is stored in one place.
- Standards. SQL databases use the SQL standard, which was adopted by ANSI in 1986 and continually evolved since. Non-SQL databases to not have clear standards to adhere to.
- Compatability. As a result of standards portability from one SQL database to another does not present a problem. Also, SQL databases conforming to established standards can easily be accessed by third party software and

application tools. This facilitates the development of other applications and solutions around SQL databases.

- Coding not required. Using standard SQL it is easier to move applications between different database systems without the need to rewrite a substantial amount of code.

Disadvantages of SQL databases include:

- In SQL it is necessary to define the tables and fields and the relatinships between them before anything else can be done.
- SQL databases are good for mature applications but not as good for developing new applications. Changes to tables and fields in an evolving application can cause compatability issues for previous versions.
- Interfacing to an SQL database is more difficult since SQL supports many advanced features.

The difference between advantages and disadvantages of SQL is continually reducing as developers make SQL as user-friendly as possible.

3.6 Open Database Connectivity (ODBC)

ODBC provides a standard software API method for using database management systems (DBMS). The objective of ODBC is to make it independent of programming languages, database systems, and operating systems.

ODBC specifications offer a procedural API for using SQL queries to access data. An implementation of ODBC contains one or more applications, a core ODBC library, and one or more database drivers. The core library is independent of the applications and DBMS systems and acts as an 'interpreter' between the applications and the database drivers. The database drivers contain the DBMS specific details.

The advantages are that a programmer can write applications that use standard types and features without concern for the specifics of each DBMS that the application may encounter. Also, database driver implementors need only to know how to attach to the core library.

ODBC operates with a variety of operating systems and drivers for non-relational data such as speradsheets, XML and text files. Because ODBC is more than ten years old it offers connectivity to a wide variety of data sources as more drivers exist for ODBC than for newer APIs.

Disadvantages of ODBC include:

- Administering a large number of machines may involve a diversity of drivers and DLLs.
- The layered architecture of ODBC can introduce a slight performance penalty.
- Driver architecture may also affect performance.

3.7 ActiveX Data Objects (ADO)

ASP uses ActiveX data object (ADO) connection to access data in a database, These ADO connections are written in server-side script inside the ASP pages. ADO is a MicroSoft technology that is installed with MiscroSoft IIS. ADO is a collection of objects that allow applications to communicate with data sources in a consistent manner.

Accessing a database from an ASP page

Folowing are the steps to access a database from inside an ASP page:

1. Create an ADO connection to the database
2. Open the database connection
3. Create an ADO recordset
4. Open the recordset
5. Extract the data required from the recordset
6. Close the recordset
7. Close the connection

This is an example of an ADO to connect to an ACCESS database called MODULEARCHIVE. The ADO connection points to an ODBC connection that would have been set-up on the server.

```
<%
        SetCON = Server.CreateObject("ADODB.Connection")
                CON.Open "DSN = MODULEARCHIVE;"
%>
```

3.8 DSN

Data Source Name.

Before a connection can be made to any database using Active Server Pages (ASP), it is necessary to set up a DSN.

According to Microsoft documentation DSN means "The name that applications use to request a connection to an ODBC data source."

The DSN represents the ODBC connection. It hides the connection details such as database name, directory, database driver, UserID, password, etc.

When making a connection to the ODBC it is not necessary to remember the database name, where it resides etc.

3.9 VMWARE

VMware Workstation software consists of a virtual-machine suite for Intel x86-compatible computers. This software suite allows users to set up multiple x86 virtual computers and to use one or more of these virtual machines simultaneously. Each virtual machine instance can execute its own guest operating system, such as (but not limited to) Windows, Linux, and BSD variants. In simple terms, VMware Workstation allows one physical machine to run numerous operating systems simultaneously. Other VMware products help manage or migrate VMware virtual machines across multiple host machines.

3.10 ASP Upload

ASP code does not natively support an upload feature (it does not have a built in file upload feature). It can be coded, but it is slow and it doesn't handle large file sizes very well. Instead we use aspupload.dll from www.aspupload.com to handle the file uploads.

Register the dll on the system and this will allow you to call its functions to upload files. Files up to 2GB are allowed by calling the persists.upload function within the code, e.g.:

Set upload = server.createobject ("Persists.Upload").

Chapter 4. DATABASE DESIGN

This Chapter looks at the analysis of requirements for the Intranet and how this analysis will influence the database design. The chapter also looks at the importance of databases and how a database can act as the backbone to any intranet application. The chapter looks at Microsoft Access and why it was used as the database management system for this project.

4.1 Analysis of Requirements

The real objective of this intranet is to build a learning organisation, which will be used by everybody. According to Stone Gonzalez (1998), "The right metric for intranet success is consistent use over time." The scope of this project is to setup a functional, user-friendly intranet that will meet the key requirements of both staff and students at the college. Probably since the introduction of email, staff at this campus have been communicating electronically with their class groups. They have also been proactive in sharing information with students and colleagues on the network shared drive. This might well be regarded as an intranet of sorts, but naturally it lacks all of the attributes that an intranet possesses.

As the number of students on campus continues to grow and as more and more lecturers are travelling long distances to deliver their lectures the development of an intranet is the first step to embrace 21st century technology in the communication of information.

A number of meetings were held with an Information Technology Focus Group consisting of three staff members, two student representatives, the IT technician, the school administrator and the Head of Centre. The purpose of this group was to develop an IT policy for the campus and the development of an intranet became a major component of this policy. Following are the requirements identified by this group in relation to the intranet.

- A simple web page to be designed that would be set as the home page on all PC's for both staff and students.

- One staff member from the focus group would become the 'super-administrator' for the intranet. The Head of Centre and school administrator would also have super-administrator rights.

- All lecturers would have administration rights which would be at a lower level than the super-administrator.

- The first phase of development would be to get the infrastructure and design in place for the intranet to function as per the requirements of this group. This is essentially the author's project.

- The second phase is to target one programme on campus and populate the tables with all the information required.

- The third phase is to train the remainder of staff on their role as administrator and get everybody uploading information onto the site. This phase would also involve iteration of the design following use and analysis by staff.

- The fourth phase is to roll the intranet to all PC's with information fully loaded for use by students. Steps two to four are outside the scope of this project. The specific requirements of phase one will now be considered.

    1. The super-administrator (SA) will have rights to edit and change everything. Logon authentication will have to be verified against details held on the database. The type of rights will include:
       Add a new course and specify what years it will be offered on. For instance a level 7 course will be offered over three years and a level 8 course will be offered over four years.
       The SA will be able to change details to have a course offered over four years instead of say three years.
       The SA will be able to add modules and link modules to specific lecturers. It will also be possible to link modules to particular years of a course. This is highly desirable in a modularised environment as it will allow a module in year two of one programme to be made available to say year one or year three of a

different programme.
The SA will be able to add specific sets of notes to a module. This will also be allowed by all lecturers who will have administrator status.
The SA will be able to change the module name and the lecturer associated with the module, as well as being able to add and remove all notes, modules and courses.

The SA will be the only person with rights to the lecturer account maintenance. This will include the ability to:
reset a password or
remove / add a lecturer to the table and / or
change the admin level of the lecturer from administrator to super-administrator and vice-versa.

2. All lecturers will be classified as administrators and will have limited rights compared to the super-administrator. These will include:
   The ability to upload notes to a module linked to a lecturer. It will be possible and allowed to have more than one lecturer linked to a module and in that case they will have equal rights.
   Lecturers will be able to decide if the notes once uploaded are displayed on the live site. This will allow lecturers have all their notes for a particular module uploaded on the intranet and only make them available to students either before or after the lecture as they decide.
   Lecturers will also be able to delete notes associated with a module.

3. A search facility should be included on the intranet.

4. The Home page should include a link to projects as there are always a number of interesting projects in progress on campus. In 2006 we had a four week exhibition at the Farmleigh estate and currently we have a smaller exhibition running in Tipperary until the end of June.
   A significant section of the homepage should be devoted to a latest news section, which might include information on what's happening today or if a

lecturer is not present to take classes etc.

5. Students will have no administration rights but will be able to download copies of all notes from the site and/or print them without the need to download.

The next section will deal with the database design and demonstrate how the design will meet the requirements identified in the first section of this chapter.

4.2 Databases

A database is simply a 'base for data storage'. Databases are very important part of today's business world. Without them it would be almost impossible to run a business effectively. A database can be used to store all kinds of data relating to an organisations employees, customers, orders, records etc. A database helps to eliminate the need for tonnes of paperwork and makes for easy storage and retrieval of information.

Tables are the foundation of any database, they store the data within the database and provide the structure by which it is organised. Each table stores a set of related data. Tables are made up of 'Records', which contain information about the object, or person etc. The records are displayed in rows where each category of information in a record is known as a 'Field'. Fields are displayed in columns, the field name appears in the database table as a column heading.

4.3 Microsoft Access 2000

Access 2000 is a very powerful Database Management System. Access 2000 makes it easy to retrieve the information required and provides tools that help to organize and share data. Access 2000 also enables Web calibration and aids in building faster and more effective business solutions.

Microsoft Access documents are called databases. An Access database is a collection of database objects: tables, queries, forms, reports, macros, and modules. New objects can be designed or existing ones opened to work with the database. Unlike many database programs, an Access database can contain all of the objects that make up a database application in a single file with the .mdb file extension. For

this reason, an Access database file is sometimes called a database container. An Access database application can call procedures from a library database after establishing a link to the database called a reference. You can create or purchase tools called add-ins to add custom features to Access. Access user-level security account information is stored in a database called a workgroup information file.

In Microsoft Access, data is stored in tables. Tables organize data into columns and rows. Each row in a table is called a record. Each column in a record is called a field. For example, as in this project, each record in the Modules table contains information about one module. Each module record contains fields named for each piece of information you want to store about the module, such as module name and module id.

One can create a separate table for each topic of data, such as Courses information, Admin and Course stages. Using a separate table for each topic means that you store that data only once, which makes your database more efficient and reduces data entry errors. If one keeps separate tables for course information and course stages, each course has only one record in the courses table, so if you need to update or correct information about a course, you need only change the data in one record. This method reduces redundant entries and reduces data entry errors. The process of eliminating redundancy by dividing data into separate related tables is called normalization. Access provides a wizard called the table analyser Wizard to help normalize the database.

4.4 Development of the system Database

From the start it was decided to use Microsoft Access as the Database Management System. This was due to its compatibility with the other technologies being used in the system development, and because it provides users with one of the simplest and most flexible DBMS solutions on the market today.

4.4.1 Design of the Database:

The first stage in designing the database was to decide what information would be needed to run behind the scenes of an intranet for this campus. This information would then have to put into tables that categorised each piece of information.

In deciding what table's and information would be needed it had to be decided what features the finished system would have as identified from the analysis of requirements. The design of the overall system will be discussed in the next chapter. For now though we will just look at the tables used in the database.
The most obvious table required would be a table with the list of modules on offer across all stages, from here a table with the module notes contained in the modules would be required. Tables would also be required to hold lecturers delivering each module (ModuleLecturer), Courses, Course schedules (Courseov), Admin etc.

The Final tables to be used in the project were designed as follows:

Admin

Lecturer_ID, Name, Password, SuperAdmin, Email

CourseModules

Entry_No, Module_ID, Course_ID, Stage_ID

CourseOV

Course_ID, Course_Ov_Link, Displayov

Courses

Course_ID, Course_Name

CourseStages

Course_ID, Stage_No

ModuleLecturer

Module_ID, Lecturer_ID

ModuleNotes

Entry_No, Module_ID, Title, DocLink, Display_Doc_Link

Modules

Module_Name, Module_ID

4.5 Creating the Database

To create the database called Iffc the following steps were taken (Iffc is an acronym for Letterfrack Furniture College):

Step 1: Launch MS Access 2000 and you will see this screen.

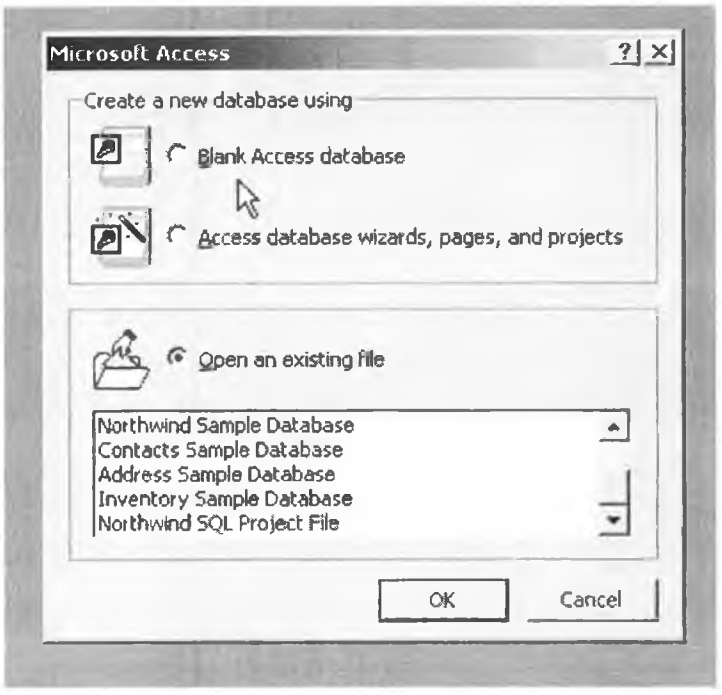


Fig. 03 *MS Access launch screen*

Step 2: Click on "Blank Access database" and hit "ok" button.

Step 3: Choose a name for the database, "LFFC.mdb" and Click OK.

Step 4: Create your Tables. "Click on the Create Table in Design View"

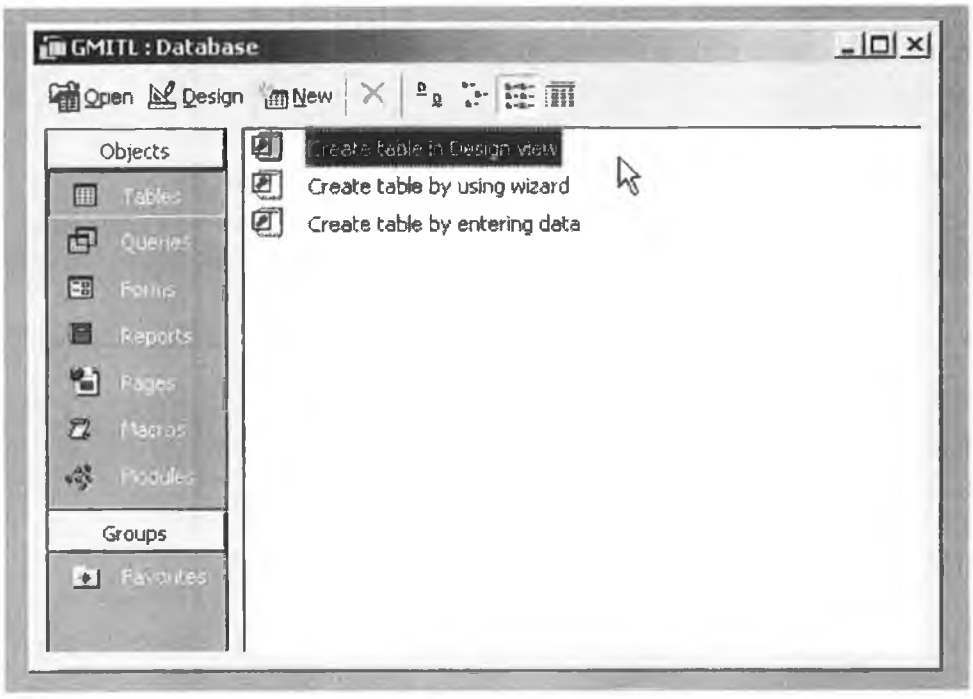


Fig. 04 *MS Access Design View*

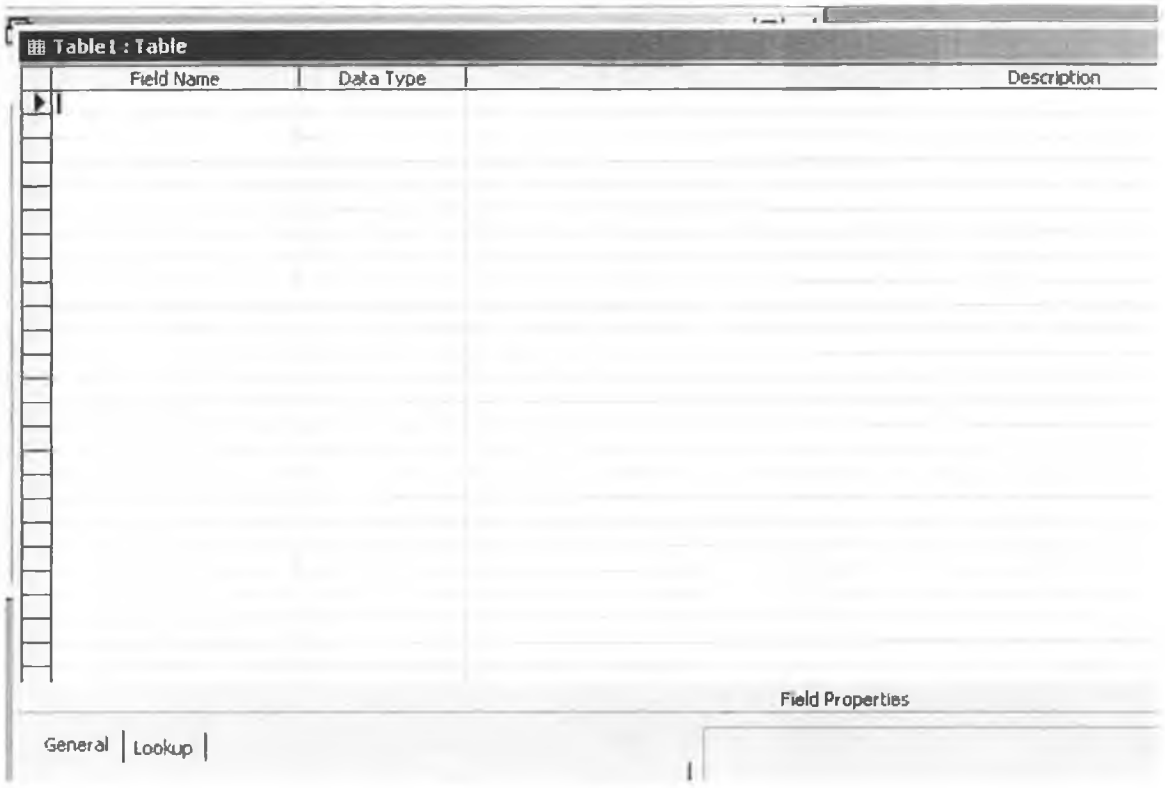


Fig. 05 *MS Access add data*

Step 5: The Design view of the tables opens up, from here you fill in the fields the table will require and their data types e.g. Text, Number, Currency.
Fig (6) Shows the Admin table in Design View. This Table contains the fields Lecturer_ID, Name, Password, SuperAdmin, Email, which can be seen, in the diagram below. The Lecturer_ID field is set as the primary key of this table. The primary key gives the primary assignment to a particular field for relationships within tables i.e. it is a way of relating tables together.
The primary key must contain a unique value and it cannot be left null.

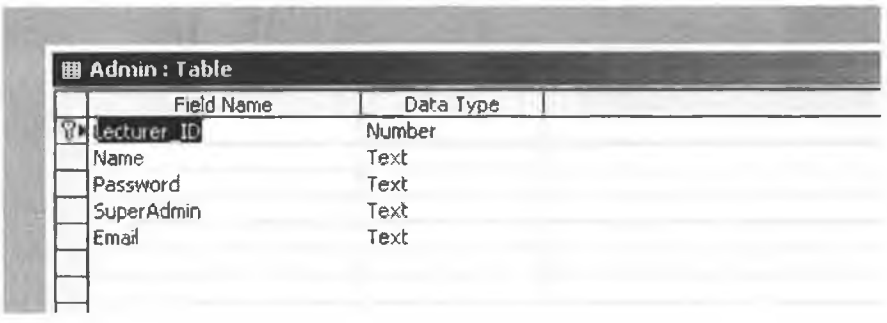
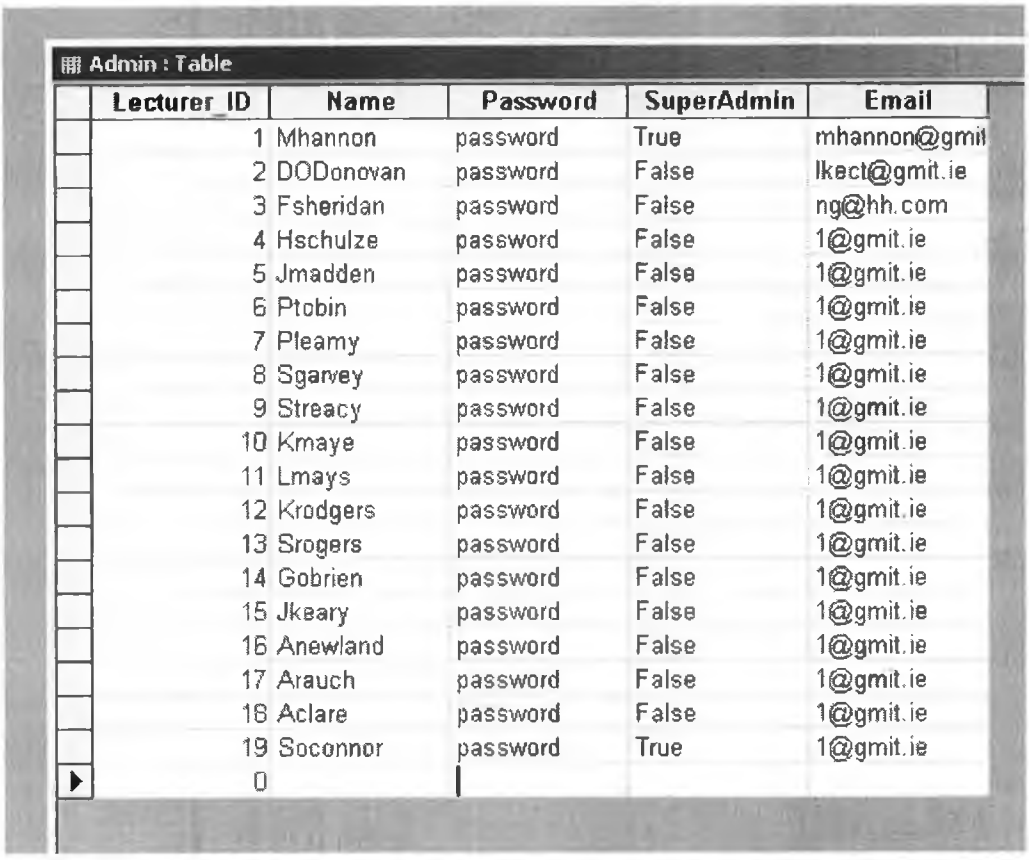


Fig. 06 *Admin table in design view*

You must repeat Step 5 until you have created all the tables required for the database.

The next Step is to fill in the information into each table of the database.
To do this double click on the table and a screen like Fig. (7) is shown, from here you simply fill in the information.



Admin : Table

Lecturer_ID	Name	Password	SuperAdmin	Email
1	Mhannon	password	True	mhannon@gmit
2	DODonovan	password	False	lkect@gmit.ie
3	Fsheridan	password	False	ng@hh.com
4	Hschulze	password	False	1@gmit.ie
5	Jmadden	password	False	1@gmit.ie
6	Ptobin	password	False	1@gmit.ie
7	Pfeamy	password	False	1@gmit.ie
8	Sgarvey	password	False	1@gmit.ie
9	Streacy	password	False	1@gmit.ie
10	Kmaye	password	False	1@gmit.ie
11	Lmays	password	False	1@gmit.ie
12	Krodgers	password	False	1@gmit.ie
13	Srogers	password	False	1@gmit.ie
14	Gobrien	password	False	1@gmit.ie
15	Jkeary	password	False	1@gmit.ie
16	Anewland	password	False	1@gmit.ie
17	Arauch	password	False	1@gmit.ie
18	Aclare	password	False	1@gmit.ie
19	Soconnor	password	True	1@gmit.ie
0				

Fig. 07 *Admin table populated*

4.6 Relationship of Tables

To bring data from multiple tables together for a query, form, or report, one defines relationships between tables based on a common piece of information stored in the tables. Fig. (8) shows tables linked together in the database.

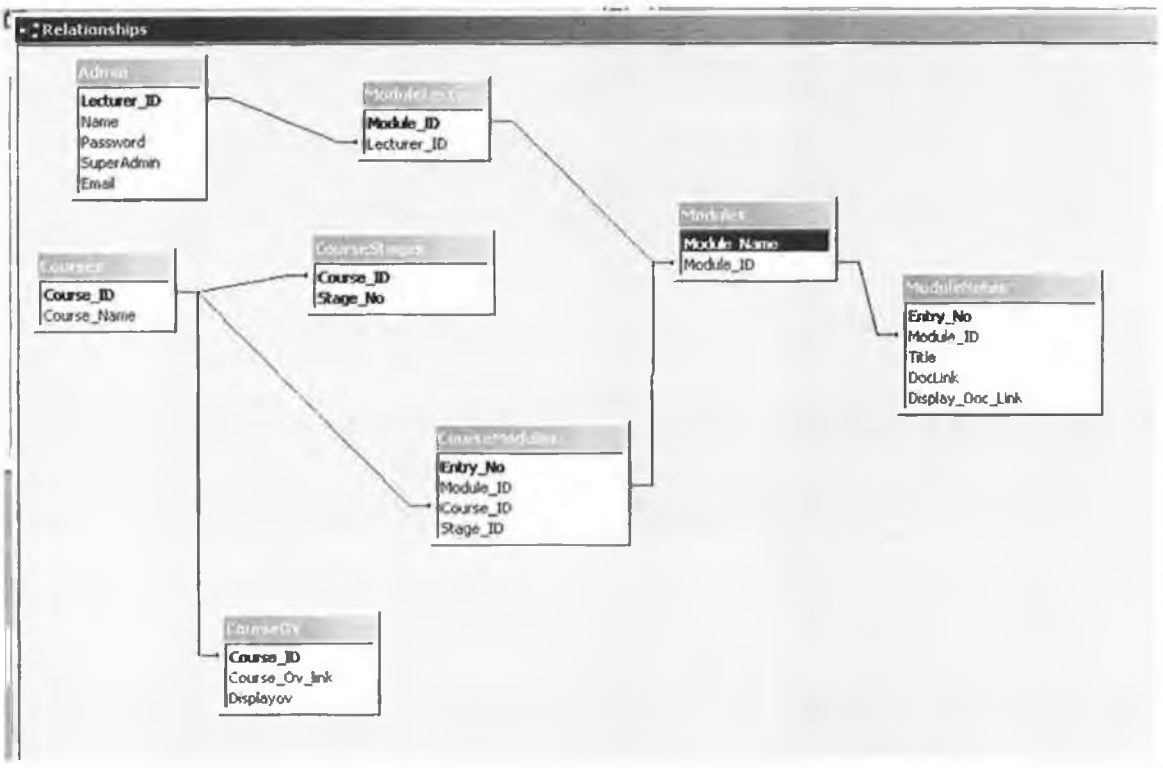


Fig. 08 *Relationships between the tables*

As we can see from the screen shot above the relationship from the Courses table to CourseStages table is a one to many relationship. For each course listed in the Courses table it will have a specific number of stages, a predetermined list of modules and a specific Course overview or Approved Course Schedule.

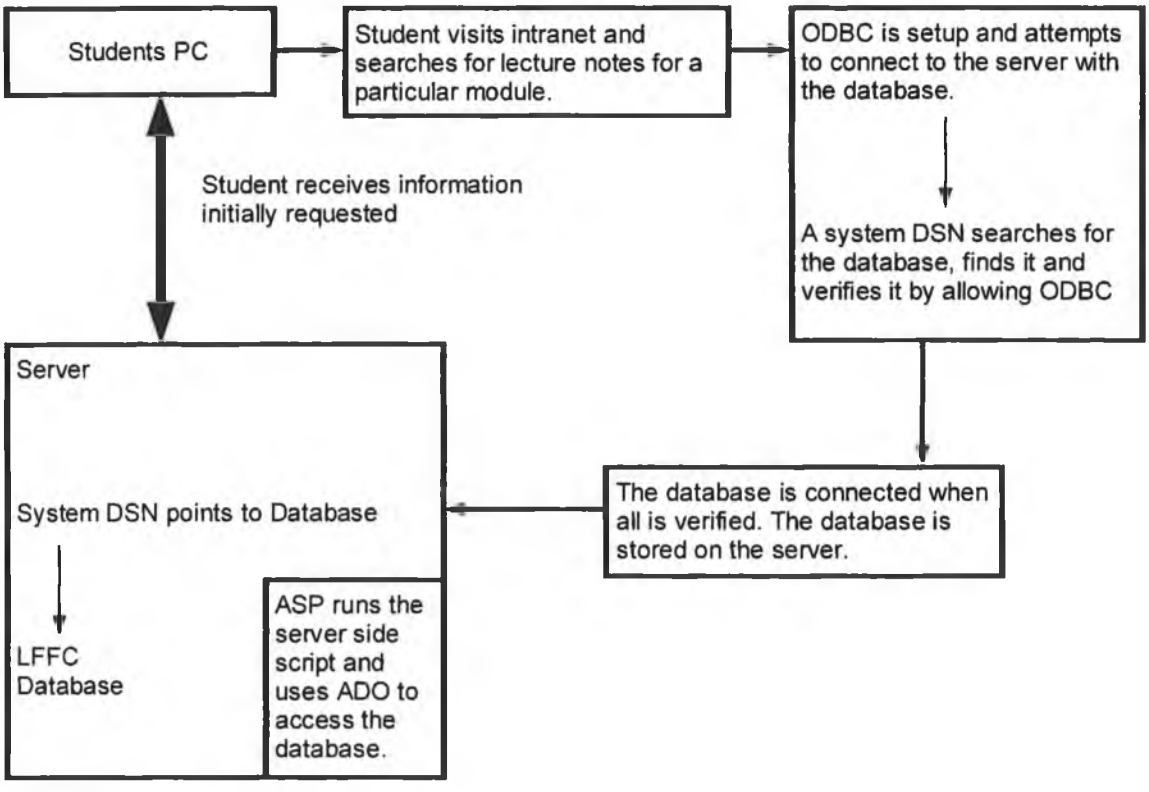


Fig. 09 *System overview*

From the outset it was clear that the system was to consist of a database which would be connected to a website. Having explained how to build the database in the last chapter, this chapter will show how to link the database to the website, and then detail the development of the intranet website and the administration website used in the system. Fig (9) The system overview diagram shows an overall picture of how the system will function.

Figs (10) and (11)
Show a detailed design of how the web interface and the Administration section are designed and how the database acts as the backbone to the overall system. These diagrams show how each table in the database is related to a Web page and the function that the page performs in the system.

Database

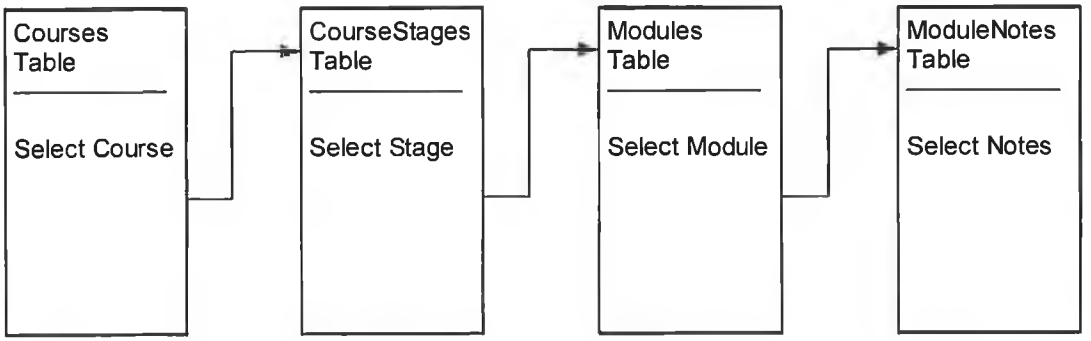


Fig. 10 *Database overview*

The Administration Section

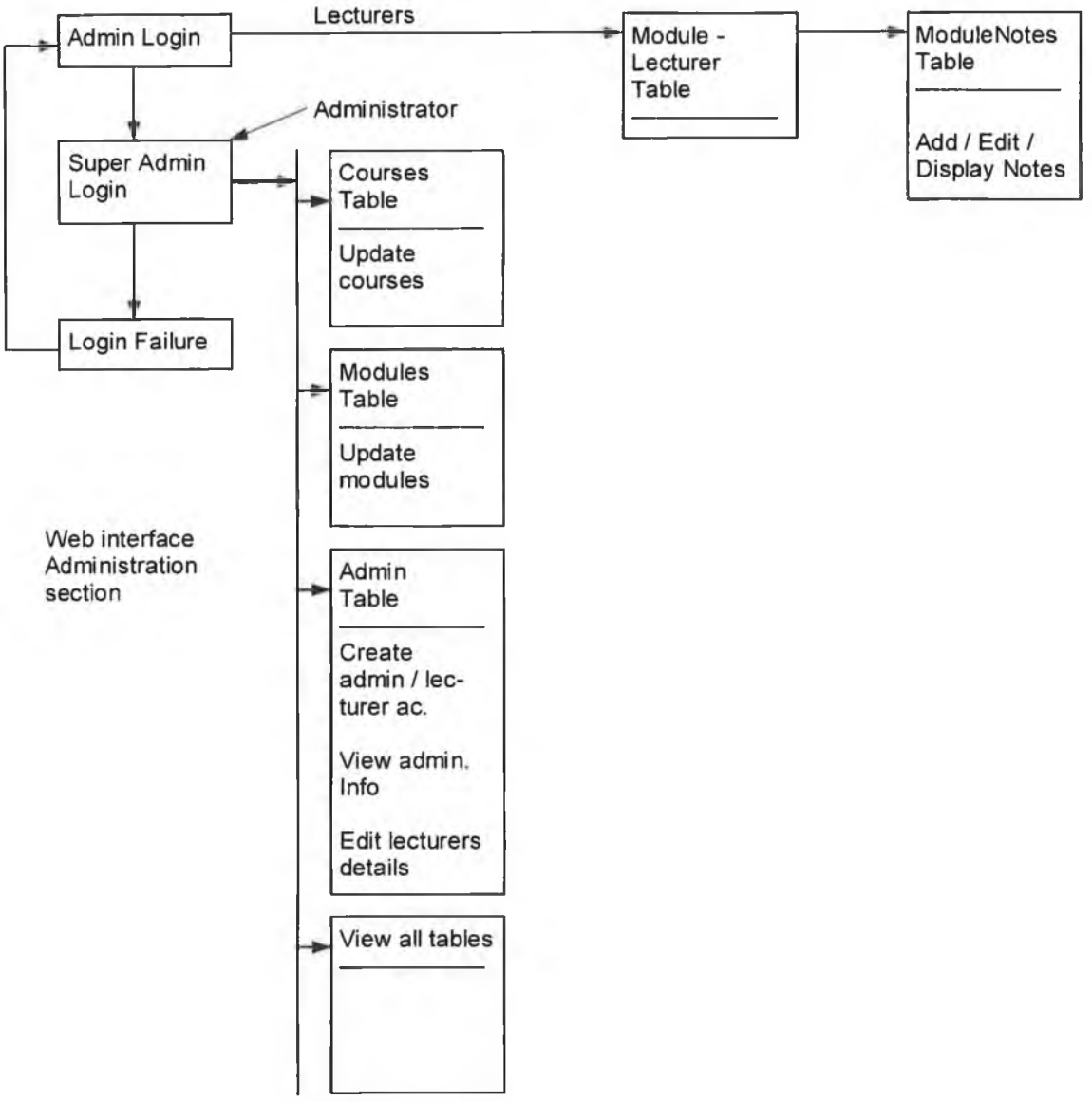


Fig. 11 *The administration section overview*

5.1 Linking The System

Creating an ODBC Connection and a System Data Source Name (DSN) to allow the database to act as the backbone of the whole system we must first create an ODBC connection between the web pages and the Database.

Before a database can be connected using Active Server Pages (ASP), it is required to set up a System Data Source Name (DSN). A DSN basically sets up a name for the database that is being used; in this case the DSN name given is "LFFC". Then when the ODBC connection is made the DSN name is required to choose the correct Database.

To Create a DSN

Step1

Go to start, Settings, Control Panel and double click on Administrative tools.

Step2

Double click on the Data Sources (ODBC) icon.

Choose the system DSN tab along the top of the window. See fig(12)

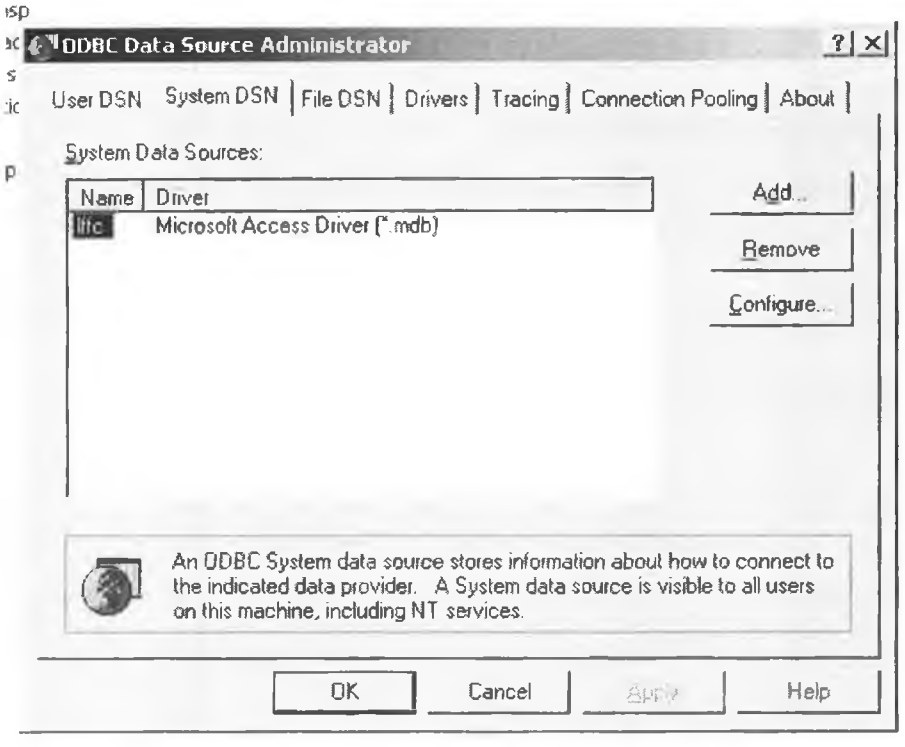


Fig. 12 *ODBC Data Source Administrator dialogue box*

This shows which DSNs exist on the machine.

Step 3

To create a new one click the 'Add..' button, which will open the Create New Data Source Dialog box.

Step 4

Select the Microsoft Access Driver (*.mdb) from the list and hit the Finish button.

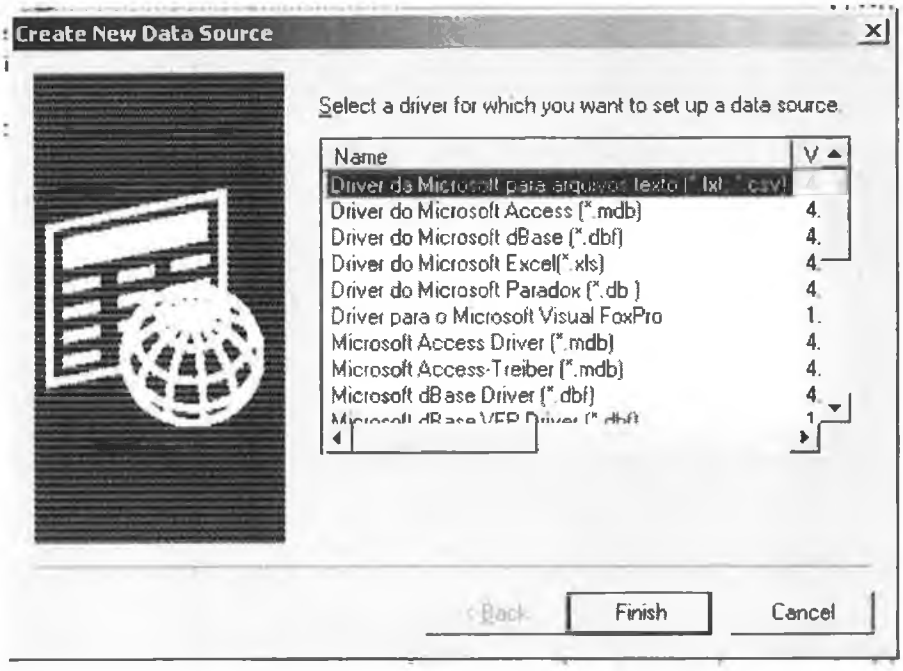


Fig. 13 *Create New Data Source dialogue box*

Step 5

Another Dialog Window appears asking you the name you want to call the Database, in our case "LFFC". Next click the 'Select..' button, this brings up the standard windows file open dialog, which will allow you to find the location of your database. Select your database and then click 'OK'. Which will return you to this screen.

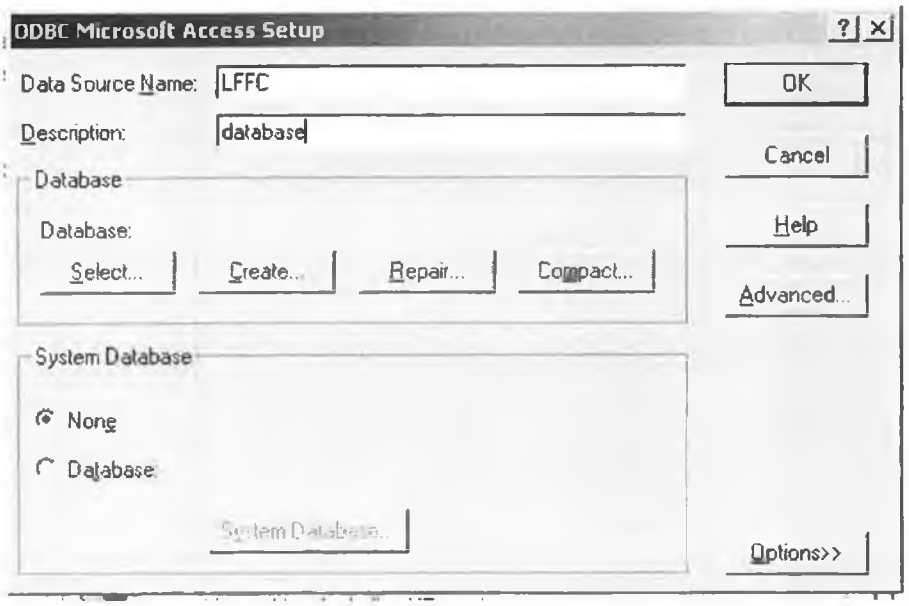


Fig. 14 *ODBC MS Setup dialogue box*

Step 6

Click the 'OK..' button in the window above. This will get you back to the system DSN window. You will now see LFFC added to the System DSN list. The database can now be accessed through the system DSN list.

How this is achieved will be explained in detail later when dealing with the ASP code for the system.

5.2 Building the System

ASP is going to perform the core functions of the system. In this section the features of each part of the system are explained, in particular how they function using ASP. Following are some explanations of features of ASP that are used within the system:

Server-side Includes

Programmers always look for ways to be efficient. Rewriting the same code time and time again is time-consuming. ASP provides a way to eliminate this: Server-side Includes (SSI). Server-side Includes are directives executed on the server that include the content of one file in another, in a specific place, before sending it to the client's browser.

Includes can play a major role in a Web site. They can be used for headers, footers, or navigation tools (see fig.15), allowing the same code or content to be reused for things like copyright information, your company contact information, and logo etc. in several Web pages.

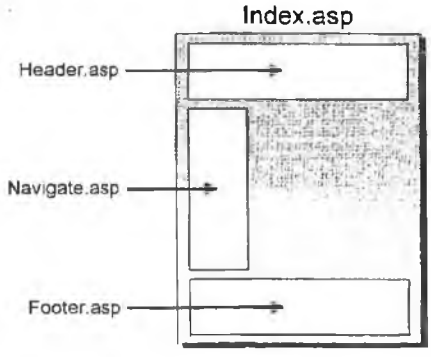


Fig. 15 *Example of includes file*

You make a subdirectory of your site to hold all your include files, and call which ever one you require for a page using a similar line of code to this:
<!--#include virtual="/Iffc/Header.html"-->
In this project Server-side includes were used to create headers in both the intranet site and the administration site, and to display the dynamically generated navigation bars that appear throughout the web site.

Applications and Sessions.

Applications
If you think of a single ASP page as a single routine or procedure. It can do things on its own, but it can also be tied together with other ASP pages to create an ASP application. But an ASP application is more than just any ad hoc group of ASP pages. An ASP application includes all files in its root directory and any of its subdirectories. An application starts the first time any user opens one of the web pages in the application, and ends only when the server shuts down. Each application has a single global.asa file, which is used to store application and session level variables and objects, as well as launch application and session level events.

ASP Session

An ASP session allows the administrator to treat a user's visit to the web page as one continuous action, rather than as a series of separate server requests.
An ASP session is started the first time a user visits a Web page. Unlike an ASP application (which is the same for all users), each user is assigned his or her own session ID. A session ends when it times out or is explicitly abandoned.

Session level variables are used in the system to tell which user is logged in so they will be able to use their own module notes. A Session level variable is also used for authentication purposes in the administration site, this will be discussed later.

ADO

ADO or Active X Data Objects are built in components of ASP. These are COM objects that use ODBC and/or OLEDB to access databases and other stores of information, so they are language independent. They can be used not only from Active Server Pages but also from Visual Basic, Visual C++ and other programming environments. ADOs enable you to connect to a database to
Insert: add new data to a table.
Update: change data already in a table.
Delete: remove data from a table.
Select: look at data in a table.

ADO Constants

ADO uses several declared constants to describe various options. When writing ASP code, you can either use the cryptic numbers that these constants represent, or you can include the constants in your ASP pages. Microsoft has provided a single, standard file called adovbs.inc, which includes all the ADO constants and their values. To use it in your ASP pages, simply include it as you would any other file:
<!-- #include virtual="/includes/adovbs.inc" -->

In a traditional Windows set up, the file is found at C:\Program Files\Common Files\system\ado\adovbs.inc. Using these constants greatly improves the readability of the ASP code.
This File is included at the beginning of pages in the project where ADO constants where required.

5.2.1 Creating a Connection to the database

Before you can use ASP to generate a database driven web page or select information from a database over the web you must connect to the database. To do this you create a Connection Object.

```
<%@ LANGUAGE="VBSCRIPT" %>
<%
set Mycon = Server.CreateObject("ADODB.Connection")
Mycon.Open "LFFC"
(ASP Code for web page goes here)
Mycon.Close
set Mycon = Nothing
%>
```

Firstly you create the connection object, and then you open it and use the DSN to select the correct Database. When you are finished you must close the connection and to free up the resources on the server you must destroy any variables or references to it, you do this by setting it equal to Nothing.
This connection is used extensively in the project, as all the core functions of the project require a connection to the database backbone.

Now that we have discussed some features that will be used through out the system, lets look at the development of the system. First lets look at the intranet site.

5.3 The Intranet Site

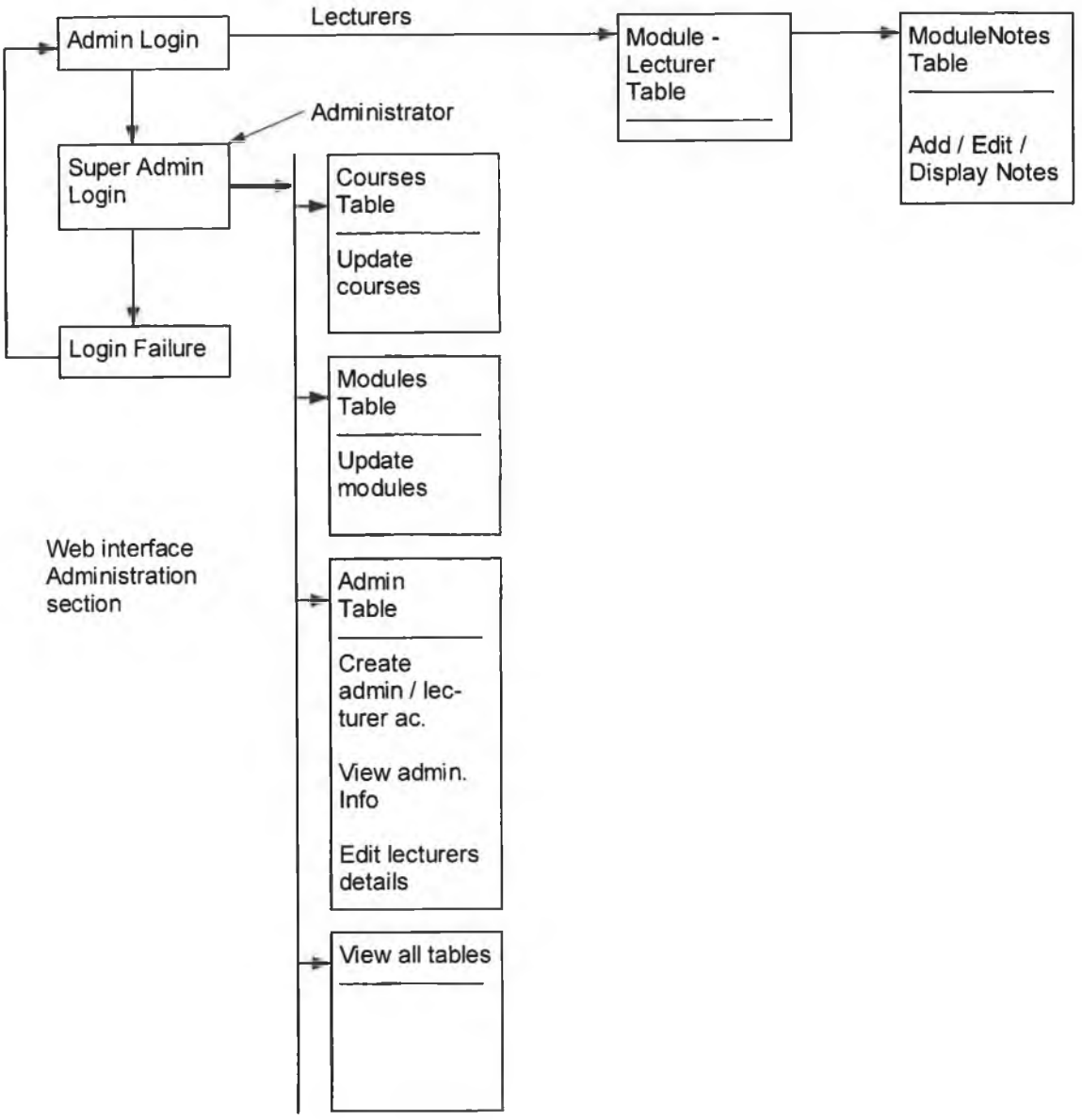


Fig. 16 *Design of the intranet site*

In the student site ASP is going to be used to:

(1) Dynamically generate the links to each course. By doing this if a new course is added to the site it will automatically be included in the site.

(2) Display the modules associated with whichever course the user wishes to visit.

(3) Search for specific module notes.

(4) Display the module notes requested by the user.

Due to the amount of pages the system contains I will just go through a selection of them briefly detailing what they do, with some code examples.
At the end of this chapter I will take one ASP page and go through the code in-depth.
The full code for each web page in the site is available in Appendix One

Main.asp

This is the homepage of the site. It contains the welcome note. The only asp it contains is the header is contained in the includes file as a server side includes file.

```
<%@ LANGUAGE="VBSCRIPT" %>
<!--#include virtual="lffc/Header.html"-->
```

Courselinks .asp

This page dynamically generates URL links to all the courses from the courses table in the database. This code generates the links to each course in the database.

```
<%
     set Mycon = Server.CreateObject("ADODB.Connection")
    Mycon.Open "lffc"
%>


<%
   set catSet = Mycon.Execute("select * from Courses")
    while not catSet.EOF
    %>


    <A HREF="course_stages.asp?p_course_id=<%
       Response.Write catset("Course_ID") %>&p_course_Name=<%
       Response.Write Server.URLencode(catset("Course_Name"))%>"><font size="3">
     <%= catSet("Course_Name") %></a>
<BR>
    <%
        catSet.MoveNext
    wend
    catSet.Close
    set catSet = Nothing

    Mycon.Close
   set Mycon = Nothing
%>
```

The first thing it does is open the connection to the database. We then select everything from the courses table and loop through each record displaying it as a Hyperlink. We have fed the link parameters p_course_id and p_course_name, We are going to pass these parameters as part of the URL so as the next page can

dynamically generate a list of stages associated with the course. To ensure we don't run into problems if we have a course name with a space in it, we are using Server.URLEncode to prevent this.

The pages coursestages.asp and coursemodules.asp use the same logic to dynamically generate the lists displayed.

Search.asp

This is the search page used for the student search for module notes. It is just a standard HTML form, which will pass in the value to search_action.asp, which will perform the actual search.
It is also saved as an ASP page as it contains Server-Side Includes for the header, as with all the pages in the site.

Search_action.asp

This page reads in the value entered in the form, it then searches the 'title' field in the ModuleNotes table for a word, which is similar to the word entered, and the moduleid field for the Moduleid of the module selected. It then displays the details of all the notes it has found and you are given an option to view the notes.

```
<%
    p_text = cstr(Request.form("p_text"))
    p_text = uCase(p_text)
    p_mod_id = Request.form("p_mod_id")

    set Mycon = Server.CreateObject("ADODB.Connection")
    Mycon.Open "lffc"
    %>


    sqlText = "SELECT * FROM ModuleNotes WHERE Module_ID = " & p_mod_id &
" and (uCase(Title) like '%"&p_text&"%')"


  set reviewSet = Mycon.Execute(sqlText)
  while not reviewSet.EOF

%>
```

```
<h5><A href="/Iffc/ModuleNotes/<%Response.write
Server.URLencode(reviewSet("Doclink"))%>"><%Response.write
reviewSet("Title")%></A></h5>
<%
    reviewSet.MoveNext
        wend
        reviewSet.Close
        set reviewSet = Nothing

      Mycon.Close
          set Mycon = Nothing
```

The above code just shows you how it searches the ModuleNotes table. It firstly gets the value p_text from the search form and then it selects from the ModuleNotes table where the title is like the searched word, the rest of this page i.e. the part that displays the results can again be found in Appendix One.

5.4 The Administration Section

This section is for controlling access to the site and for managing and uploading content. There are two administration levels catered for, the Super Administrator who has full access and editorial rights and the Administrator who only has access to modules and courses he/she lectures on.

The ordinary administrator can only add, edit and display notes linked to modules that he/she has rights to. These rights are controlled by the Super Administrator (SA). The SA can add new lecturers, delete existing ones, change the level of administration to super or ordinary, add courses, delete courses, add/display modules, add/display/edit notes and so on.

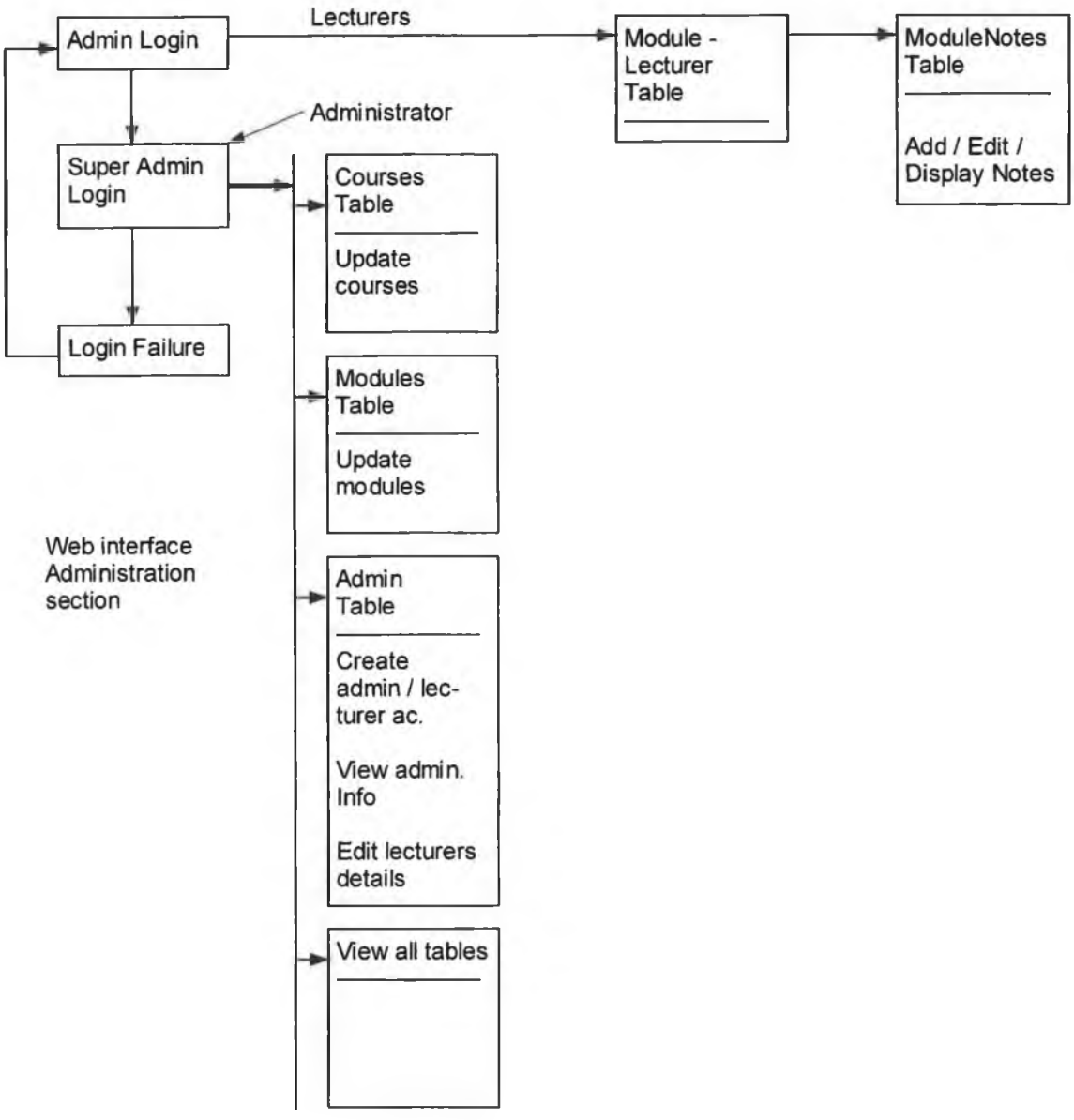


Fig. 17 *Overview of administration section*

ASP is going to be used to perform the following functions in the administration section.

(1) Login authentication

(2) Security

(3) Update, Add, Remove Courses from the Database

(4) Add, Remove Update modules from the database

(5) Add notes for each module to the site

(6) View the Tables of the Database

(7) View administrator details

(8) Add Remove Edit administrator information

Detailed explanation of an ASP page from the system.

We will look at the following page and step through it line by line.

This code is from sa_update_lect_action.asp.

```
1.  <%@ LANGUAGE="VBSCRIPT" %>
2.  <%
3.  if not Session("Auth")="SA" then
4.      Response.Redirect"admin.asp"
5.  end if
6.
7.  Const adOpenKeyset = 1
8.  Const adLockOptimistic = 3
9.  Const adCmdText = &H0001
10.
11.     p_lect_id = Request.Form("p_lect_id")
12.
13.     set Mycon = Server.CreateObject("ADODB.Connection")
14.     Mycon.Open "lffc"
15.
16.     mySQL = "select * from Admin where Lecturer_ID= " & p_lect_Id
17.
18.     set myRS = Server.CreateObject("ADODB.RecordSet")
19.     myRS.Open mySQL, Mycon, adOpenKeyset, adLockOptimistic, adCmdText
20.     myRS("Name") = Request.form("p_name")
21.     myRS("Password") = Request.form("p_pass")
22.     myRS("SuperAdmin") = Request.form("p_sadmin")
23.     myRS("Email") = Request.form("p_email")
24.     myRS.Update
25.
26.     myRS.Close
27.     Set myRS = Nothing
28.
29.     Mycon.Close
30.     set Mycon = Nothing
31. %>
32. <!--#include virtual="lffc/sa_Adminheader.html"-->
```

Line 1 declares what scripting language you are going to use. This line goes at the start of every ASP page. We are using VBScript. This is the most common one used for ASP pages.

Line 2 is the tag to open the script.

Lines 3 – 5 Checks to see if the user has gone through the login page, if they have the Session("Auth") variable is set to SA, and they can view this page. The code simply means if the Session("Auth") variable is not set to SA, then redirect them back to the login page.

Line 7 Declares the ado constant adopenkeyset=1

Line 8 Declares the ado constant adlockoptimistic=3.

Line 9 Declares the ado constant adcmdtext=&h0001

Line 11 Sets a variable p_lect_id to the p_lect_id value which was passed in from a form on the previous page.

Lines 13-14 Creates the connection to the database. Line 13 Creates an Active x data object database connection object, which is called Mycon.
On line 14 it then opens this connection, and the DSN name "Iffc" points it the correct database.

Line 16 Declares an SQL statement (mySQL), which selects all from the Admin table where the lecturer_id is equal to the variable p_lect_id which contains the p_lect_id passed in from the form.

Line 18 Creates a Recordset object called myRS.

Line 19 Opens the Recordset, and executes mySQL (to select administrator that we are looking for) it uses the connection object Mycon to tell it which database and contains 3 ADO declared constants, which are required for this action.

- AdOpenKeyset – Allows for concurrent updates by other users to be seen be you.
- AdLockOptimistic – Other users can still access a record while a user is editing it, but only until the changes are committed.
- AdCmdText – Allows for a text-defined command such as an SQL statement.

Lines 20-23 Sets the fields in the Recordset equal to the appropriate values passed in from the form.

Line 24 Performs the update action.

Line 26 Closes the Recordset.

Line 27 Sets the Recordset equal to nothing to destroy any references to the recordset and free up server resources.

Line 29 Closes the connection to the database.

Line 30 Sets the Connection equal to nothing to destroy any references to it and clear up server resources.

Line 31 Close of script tag.

Line 32 Includes the Server-Side Includes (SSI) that makes up the header of the web page you will see when the update has been performed.

5.5 Securing the application

IIS security implementation

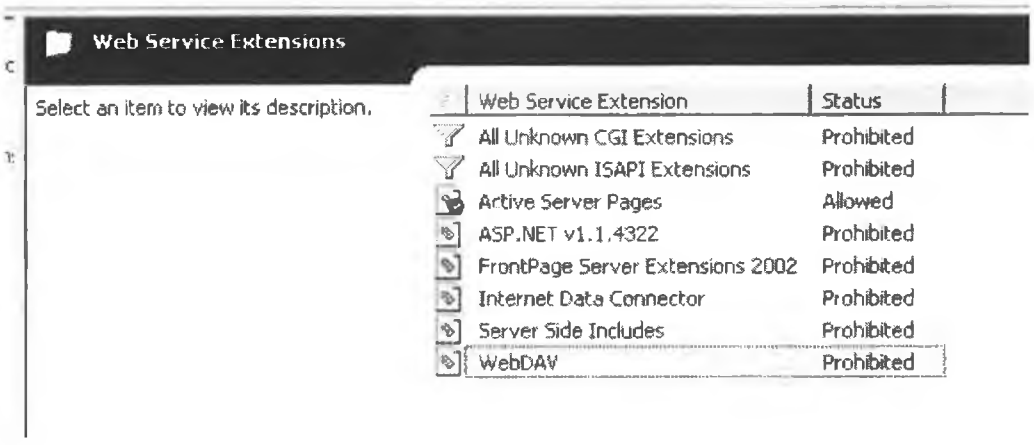


Fig. 18 *ASP pages are the only web service extensions allowed.*

IIS v.6.0 by default only allows specified applications to run on the server. As can be seen from above only ASP pages are allowed to run on this site. Also, the default permission on the website is read-only access so that anonymous users do not have full access to the server.

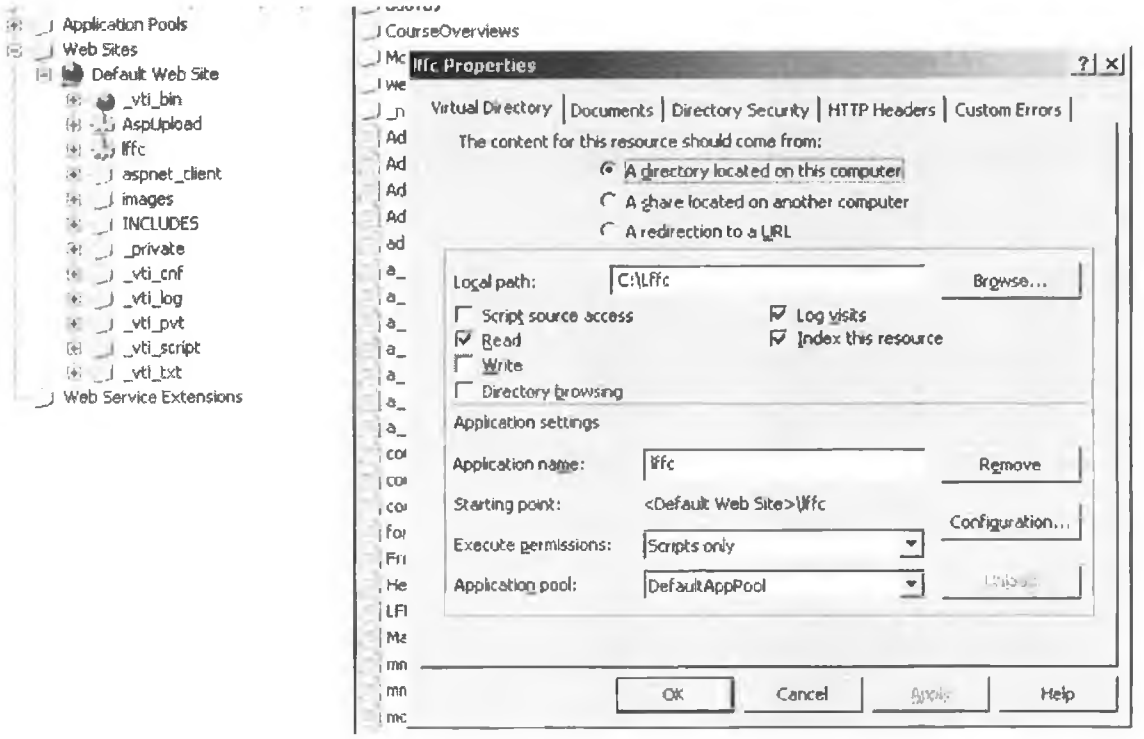


Fig. 19 *Default permission on the website is set to 'read only'*

In this project, to allow updating of the database and upload of files the webuser has been given NTFS write permission to the directory containing the database and upload directories. This is the minimum amount of permissions required for the application to work, as seen in the screen shot below of NTFS permissions.

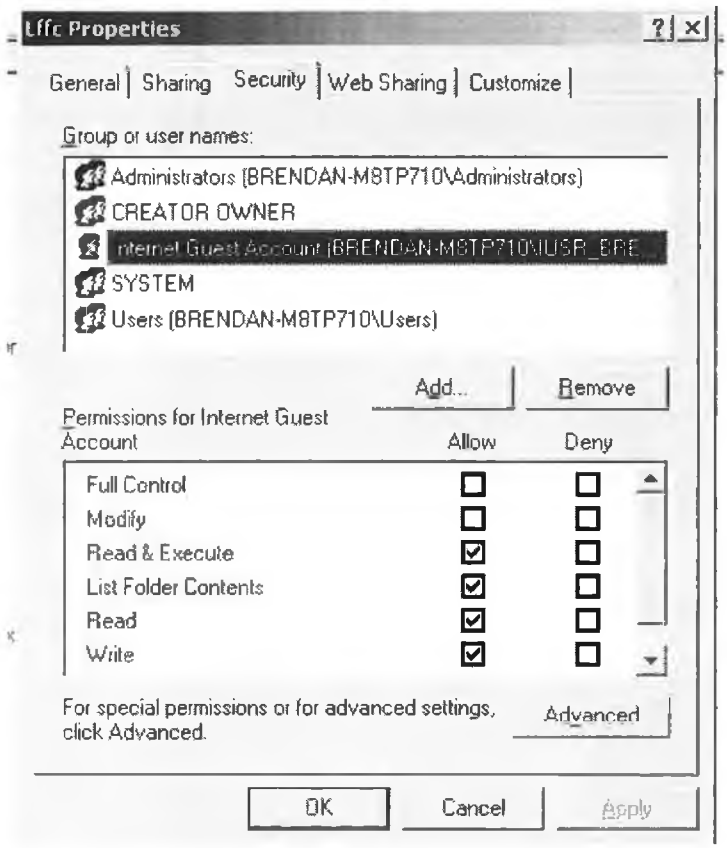


Fig. 20 *NTFS permissions allowed*

The normal way to secure a web site is to force users to use a login page where their usernames and passwords can be verified against a table containing a list of usernames and passwords in the database. The login page acts as a gatekeeper for the rest of the site if their user name and password cannot be verified they cannot access the rest of the site.

This method of security would be implemented using the session object in ASP. The rest of the site would be made unavailable to anyone who hasn't logged in by using a Session Authentication code in every

Page that requires the user to be logged in.

E.g.

```
<%
        If not Session("Auth") = "SA" then
                Response.Redirect"../default.asp"
        End if
%>
```

5.6 How it all works

This section demonstrates how the site functions from both the live site perspective and from the administration section. Screen shots are used from the site after it was installed on a virtual server. This demonstrates how ASP is used to perform a range of functions.

Fig. 21 *The Homepage Design. This is a standard template taken from Macromedia Dreamweaver. It complies with the requirements identified by the focus group, but can easily be changed at any point in the future.*

Fig. 22 *Selecting 'Course Notes' from the front end calls an ASP page which uses SQL to query the courses table in the database and dynamically generates URL links to all the courses from the courses table in the database.*

Courses : Table

	Course_ID	Course_Name
+	1	BSc in Furniture Design & Manufacture
+	2	BSc in Furniture Production & Technology
+	3	BSc (Hons) in Product Design (Furniture)
+	4	BSc (Hons) in Manufacturing Technology (Furniture)
+	5	BSc (Hons) in Design + Technology Education

Fig. 23 *Courses table from database referred to at Fig. 22.*

Fig. 24 *Once a course is selected links to the stages in that course are displayed. The results displayed are dynamically generated by querying the database to see what stages are available for the selected course.*
The Course Overview is also displayed dynamically from the database as some courses may not have an overview available.

Fig. 25 *When a stage for a particular course is selected the list of modules linked to that stage are dynamically displayed.*

Fig. 26 *Once a module title is selected the list of lecture notes attached to that module is displayed. The accompanying video demonstrates that a range of different file types function (Word, Powerpoint and Adobe Acrobat).*

Fig. 27 *The search page is used for the student search of module notes. It is a standard HTML form, which will pass the value to 'search_action.asp', which will perform the actual search.*

Fig. 28 *Admin section showing login details required. When the login button is pressed this action calls an ASP page, which 1) queries the database to ensure the username is valid and 2) checks the password. If either is invalid no login is allowed. If both are valid the session variable is set to authenticate the data. A variable is also set depending on the admin level, which is queried from the database.*

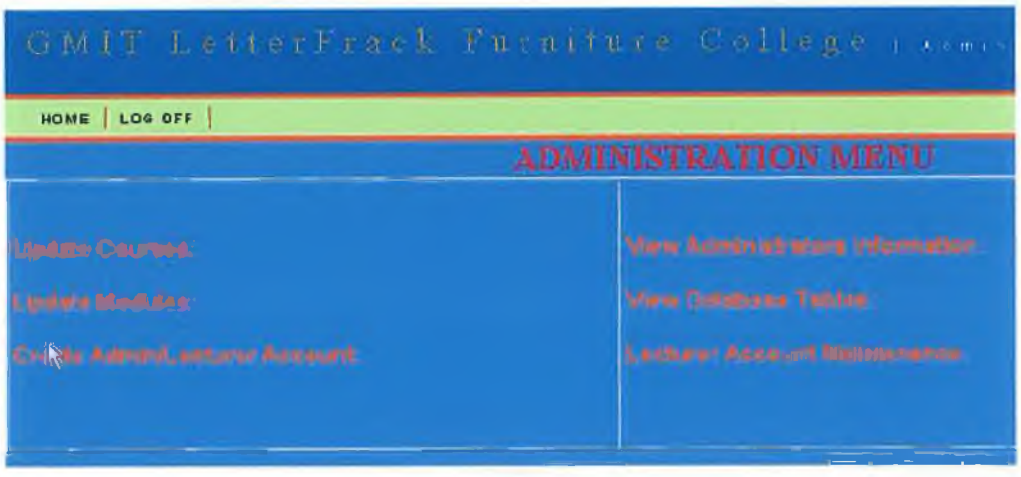


Fig. 29 *The administration menu which is only available to the Super Administrator (SA).*

Fig. 30 *Updating Courses Admin Menu Three options*

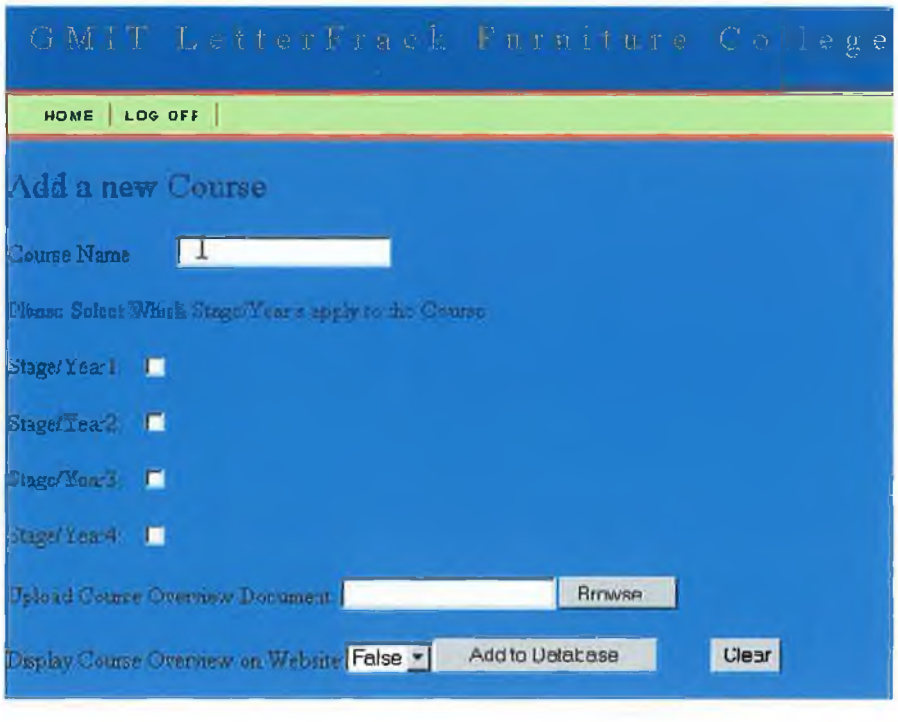


Fig. 31 *Add a new course form*

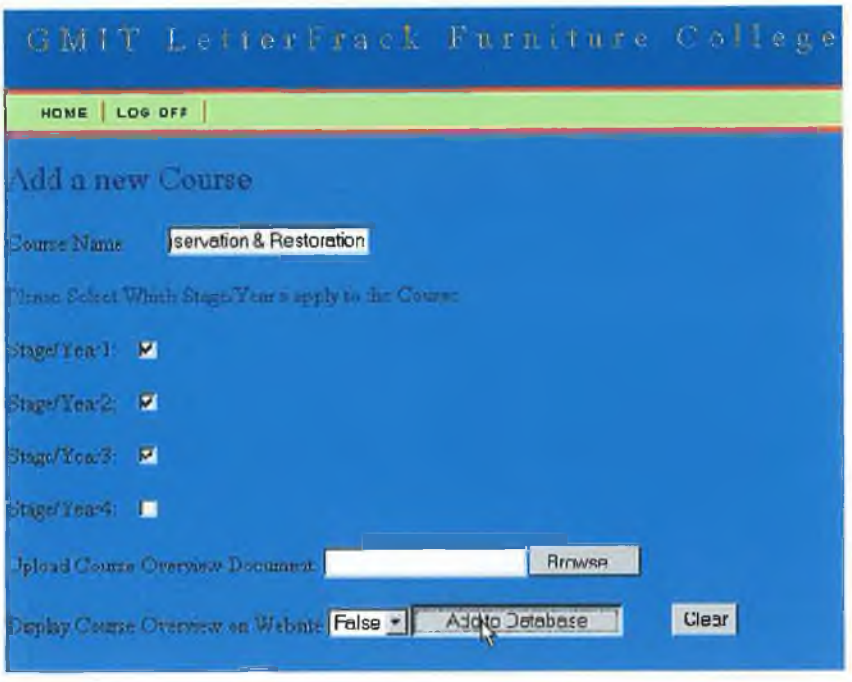


Fig. 32 *With the course name inserted, the SA can select what year/stage is associated with this course before it is added to the database. ASP performs a SQL add function to add the data to the relevant table in the database.*

Fig. 33 *Confirmation that the course was added to the database.*

Fig. 34 *To edit course details, select 'edit existing course details' from the Update Courses menu. Then select the course you wish to edit. The list of courses in the drop down menu is dynamically populated from the database.*

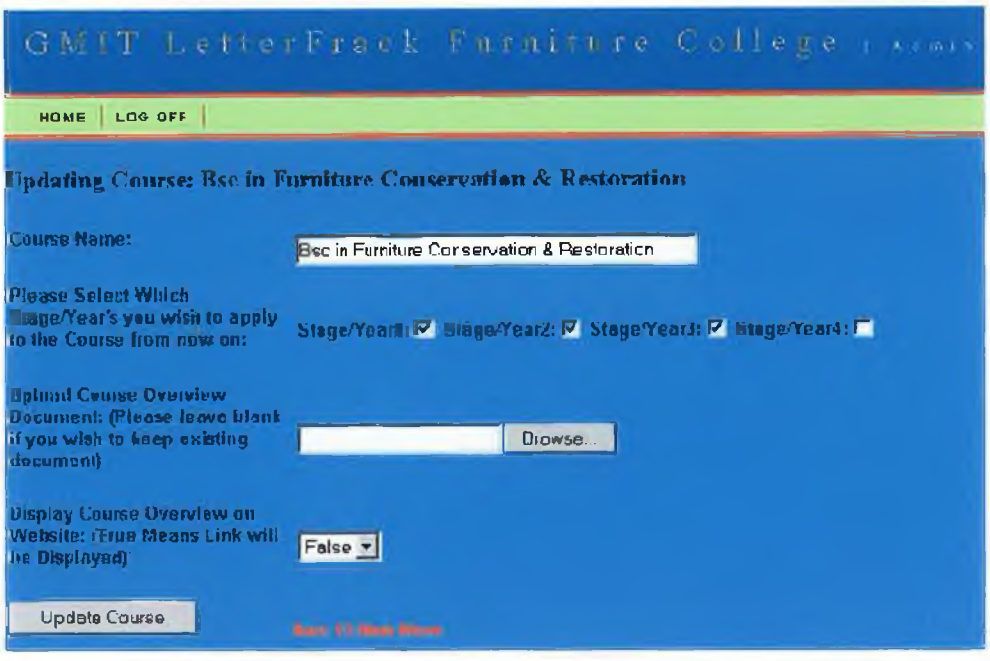


Fig. 35 *Selecting the 'Get Course Details' (Fig. 34), calls this form.*

Fig. 36 *Returning to the live site verifies that a 4th stage has been added to this course.*

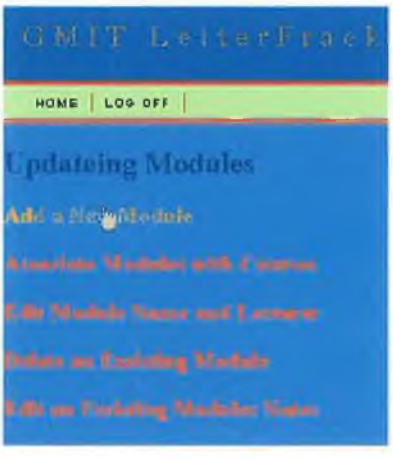


Fig. 37 *To add a new module, first select 'Update Modules' from the Admin menu and then select 'Add a new module' as shown opposite.*

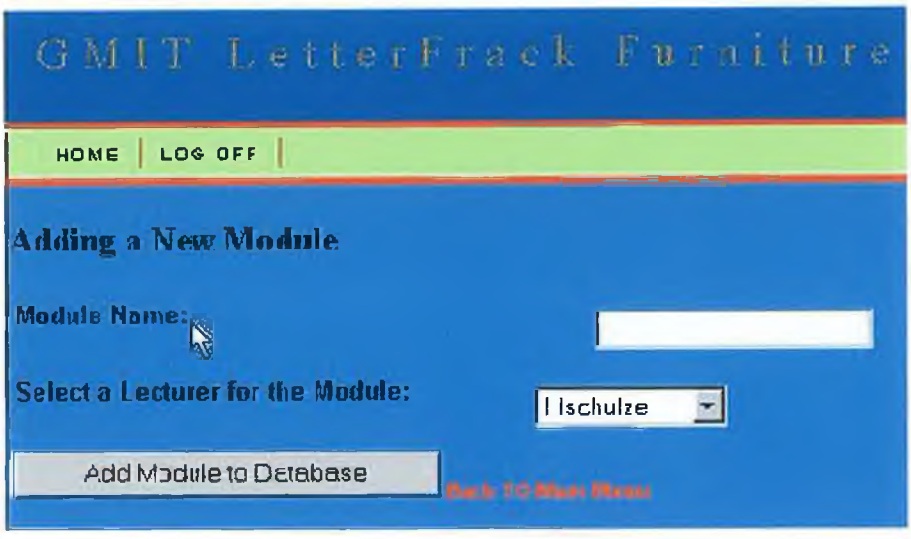


Fig. 38 *Selecting 'Add a new module' calls this form. This form also allows a module to be associated with a particular lecturer.*

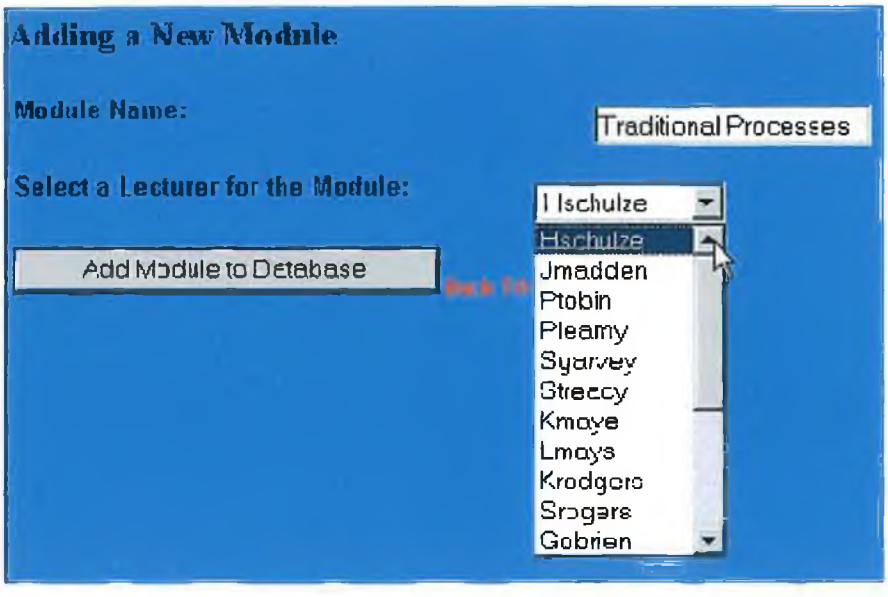


Fig. 39 *The lecturer's name can be selected from a drop down list, which is dynamically populated from the admin table in the database.*

Fig. 40 *Confirmation that the module has been added to the database.*

Fig. 41 *The next step is to associate this module with a course.*

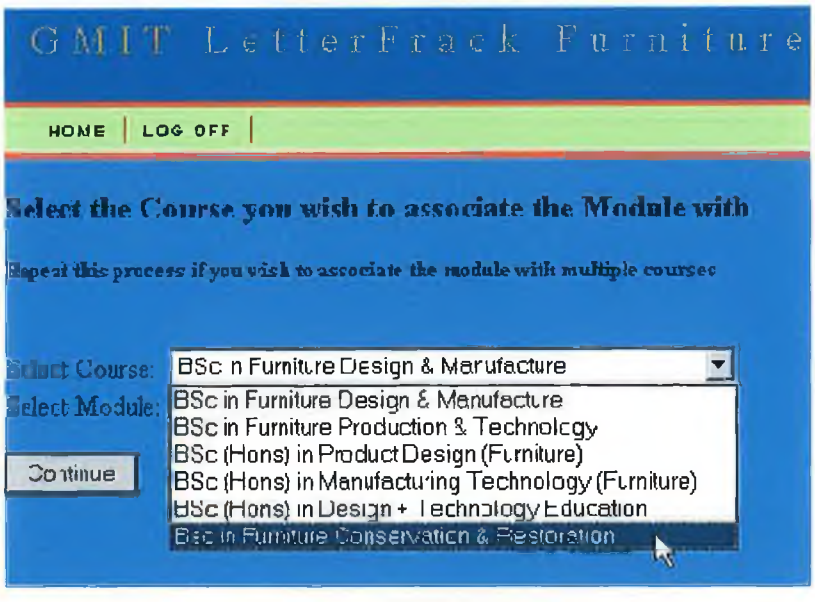


Fig. 42 *Drop down menu of available courses.*

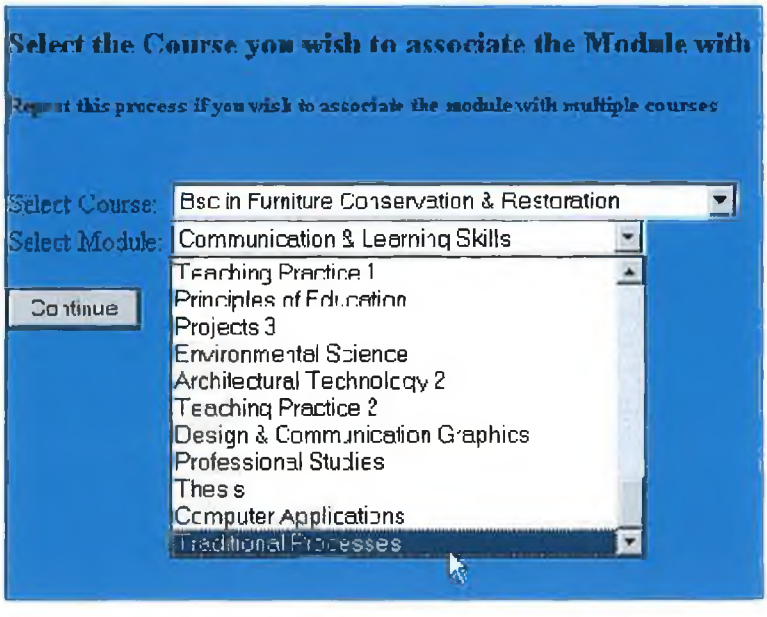


Fig. 43 *Module to be associated with the course selected from a drop down list.*

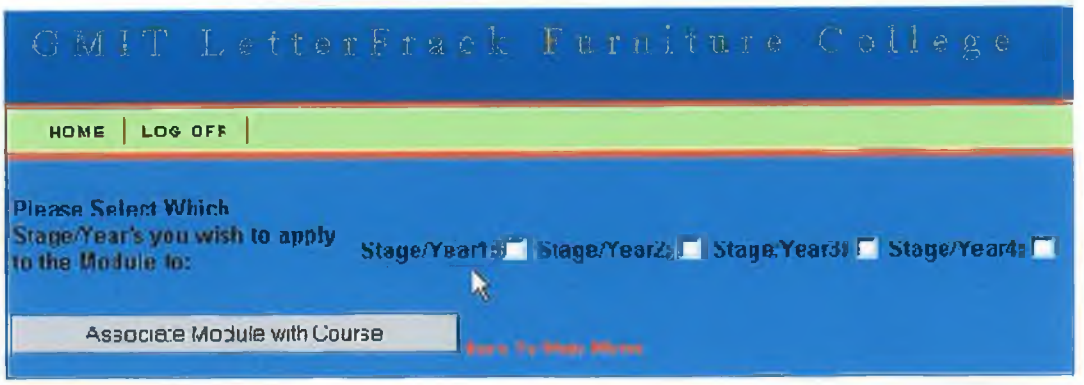


Fig. 44 *The module can then be linked to one or more years of the course.*

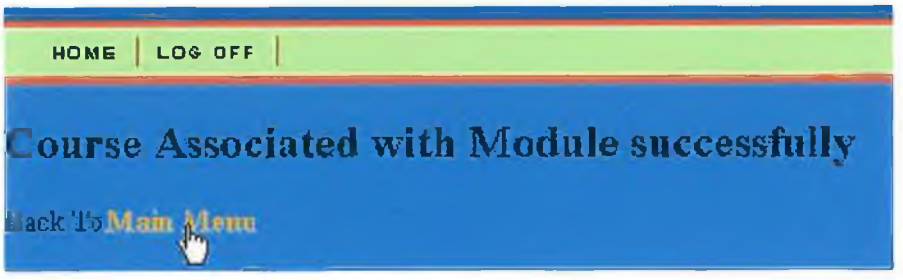


Fig. 45 *Confirmation that the module was successfully associated with the course.*

Fig. 46 *Select this option to add a set of notes to a module.*

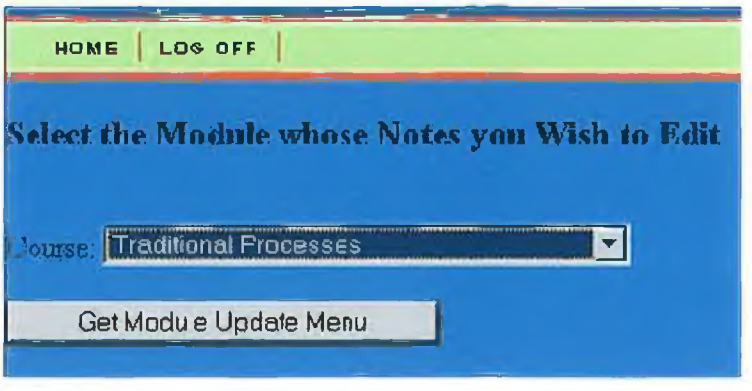


Fig. 47 *Select the module to add notes to. The list of mofules is dynamically populated from the database.*

Fig. 48 *This menu allows for notes to be added, deleted or edited.*

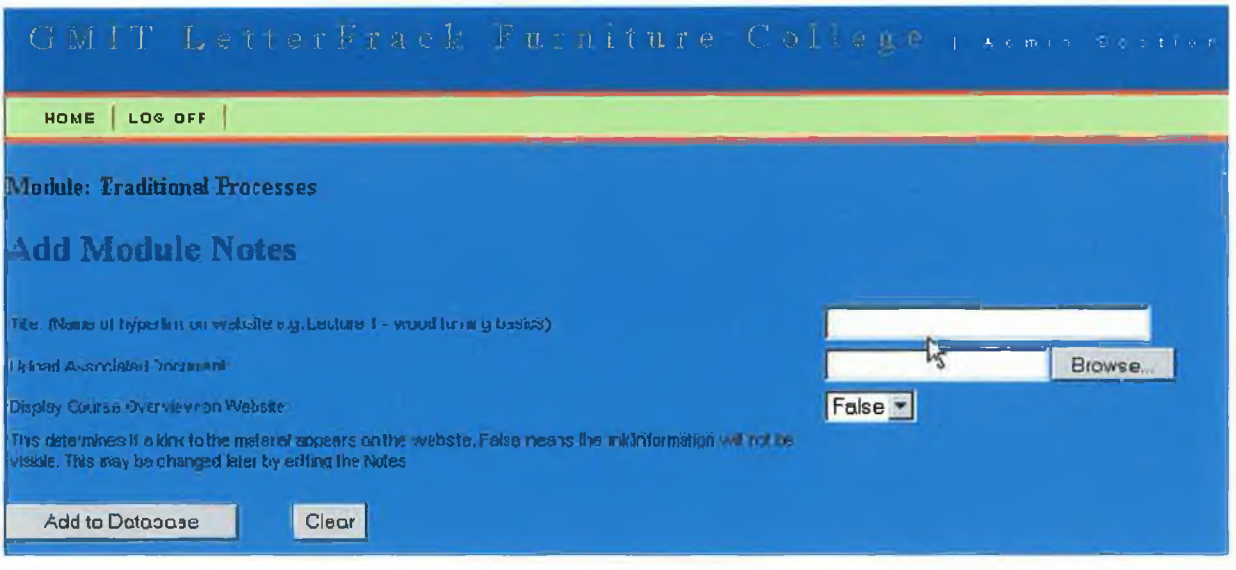


Fig. 49 *Selecting the 'Add New Notes' calls this form, which also allows whether or not the notes are displayed on the live site.*

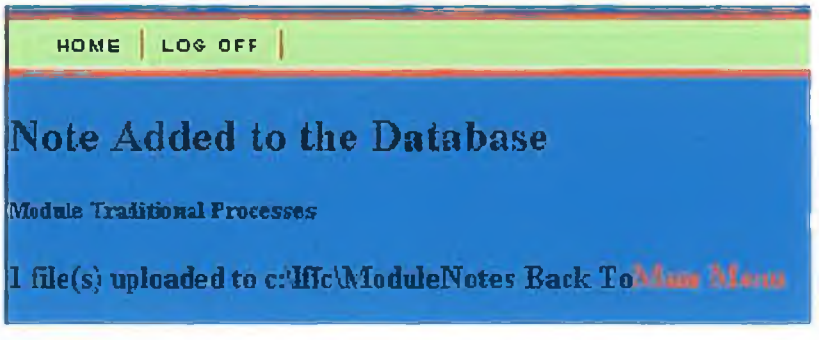


Fig. 50 *Confirmation that the notes have been added to the database.*

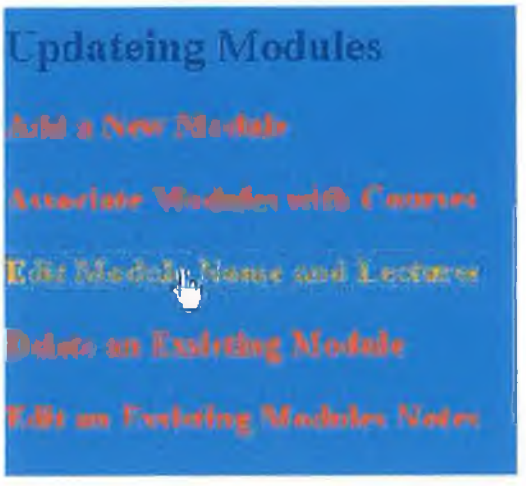


Fig. 51 *Select this option to edit the module name and the lecturer associated with the module.*

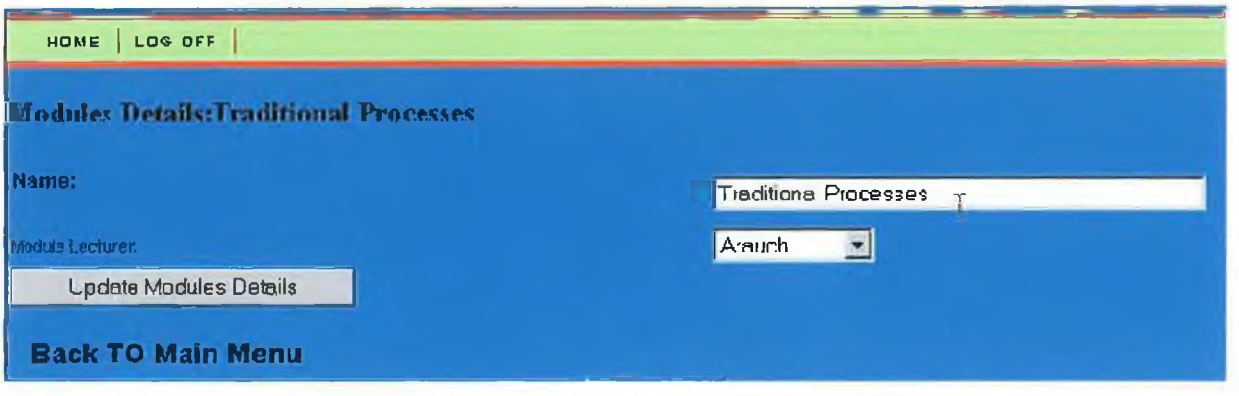


Fig. 52 *Changes to Module Names and associated Lecturers are made here.*

Fig. 53 *Select this option to add a new course to the database.*

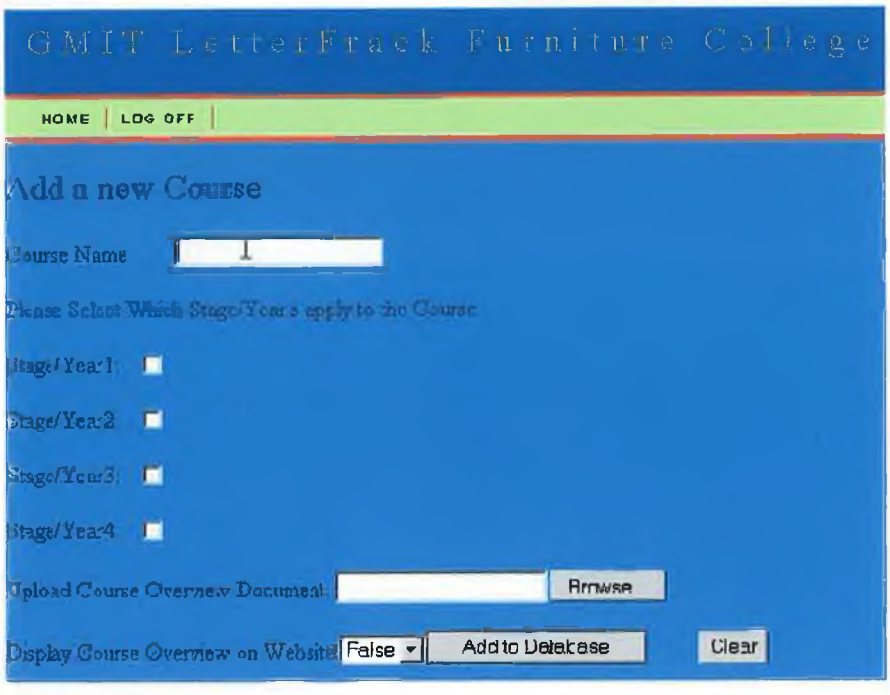


Fig. 54 *New course details form.*

Fig. 55 *Confirmation from the live site that the new course (Dummy Course) has been successfully added.*

Fig. 56 *Similarly courses can be deleted from the database.*

Fig. 57 *Select the course to be deleted.*

Fig. 58 *Select to edit an existing course.*

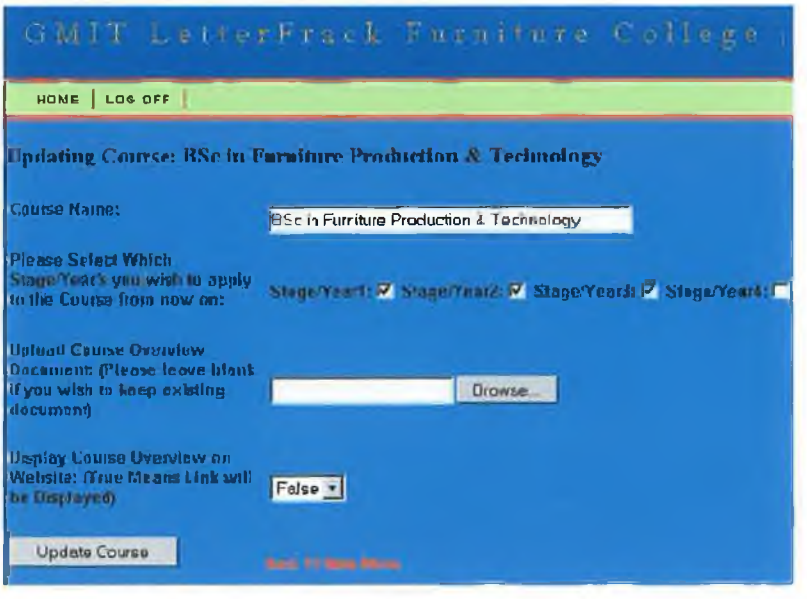


Fig. 59 *Form to edit an existing course.*

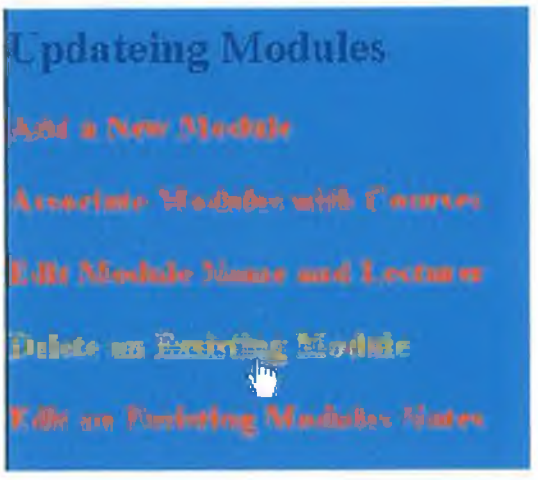


Fig. 60 *Select this option to delete a module*

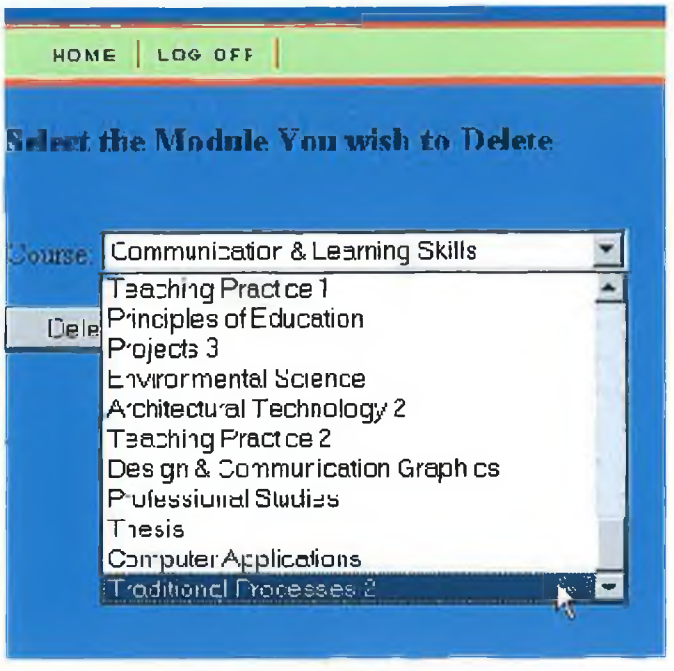


Fig. 61 *Select the module to delete from the dynamically generated drop down list.*

Fig. 62 *Confirmation that the module has been deleted.*

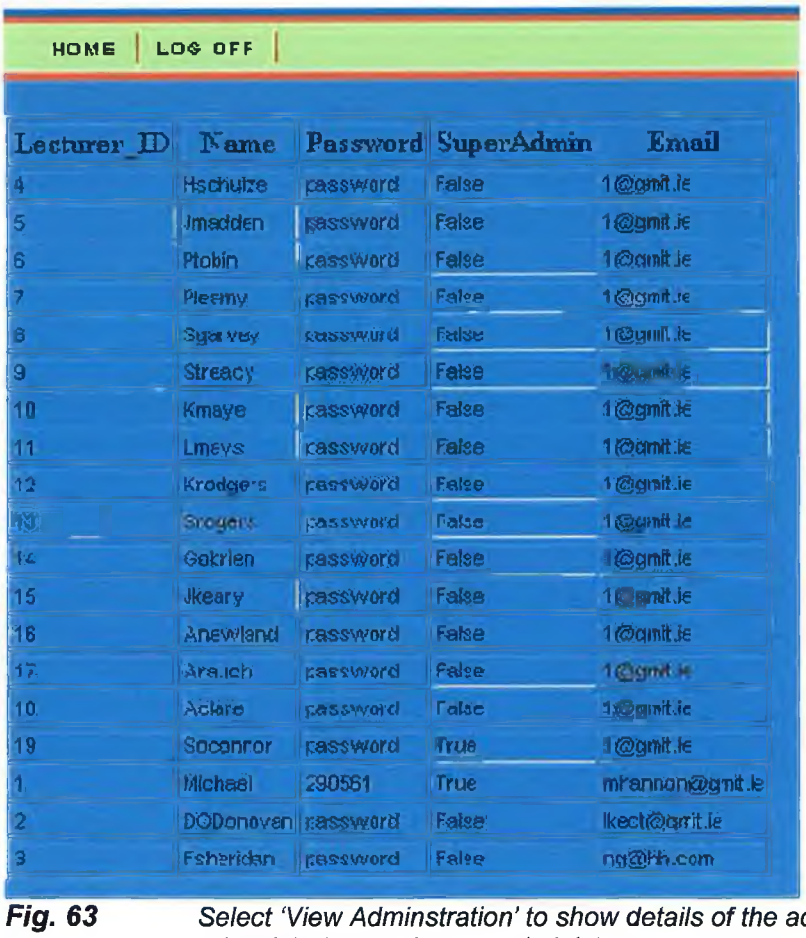


Fig. 63 *Select 'View Adminstration' to show details of the admin table which includes administrators and super administrators.*

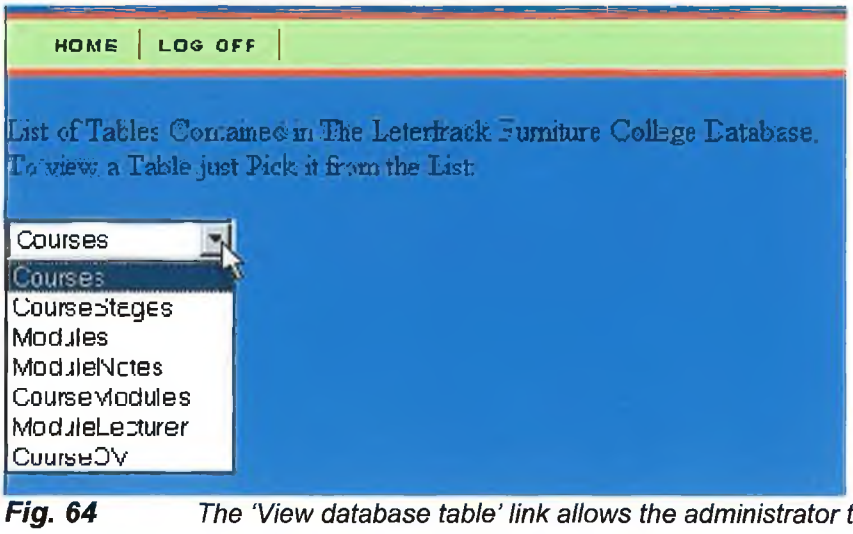


Fig. 64 *The 'View database table' link allows the administrator to view all tables in the database. Select from the drop down list.*

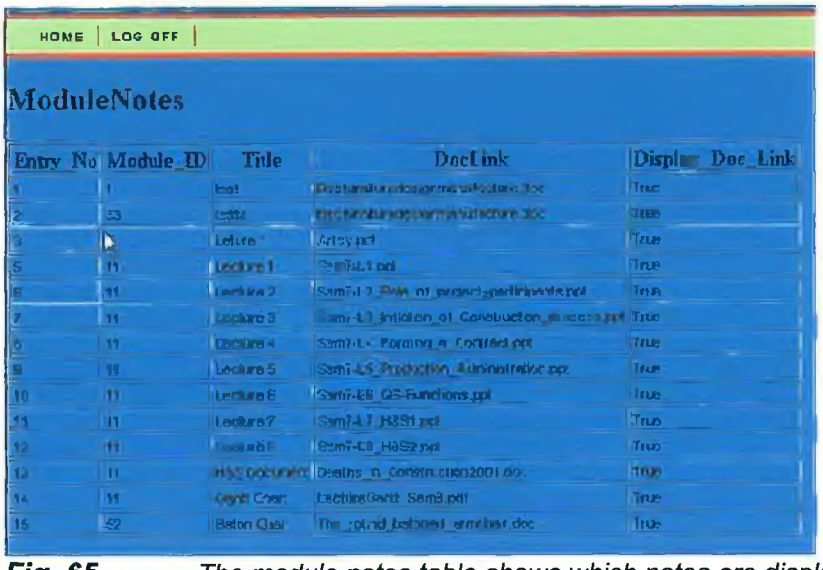


Fig. 65 *The module notes table shows which notes are displayed.*

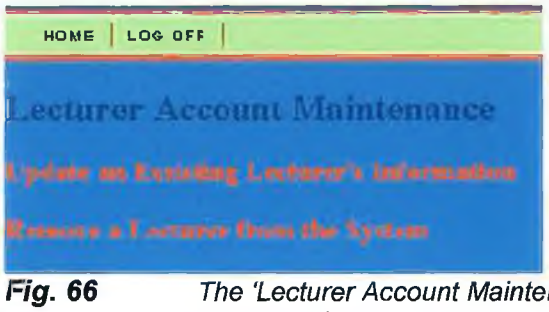


Fig. 66 *The 'Lecturer Account Maintenance' menu allows the administrator manage lecturer's rights.*

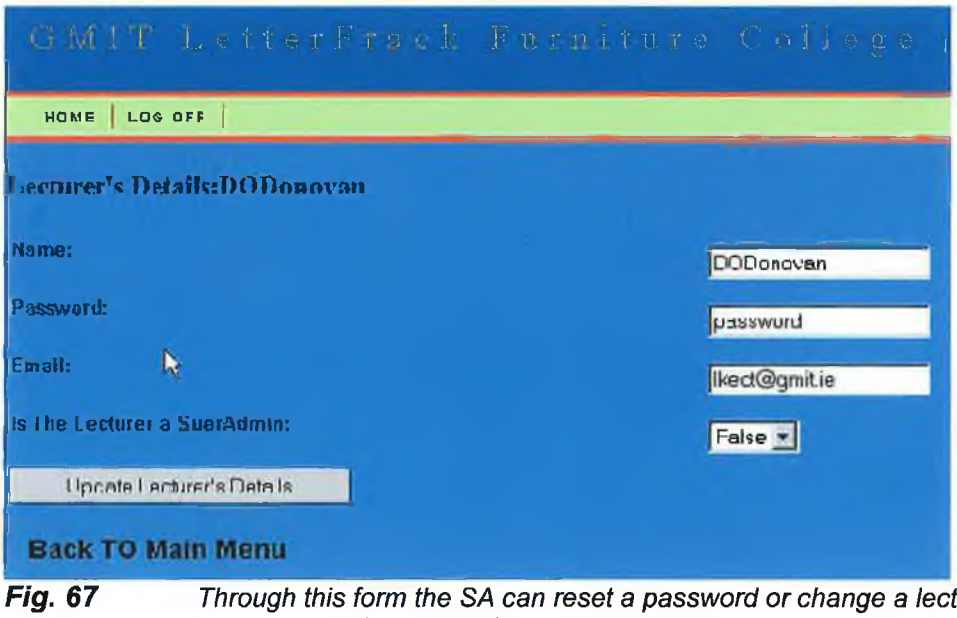


Fig. 67 *Through this form the SA can reset a password or change a lecturer's status to Super Administrator by setting the value above to True.*

Fig. 68 *Confirmation that changes have been updated.*

Fig. 69 *Select this option to add a new lecturer to the system.*

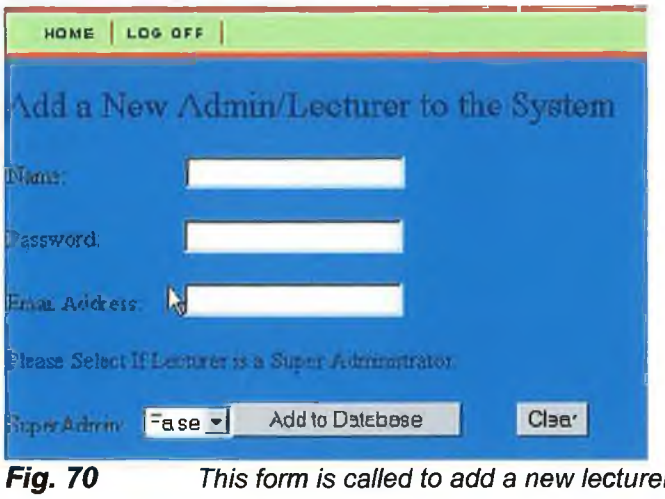


Fig. 70 *This form is called to add a new lecturer to the system.*

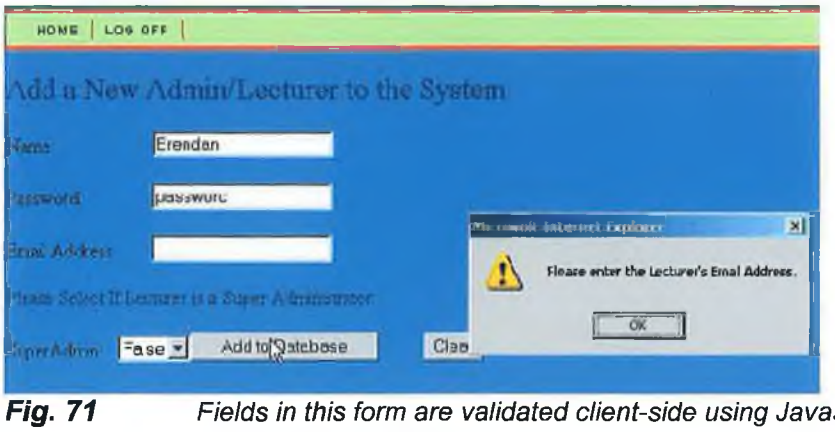


Fig. 71 *Fields in this form are validated client-side using Javascript to ensure correct information is entered.*

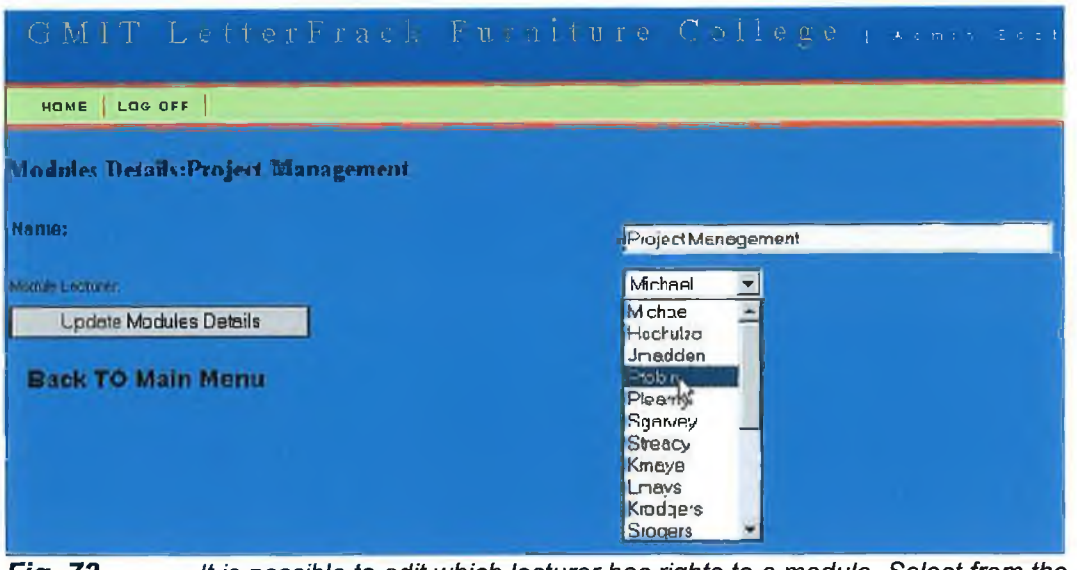


Fig. 72 *It is possible to edit which lecturer has rights to a module. Select from the dynamically generated list.*

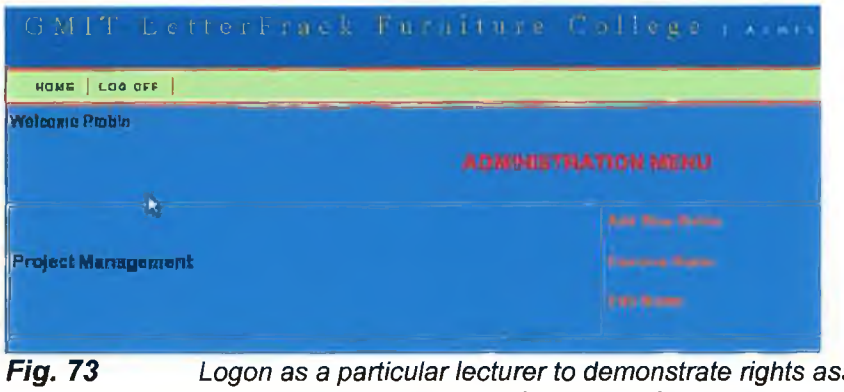


Fig. 73 *Logon as a particular lecturer to demonstrate rights assigned.In this example Ptobin only has rights to the module Project Management.*

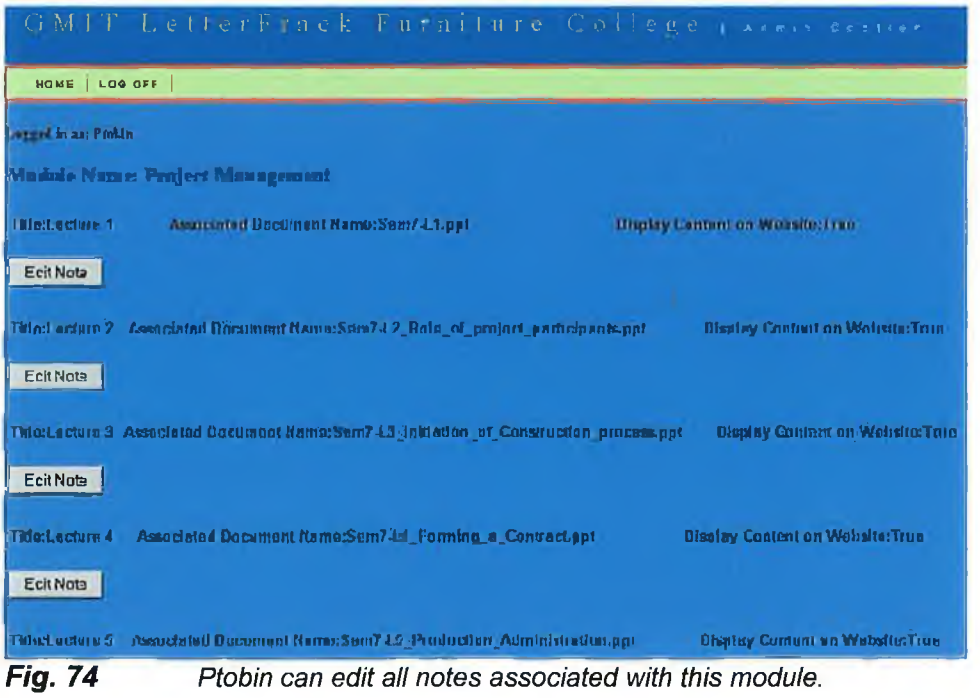


Fig. 74 *Ptobin can edit all notes associated with this module.*

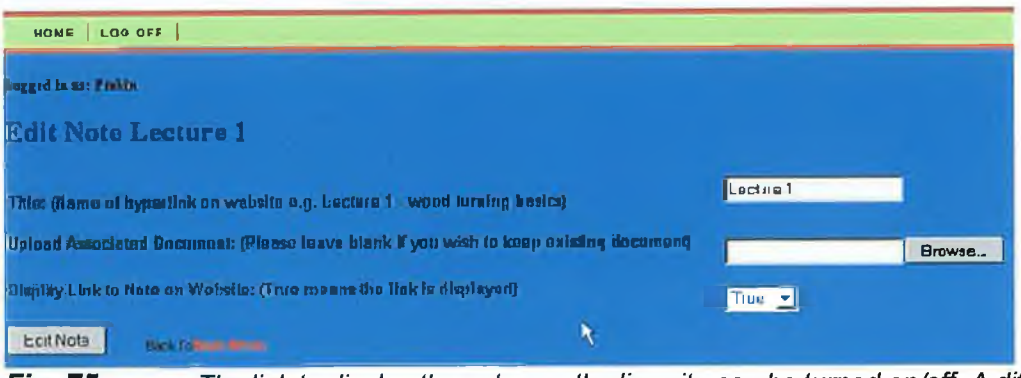


Fig. 75 *The link to display the notes on the live site can be turned on/off. A different document can also be associated with the lecture.*

Fig. 76 *Lecture 1 notes no longer displayed on the live site as decided by the lecturer (administrator).*

Fig. 77 *Only the SA can associate a module with a lecturer. In this example the Furniture Construction module is now linked to Ptobin.*

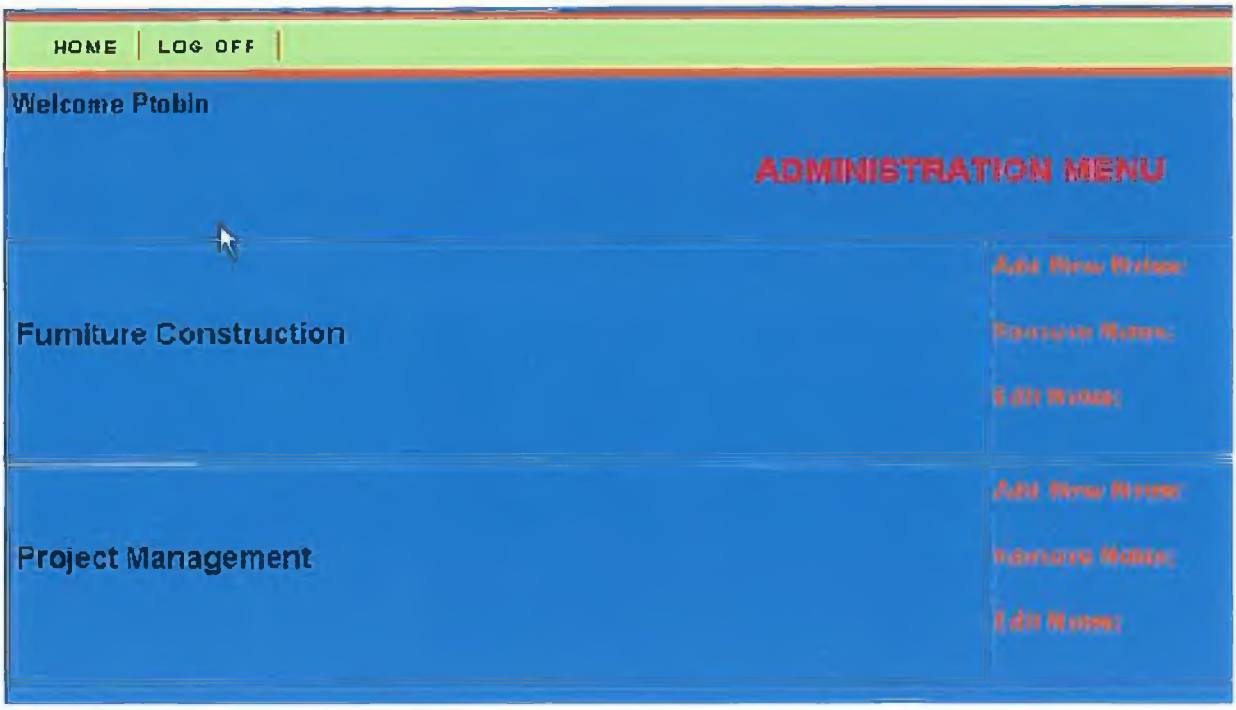


Fig. 78 *The next time Ptobin logs in he will see he has rights to two modules as shown above.*

Additional information on the functionality of this project is contained on the .avi file on the accompanying CD.

Chapter 6. DISCUSSIONS and CONCLUSIONS

Intranets are designed to focus on employees and improve workflow within an organisation. The idea that the intranet needs to be all things to all people has been replaced. According to Pastore (2003), ".. the focus has switched to adding value. In place of news headlines, the best intranets feature company news and information that makes employees better informed and increase employee morale."
He goes on to say that one common characteristic in the best intranets is that instead of making users find relevant information on the site, information is being given to users. The author has attempted to achieve this with the first phase of the intranet developed as part of this project.

Giving relevant information to users does present a challenge and that is the fine line of offering the users the information they need when they log-on and giving them too much information. According to Neilsen (2005) the answer to the problem is within the intranet environment by filtering the information that each person needs to see.

What does the future hold for intranet development?
This project is the first step in developing and implementing an intranet that meets the needs of the organisation as identified by a focus group. Developments and trends in relation to intranets need to be monitored and incorporated into this design where it is deemed appropriate. Some of the current intranet trends include:

- Customers/employees are becoming the focus of the system.
  This trend has already been incorporated by the author through regular meetings of the focus group and presentations at staff meetings. An international example of this trend is the Dell intranet/extranet. The website concentrates on helping the customer to succeed. With over 10,000 customers the ordering process is extremely simplified. Intranet users are now demanding a more aesthetic, simpler and more intuitive experience. Singh (2006) refers to this as the "Google Effect". He says that if Google can provide a quality user experience then so should the company intranet.

- The intranet is a utility.

  The intranet is now essential for doing business. The realisation of this principle was the driving force for the author in selecting this project in the first instance. As an Institute of Technology we were not using technology to effectively conduct and manage our business. For intranets to succeed it is important that they are supported and funded by senior management. When senior management actively get involved in their development, as in this case, success must be assured?
  In industry the importance of the intranet is well recognised. In Cisco for example, ninety-eight percent of its employees use their intranet on a regular basis. At Ford the objective is to have as many employees as possible find the information they need on the intranet.

- "Less is More."

  Architect Ludwig Mies van der Rohe adopted the motto "*Less is more*" to describe his aesthetic tactics of flattening and emphasizing the building's frame, eliminating interior walls and adopting an open plan, and reducing the structure to a strong, transparent, elegant skin. (www.wikipedia.org - April 2007).
  This phrase also has relevance to trends in intranet systems as more and more companies are using small teams to build and develop the intranet system. Success is achieved by focusing on planning and design. Less is more coule equally be applied to this project as a small team is driving the development initially.

- Intranets grow up.

  In the future, according to Singh (2006), intranets will move "beyond communication, information sharing and employee self-service. They will include more process-specific business applications and personal information management tools like calendaring." At present this project is at the communication and information sharing stage. This is adequate as a live project to bring together the technologies experienced in the taught modules of this programme and to demonstrate how they can be combined into a full working project.

- Intranet ROI no longer an issue.

  The literature review has shown up until recently that inabilty to demonstrate Return on Investment was the main reason for lack of support from management for intranet development. That is no longer the case as executives have now seen how intranets can empower the workforce and increase employee productivity. Intranets have become indispensable tools for information sharing, knowledge management and news dissemination. The author looks forward to these goals being realised when this intranet is fully implemented and used by all employees. In an educational context ROI is not an issue and never would have been from my perspective.

- The future of blogs?

  Chat rooms, discussion boards and blogs were not considered by the author for this project. However, their development should be considered in the next development phase as they are far more useful as collaborative tools than say email. Blogs are a recent phenomena but their future is not certain. For blogs to be successful it is important for the blogger to have something important and/or interesting to say that will attract return visits. The most popular blogs, according to recent survey by America Online, are the most personal and opinionated ones. If the information is censored bloggers will be reluctant to offer information and it will have less appeal to the audience.

- Real Simple Syndication (RSS)

  RSS is an XML format for syndicating web content. The forecast is that companies using this technology to publish information on intranets will have greater success than with blogging. This technology will enable users to subscribe to subject and department specific information, they will be able to view it via readers thus enabling more targeted conversations in the workplace. For RSS to work effectively users will have to be allowed to both internal and external RSS feeds. If this happens, blogging along with RSS readers might well become more relevant that the intranet for some organisations.

- The wiki phenomena.

  Wiki is another relatively new web publishing phenomena. It allows users to edit

and update existing web pages. The Wikipedia encyclopedia is a well known example of this technology. A key disadvantage of this approach for intranets is the concern management have with users/employees having the ability to edit the intranet. Who is in control of the image/site if everybody has editorial rights?

To conclude therefore, it can be seen that the aim of this project was to develop a database driven intranet site. This was achieved by firstly identifying the requirements of the site, creating a database using Microsoft Access and then using an ODBC connection with a System DSN to connect to ASP web pages.
The overall system is a study of how information can be exchanged between a database and a web page. The system developed for this project gives an idea of what can be achieved using the technologies involved. In the course of this project I have gained an understanding of ASP and what is involved in the development of a database driven website.

The phenomenom that is intranets, according to Hinrichs (1997), was happening before the term was around to describe it. In that context it could be argued that developing a project such as that described by the author in this report is coming to the game late. This view might be further reinforced when one considers the future of intranets as described in this chapter. However, it can be counter argued that intranets are designed to serve the needs of the organisation and the project as described and setup meets those needs.

According to Stone Gonzalez (1998), "The right metric for intranet success is consistent use over time." The author will be extremely pleased if this metric holds true for this project. Once this is realised further elements can be added to the intranet, which were outside the scope of this project. These include *inter alia*:

- An events calendar
- Message / discussion boards
- Staff and student contact details
- Repository of claim forms
- Integrating with a Learning Management System such as Moodle
- Content management system

BIBLIOGRAPHY

Ablan, J. (1996) Developing Intranet Applications with Java. Sams Publishing.

Active Server Pages. Available at: http://www.visionengineer.com. Accessed on 14-01-2007.

Bangar, P. (2003) Introduction to Intranets. Available at http://www.promotiondata.com. Accessed on 20-08-2006

Berlind, D. (1996) Intranets Could Solve Many IT Issues. PC Week 13, no. 3 (January 22, 1996): 68.

Berners-Lee, T. (1989) Information Management: A Proposal. Available at http://www.w3.org/History/1989/proposal.html. Accessed on 14-10-06

Butler, R. (2000) The Internet Demystified. Oak Tree Press.

Buyens, J. (2000) Step by Step Web Database Development. Microsoft Press.

Chase, N. (2000) Active Server Pages 3.0. Que publishing.

Chipman, M., Baron, A. (2001) Microsoft Access Developer's Guide to SQL Server. Sams Publishing.

Colby, J., Downes-Powell, G., Haas, J., Harkness, D.J., Pappas, F., Parsons, M., Storr, F., Surguy, I., Voigt, R. (2003) Practical Intranet Development. Glasshaus Ltd.

File DSNs – Use and Reuse. Available at: http://www.kamath.com. Accessed on 14-01-2007.

Finding ROI (2005) Measuring Intranet Investments. Available at http://www.prescientdigital.com. Accessed on 12-08-2006

General advantages and disadvantages of HTML vs XML and XHTML. Available at http://www.e-articles.info/e/a/title Accessed on 12-01-07

Goldfarb. C. SGML Source Home Page. Available at http://www.sgmlsource.com/index.htm. Accessed on 12-10-06

Hafner, K., Lyon, M. (1996) Where Wizards Stay Up Late: The Origins of the Internet. Simon & Schuster

HEA (2004) Review of Training of Trainers Programme. Available at http://hea.ie/uploads/pdf. Accessed on 20-07-2006

Hinrichs, R.J. (1997) Intranets – What's the Bottom Line? Prentice Hall
Horgan, T. What's Happening Inside? Available at: http://www.cio.com/research/intranet. Accessed on 10-01-2007

http://en.wikipedia.org/wiki/Vmware Accessed on 14-08-2006
Intranet Research Main Findings Available at: http://www.ptools.com/More_pTools/Intranet_Research/Main_Findings. Accessed on 05-04-2007

Introduction to ADO. Available at: http://www.w3schools.com. Accessed on 14-01-2007.

JavaScript. Available at http://en.wikipedia.org/wiki/JavaScript. Accessed on 12-01-2007.

Key Trends that Raised the Bar. (2006) Available at: http://www.zeroonezero.com. Accessed on 6-01-2007.
Koehler, J.W., Dupper, T., Scaff, M.D., Reitberger, F., Paxson, P. (1998) The Human Side of Intranets. St. Lucie Press

Lynch, P.J., Horton, S. (1999) Web Style Guide. Yale University Press.

Miller, M.A. (1991) Internetworking. M&T Books.

Molich, R. (2004) Strengths and Weaknesses of Heuristic Reviews. Available at http://www.iorg.com/papers. Accessed on 20-08-2006

Neilsen, J. (2006) Quantitative Studies: How many users to test? Available at http://www.useit.com. Accessed on 20-08-2006

Neilsen, J., Loranger, H. (2006) Prioritizing Web Usability. New Riders.

Neilsen,J. (2005) The Canonical Intranet Homepage. Available at http://www.useit.com. Accessed on 23-05-2006

Norman, D. (1993) Things That Make Us Smart: Defending Human Attributes in the Age of the Machine. Addison-Wesley.

Open Database Connectivity. Available at: http://en.wikipedia.org. Accessed on 20-01-2007.

Pastore, M. (2003) Building the perfect beast. Available at: http://www.intranetjournal.com. Accessed on 20-01-2007

Preece, J., Rogers, Y., Sharp, H. (2002) Interaction Design. John Wiley & Sons, Inc.

Schwartz, P., Leyden, P. (1997) The Long Bloom: A History of the Future, 1980-2020.

Singh, S. (2006) Intranet Trends to Watch For: 2006. Available at http://www.avenuea-razorfish.com. Accessed on 20-08-2006.

Singh, S., Velez, R. (2006) Corporate Intranets Best Practices Report. Available at http://www.avenuea-razorfish.com. Accessed on 20-08-2006.

Smith, B., Komar, B. (2005) MS Windows Security. Microsoft Press.

Spoull, L., Kiesler, S. (1993) Connections: New Ways of Working in the Networked Organization Cambridge: MIT Press (p. 9).

Stone Gonzalez, J. (1998) The 21st Century Intranet. Prentice Hall PTR.

SQL Databases – powering todays Application.
Available at http://www.daffodildb.com. Accessed on12-01-2007

Teelan, S. L. (2003) Do you really want an intranet. Available at http://www.iorg.com. Accessed on 24-08-2006

Teelan, S. T. (2005) Intranet Organization: Strategies for managing change. Available at http://www.iorg.com. Accessed on 3-08-2006

Ward, T. (2006) Best Practices: Five Winning Intranet Characteristics. Available at http://www.intranetblog.com. Accessed on 7-07-2006

APPENDIX

A_ADDMODNOTES.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<%
    if not Session("Auth")="A" then
       Response.Redirect"admin.asp"
    end if

   p_lect_id = Session("Lecturer")
   p_mod_id = Request.querystring("p_mod_id")

   set Mycon = Server.CreateObject("ADODB.Connection")
   Mycon.Open "lffc"

   set catSet = Mycon.Execute("select * " _
   & "from Admin " _
   & "Where Lecturer_ID = "&p_lect_id)
   while not catSet.EOF
   p_name = catSet("Name")
        catSet.MoveNext
    wend
    catSet.Close
    set catSet = Nothing


%>

<html>
<head><SCRIPT LANGUAGE = "JAVASCRIPT">
    function validate() {
       if (document.form4.p_title.value.length < 5) {
          alert("Please enter a Title for the Notes Link.");
          return false;
          }
          return true;
       }
    </SCRIPT></head>
<!--#include virtual="lffc/Adminheader.html"-->
<body>
<b><h5>Logged in as: <%=p_name%></h5><B>
<p><font size="5" color="#000080">Add Module Notes</font></p>
<form method="POST" enctype="multipart/form-data"
action="a_add_modnotes_action.asp" name="form4" onSubmit="return validate();">
<table width="837">
  <TR><TD><INPUT TYPE="hidden" NAME="p_mod_id" VALUE="<%=p_mod_id%>"></TD></TR>
  <tr><td width="529">Title: (Name of hyperlink on website e.g. Lecture 1 - wood
   turning basics)</td><td><input type="text" name="p_title" size="31"></td></tr>
  <tr><td width="529">Upload Associated Document:</td><td><input type="file"
name="p_file"></td></tr>
  <tr><td width="529">Display Course Overview on Website:</td><td><SELECT
NAME="p_display"><OPTION VALUE="False">False<OPTION value="True">True</td></tr>
  <tr><td width="529">This determines if a kink to the material appears on the
website,
   False means the link/information will not be visible. This may be changed
   later by editing the Notes</td><td></td></tr>
</table>
  <p><input type="submit" value="Add to Database"
name="B1">        
  <input type="reset" value="Clear" name="B2"></p>
</form>
<%
Mycon.Close
set Mycon = Nothing
%>
</body>
</html>
```

A_DELMODNOTES_ACTION.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<!--#include virtual="lffc/adminheader.html"-->
<%
    if not Session("Auth")="A" then
       Response.Redirect"admin.asp"
    end if

   p_lect_id = Session("Lecturer")
   p_entry_no = Request.Form("p_entry_no")

   set Mycon = Server.CreateObject("ADODB.Connection")
   Mycon.Open "lffc"

   set catSet = Mycon.Execute("select * " _
   & "from Admin " _
   & "Where Lecturer_ID = "&p_lect_id)
   while not catSet.EOF
   p_name = catSet("Name")
        catSet.MoveNext
    wend
    catSet.Close
    set catSet = Nothing

    mySQL = "DELETE FROM ModuleNotes WHERE Entry_No = " &p_entry_no
    Mycon.Execute(mySQL)

     Mycon.Close
     set Mycon = Nothing
%>
<b><h5>Logged in as: <%=p_name%></h5><B>
<P><a href="admin_main.asp">Return To Main Menu</a></P>
</BODY>
</HTML>
```

A_EDITMODNOTES.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<%
    if not Session("Auth")="A" then
       Response.Redirect"admin.asp"
    end if

   p_lect_id = Session("Lecturer")
   p_mod_id = Request.querystring("p_mod_id")

   set Mycon = Server.CreateObject("ADODB.Connection")
   Mycon.Open "lffc"

   set catSet = Mycon.Execute("select * " _
   & "from Admin " _
   & "Where Lecturer_ID = "&p_lect_id)
   while not catSet.EOF
   p_name = catSet("Name")
        catSet.MoveNext
    wend
    catSet.Close
    set catSet = Nothing


   set catSet1 = Mycon.Execute("select * " _
       & "from Modules " _
       & "Where Module_ID = "&p_mod_id)
   while not catSet1.EOF

   p_mod_name = catset1("Module_Name")
           catSet1.MoveNext
    wend
    catSet1.Close
    set catSet1 = Nothing
%>
<html>
<!--#include virtual="lffc/Adminheader.html"-->
<body>
<b><h5>Logged in as: <%=p_name%></h5>
<p><font size="4" color="#000080">Module Name: <%=p_mod_name%></font></p>
<%
sqlText = "select * " _
          & "from ModuleNotes where " _
          & "Module_ID = " & p_mod_id
   set reviewSet = Mycon.Execute(sqlText)
   while not reviewSet.EOF
%>
<form method="POST" action="a_editmodnotes_list.asp">
<table width="837">
  <TR><TD><h5><INPUT TYPE="hidden" NAME="p_entry_no"
VALUE="<%=reviewSet("Entry_No")%>"></h5></TD></tr>
  <tr><td><h5>Title:<% Response.Write reviewSet("Title") %></h5></td>
<td><h5>Associated Document Name:<% Response.Write reviewSet("DocLink")
%></h5></td>
<td><h5>Display Content on Website:<% Response.Write reviewSet("Display_Doc_Link")
%></h5></td></tr>
<tr><td><input type="submit" value="Edit Note"></td></tr>
</form>
</table>
<%
reviewSet.MoveNext
       wend
       reviewSet.Close
       set reviewSet = Nothing

Mycon.Close
```

```
set Mycon = Nothing
%>
</body>
</html>
```

A_EDITMODNOTES_ACTION.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<%
    Const adOpenKeyset = 1
    Const adLockOptimistic = 3
    Const adCmdText = &H0001

     set Mycon = Server.CreateObject("ADODB.Connection")
    Mycon.Open "lffc"

    p_lect_id = Session("Lecturer")


    set catSet = Mycon.Execute("select * " _
    & "from Admin " _
    & "Where Lecturer_ID = "&p_lect_id)
    while not catSet.EOF
    p_name = catSet("Name")
         catSet.MoveNext
     wend
     catSet.Close
     set catSet = Nothing


    Set Upload = Server.CreateObject("Persits.Upload")
     Count = Upload.Save("c:\lffc\ModuleNotes")

     p_title = cstr(Upload.form("p_title"))
     p_entry_no = cint(Upload.form("p_entry_no"))
    p_display1 = cstr(Upload.form("p_display1"))

    if count = 1 then
    p_filename1 = Upload.Files(1).Path
    p_filename = replace(p_filename1, "c:\lffc\ModuleNotes\", "")
    fileuploaded = true
    else
    fileuploaded = false
    end if

     set Upload = Nothing

    SQL2 = "select * from ModuleNotes where Entry_No=" & p_entry_no

    if fileuploaded = true then
        set myRS2 = Server.CreateObject("ADODB.RecordSet")
         myRS2.Open SQL2, Mycon, adOpenKeyset, adLockOptimistic, adCmdText
         myRS2("Title") = p_title
         myRS2("DocLink") = p_filename
         myRS2("Display_Doc_Link") = p_display1
         myRS2.Update
         myRS2.Close
        Set myRS2 = Nothing
     else
         set myRS3 = Server.CreateObject("ADODB.RecordSet")
         myRS3.Open SQL2, Mycon, adOpenKeyset, adLockOptimistic, adCmdText
         myRS3("Title") = p_title
         myRS3("Display_Doc_Link") = p_display1
         myRS3.Update
         myRS3.Close
        Set myRS3 = Nothing
    end if

        if Err.number = 0 then

            'All is well with the world %>
            <!--#include virtual="lffc/Adminheader.html"-->
```

```
            <H2>Note Edited</H2>

     <b><h5>Logged in as: <%=p_name%></h5><B>
    <%
    Response.Write Count & " file(s) uploaded to c:\lffc\ModuleNotes"
    %>
            Back To<A HREF="admin_main.asp">Main Menu</A>

<%      else %>

            <H2> Problem </H2>
            There was a problem editing the Note
            Please go back and try again
            <A HREF="adminmain.asp">ADD Notes</A>

<%       end if

Mycon.Close
    set Mycon = Nothing
%>
```

A_EDITMODNOTES_LIST.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<%
    if not Session("Auth")="A" then
       Response.Redirect"admin.asp"
    end if

   p_lect_id = Session("Lecturer")
   p_entry_no = Request.form("p_entry_no")

   set Mycon = Server.CreateObject("ADODB.Connection")
   Mycon.Open "lffc"

   set catSet = Mycon.Execute("select * " _
   & "from Admin " _
   & "Where Lecturer_ID = "&p_lect_id)
   while not catSet.EOF
   p_name = catSet("Name")
        catSet.MoveNext
    wend
    catSet.Close
    set catSet = Nothing


%>

<html>
<!--#include virtual="lffc/Adminheader.html"-->
<body>
<b><h5>Logged in as: <%=p_name%></h5><B>
<%
sqlText = "select * " _
          & "from ModuleNotes where " _
          & "Entry_No = " & p_entry_no
   set EditSet = Mycon.Execute(sqlText)
   while not EditSet.EOF
%>
<p><font size="5" color="#000080">Edit Note <% Response.Write EditSet("Title")
%></font></p>
<form method="POST" enctype="multipart/form-data"
action="a_editmodnotes_action.asp">
<table width="837">
  <TR><TD><h5><INPUT TYPE="hidden" NAME="p_entry_no"
VALUE="<%=EditSet("Entry_No")%>"></h5></TD></tr>
 <tr><td width="529"><h5>Title: (Name of hyperlink on website e.g. Lecture 1 - wood
   turning basics)</td><td><input type="text" name="p_title"
VALUE="<%=EditSet("Title")%>"></h5></td></tr>
<td><h5>Upload Associated Document: (Please leave blank if you wish to keep
existing document)</h5></td><td><input type="file" name="p_file"></td></tr>
<tr><td width="458"><h5><b>Display Link to Note on Website: (True means the
   link is displayed)</b></h5></td><td><SELECT NAME="p_displayl"><OPTION
VALUE="<%=EditSet("Display_Doc_Link")%>"><%=EditSet("Display_Doc_Link")%><OPTION
VALUE="False">False<OPTION value="True">True</td></tr>
<tr><td><input type="submit" value="Edit
Note">           Back To<A
HREF="admin_main.asp">Main Menu</A></td></tr>
</form>
</table>
<%
EditSet.MoveNext
       wend
       EditSet.Close
       set EditSet = Nothing

Mycon.Close
set Mycon = Nothing
```

```
%>
</body>
</html>
```

A_REMOVEMODNOTES.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<%
    if not Session("Auth")="A" then
       Response.Redirect"admin.asp"
    end if

   p_lect_id = Session("Lecturer")
   p_mod_id = Request.querystring("p_mod_id")

   set Mycon = Server.CreateObject("ADODB.Connection")
   Mycon.Open "lffc"

   set catSet = Mycon.Execute("select * " _
   & "from Admin " _
   & "Where Lecturer_ID = "&p_lect_id)
   while not catSet.EOF
   p_name = catSet("Name")
        catSet.MoveNext
    wend
    catSet.Close
    set catSet = Nothing


   set catSet1 = Mycon.Execute("select * " _
       & "from Modules " _
       & "Where Module_ID = "&p_mod_id)
   while not catSet1.EOF

   p_mod_name = catset1("Module_Name")
           catSet1.MoveNext
    wend
    catSet1.Close
    set catSet1 = Nothing
%>
<html>
<!--#include virtual="lffc/Adminheader.html"-->
<body>
<b><h5>Logged in as: <%=p_name%></h5>
<p><font size="4" color="#000080">Module Name: <%=p_mod_name%></font></p>
<%
sqlText = "select * " _
          & "from ModuleNotes where " _
          & "Module_ID = " & p_mod_id
    set reviewSet = Mycon.Execute(sqlText)
    while not reviewSet.EOF
%>
<form method="POST" action="a_delmodnotes_action.asp">
<table width="837">
  <TR><TD><h5><INPUT TYPE="hidden" NAME="p_entry_no"
VALUE="<%=reviewSet("Entry_No")%>"></h5></TD></tr>
  <tr><td><h5>Title:<% Response.Write reviewSet("Title") %></h5></td>
<td><h5>Associated Document Name:<% Response.Write reviewSet("DocLink")
%></h5></td>
<td><h5>Display Content on Website:<% Response.Write reviewSet("Display_Doc_Link")
%></h5></td></tr>
<tr><td><input type="submit" value="Delete Note"></td></tr>
</form>
</table>
<%
reviewSet.MoveNext
       wend
       reviewSet.Close
       set reviewSet = Nothing

Mycon.Close
```

```
set Mycon = Nothing
%>
</body>
</html>
```

ADMIN.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<!--#include virtual="lffc/Adminheader.html"-->

     <H2>Lecturer Login</H2>


     <FORM ACTION="Adminlogin_action.asp" METHOD="post">
     Username:  <INPUT TYPE="text" NAME="p_username"><BR>
     Password:  <INPUT TYPE="password" NAME="p_password"><BR>
     <BR>

     <P>
     <INPUT TYPE="submit" VALUE="Log In">
     </FORM>
    <p><a href="forgottenpassword.asp">Have forgotten Your
    password?</a></p>
```

ADMINLOGIN_ACTION.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<!--#include virtual="lffc/Adminheader.html"-->
<%    p_username = Request.form("p_username")
      p_password = Request.form("p_password")


      set Mycon = Server.CreateObject("ADODB.Connection")
      Mycon.Open "lffc"

      sqlText="select * from Admin where Name='"&p_username&"'"
      set userSet = Mycon.Execute(sqlText)
       if userSet.EOF then
            'No such username, so the recordSet is empty
%><BR>
<BR>
<h3><%
            Response.Write "Retry There is problem with your username"
%>
</h3>
            <BR>
            <BR>
            <h4><A HREF="Admin.asp">Retry to Login</A></h4>
<%
      else
            'The username is good, now let's check the password
            real_password = trim(userSet("Password"))
           if p_password = real_password then
           'Password is good now check adminlevel

          if userSet("SuperAdmin") = "True" then
            Session("Auth") = "SA"
            Response.Redirect "sa_admin_main.asp"
           else
            Session("Auth")="A"
            Session("Lecturer")=userSet("Lecturer_ID")
            Response.Redirect "Admin_main.asp"
          end if
          else
           'Username is good, but password is wrong
%><BR>
<BR>
<h3><%
           Response.Write  "Retry there was a problem with your password"
%>
</h3>
          <BR>
          <BR>
          <h4><A HREF="Admin.asp">Retry to Login</A></h4>
<%
       end if
      end if
 %><%  userSet.Close
      set userSet = Nothing

      Mycon.Close
      set Mycon = Nothing
 %></BODY></HTML>
```

COURSE_MODULES.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<!--#include virtual="lffc/Header.html"-->
<%
    p_course_id = Request.querystring("p_course_id")
    p_stage_No = Request.querystring("p_stage_No")

    set Mycon = Server.CreateObject("ADODB.Connection")
    Mycon.Open "lffc"
%>
<head>
<meta http-equiv="Content-Language" content="en-ie">
<title></title>
<link rel="stylesheet" href="mm_training.css" type="text/css" />
</head>
<body bgcolor="#64748B">
<table width="100%" border="0" cellspacing="0" cellpadding="0">
    <tr bgcolor="#D3DCE6">
    <td width="230" colspan="2" valign="top" bgcolor="#26354A"><img
src="webimages\mm_training_photo.jpg" alt="Home Page Image" width="230"
height="286" border="0" /><br />
    <table border="0" cellspacing="0" cellpadding="0" width="230">
        <tr>
        <td width="230" class="sidebarText" id="padding"><br />     </td>
        </tr>
    </table>  </td>
    <td width="50" valign="top"><img src="mm_spacer.gif" alt="" width="50"
height="1" border="0" /></td>
    <td width="440" valign="top"><br />
    <br /> </td>
    <td width="40"> </td>
    <td width="100%"><%
           set catSet1 = Mycon.Execute("select * " _
        & "from Courses " _
        & "Where course_id = "&p_course_id)

 while not catSet1.EOF
 p_Course_Name = catset1("Course_Name")
%>
 <b><H3><%=p_Course_Name%> Stage/Year: <%=p_stage_No%></b></H3>
   <h4>Choose the Module you require from the List below:</h4>
<%   catSet1.MoveNext
    wend
    catSet1.Close
    set catSet1 = Nothing

    sqlText = "select * " _
          & "from CourseModules where " _
          & "Course_ID = " & p_course_id _
       & " AND Stage_ID = " & p_stage_No


   set reviewSet = Mycon.Execute(sqlText)

       while not reviewSet.EOF
  p_mod_id = reviewSet("Module_ID")
  set catset2 = Mycon.Execute("select * " _
  & "from Modules " _
  & "where Module_ID = " &p_mod_id)
   while not catSet2.EOF
%>
<A HREF="module_notes.asp?p_mod2_id=<%=p_mod_id%>&p_module_name=<%
        Response.Write Server.URLencode(catset2("Module_Name"))%>"><H5><b><%=
catSet2("Module_Name") %></b></a></H5>
<%
```

```
                catset2.MoveNext
        wend
        catset2.Close
        set catSet2 = Nothing

        reviewSet.MoveNext
        wend
        reviewSet.Close
        set reviewSet = Nothing


    Mycon.Close
    set Mycon = Nothing

%>

</td>
    </tr>

    <tr bgcolor="#D3DCE6">
    <td colspan="6"><img src="mm_spacer.gif" alt="" width="1" height="1" border="0"
/></td>
    </tr>

    <tr>
    <td width="15"> </td>
    <td width="215"> </td>
    <td width="50"> </td>
    <td width="440"> </td>
    <td width="40"> </td>
    <td width="100%"> </td>
    </tr>
</table>
</body>
</html>
```

COURSE_STAGES.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<!--#include virtual="lffc/Header.html"-->
<%
    p_course_id = Request.querystring("p_course_id")
    p_course_Name = Request.querystring("p_course_Name")

    set Mycon = Server.CreateObject("ADODB.Connection")
    Mycon.Open "lffc"
%>
<head>
<meta http-equiv="Content-Language" content="en-ie">
<title></title>
<link rel="stylesheet" href="mm_training.css" type="text/css" />
</head>
<body bgcolor="#64748B">
<table width="100%" border="0" cellspacing="0" cellpadding="0">
    <tr bgcolor="#D3DCE6">
    <td width="230" colspan="2" valign="top" bgcolor="#26354A"><img
src="webimages\mm_training_photo.jpg" alt="Home Page Image" width="230"
height="286" border="0" /><br />
    <table border="0" cellspacing="0" cellpadding="0" width="230">
        <tr>
        <td width="230" class="sidebarText" id="padding"><br />     </td>
        </tr>
    </table>  </td>
    <td width="50" valign="top"><img src="mm_spacer.gif" alt="" width="50"
height="1" border="0" /></td>
    <td width="440" valign="top"><br />
    <br /> </td>
    <td width="40"> </td>
    <td width="100%"><H2><%=p_Course_Name%></H2>
<%
        set catSet = Mycon.Execute("select * " _
    & "from CourseStages " _
    & "Where course_id = "&p_course_id)
     while not catSet.EOF
 %>
        <A HREF="course_modules.asp?p_course_id=<%=p_course_id%>&p_stage_No=<%
        Response.Write Server.URLencode(catset("Stage_No"))%>"><H3>Stage/Year: <%=
catSet("Stage_No") %></a></H3>
<%
         catSet.MoveNext
     wend
     catSet.Close
     set catSet = Nothing

    p_dis = "True"
    sqlText = "select * " _
           & "from CourseOV where " _
           & "Course_ID = " & p_course_id _
        & " AND Displayov = '" & p_dis & "'"
    set reviewSet = Mycon.Execute(sqlText)
    while not reviewSet.EOF
%>
<A href="/lffc/CourseOverviews/<%Response.write
Server.URLencode(reviewSet("Course_Ov_link"))%>">View The Course OverView</A>
<%
        reviewSet.MoveNext
        wend
        reviewSet.Close
        set reviewSet = Nothing

Mycon.Close
    set Mycon = Nothing
%>
```

```
    </td>
    </tr>

    <tr bgcolor="#D3DCE6">
    <td colspan="6"><img src="mm_spacer.gif" alt="" width="1" height="1" border="0"
/></td>
    </tr>

    <tr>
    <td width="15"> </td>
    <td width="215"> </td>
    <td width="50"> </td>
    <td width="440"> </td>
    <td width="40"> </td>
    <td width="100%"> </td>
    </tr>
</table>
</body>
</html>
```

COURSELINKS.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<!--#include virtual="lffc/Header.html"-->
<head>
<meta http-equiv="Content-Language" content="en-ie">
<title></title>
<link rel="stylesheet" href="mm_training.css" type="text/css" />
</head>
<%
    set Mycon = Server.CreateObject("ADODB.Connection")
    Mycon.Open "lffc"
%>
<body bgcolor="#64748B">
<table width="100%" border="0" cellspacing="0" cellpadding="0">
   <tr bgcolor="#D3DCE6">
   <td width="230" colspan="2" valign="top" bgcolor="#26354A"><img
src="webimages\mm_training_photo.jpg" alt="Home Page Image" width="230"
height="286" border="0" /><br />
   <table border="0" cellspacing="0" cellpadding="0" width="230">
      <tr>
      <td width="230" class="sidebarText" id="padding"><br />     </td>
      </tr>
   </table>  </td>
   <td width="50" valign="top"><img src="mm_spacer.gif" alt="" width="50"
height="1" border="0" /></td>
   <td width="440" valign="top"><br />
   <br /> </td>
   <td width="40"> </td>
   <td width="100%"><b><H4>Please choose your course from the list Below
:</b></H4><br>
   <%
   set catSet = Mycon.Execute("select * from Courses")
    while not catSet.EOF
    %>
<br>
   <A HREF="course_stages.asp?p_course_id=<%
      Response.Write catset("Course_ID") %>&p_course_Name=<%
      Response.Write Server.URLencode(catset("Course_Name"))%>"><font size="3">
    <%= catSet("Course_Name") %></a>
<BR>
   <%
       catSet.MoveNext
    wend
    catSet.Close
    set catSet = Nothing

    Mycon.Close
   set Mycon = Nothing
%>


   </td>
   </tr>

   <tr bgcolor="#D3DCE6">
   <td colspan="6"><img src="mm_spacer.gif" alt="" width="1" height="1" border="0"
/></td>
   </tr>

   <tr>
   <td width="15"> </td>
   <td width="215"> </td>
   <td width="50"> </td>
   <td width="440"> </td>
   <td width="40"> </td>
   <td width="100%"> </td>
```

```
    </tr>
</table>
</body>
</html>
```

MAIN.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<!--#include virtual="lffc/Header.html"-->
<head>
<meta http-equiv="Content-Language" content="en-ie">
<title></title>
<link rel="stylesheet" href="mm_training.css" type="text/css" />
</head>
<body bgcolor="#64748B">
<table width="100%" border="0" cellspacing="0" cellpadding="0">
    <tr bgcolor="#D3DCE6">
    <td width="230" colspan="2" valign="top" bgcolor="#26354A"><img
src="webimages\mm_training_photo.jpg" alt="Home Page Image" width="230"
height="286" border="0" /><br />
    <table border="0" cellspacing="0" cellpadding="0" width="230">
        <tr>
        <td width="230" class="sidebarText" id="padding"><br />     </td>
        </tr>
    </table>  </td>
    <td width="50" valign="top"><img src="mm_spacer.gif" alt="" width="50"
height="1" border="0" /></td>
    <td width="4" valign="top"><br />
    <br /> </td>
    <td width="317">WELCOME MESSAGE TO Go HERE</td>
    <td width="40%"><!--#include virtual="lffc/FrontpageInfo.html"--></td>
    </tr>

    <tr bgcolor="#D3DCE6">
    <td colspan="6"><img src="mm_spacer.gif" alt="" width="1" height="1" border="0"
/></td>
    </tr>

    <tr>
    <td width="15"> </td>
    <td width="215"> </td>
    <td width="50"> </td>
    <td width="4"> </td>
    <td width="317"> </td>
    <td width="40%"> </td>
    </tr>
</table>
</body>
</html>
```

MODULE_NOTES.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<!--#include virtual="lffc/Header.html"-->
<%
    p_module_name = Request.querystring("p_module_name")
    p_mod_id = Request.querystring("p_mod2_id")

    set Mycon = Server.CreateObject("ADODB.Connection")
    Mycon.Open "lffc"
%>
<head>
<meta http-equiv="Content-Language" content="en-ie">
<title></title>
<link rel="stylesheet" href="mm_training.css" type="text/css" />
</head>
<body bgcolor="#64748B">
<table width="100%" border="0" cellspacing="0" cellpadding="0">
    <tr bgcolor="#D3DCE6">
    <td width="230" colspan="2" valign="top" bgcolor="#26354A"><img
src="webimages\mm_training_photo.jpg" alt="Home Page Image" width="230"
height="286" border="0" /><br />
    <table border="0" cellspacing="0" cellpadding="0" width="230">
        <tr>
        <td width="230" class="sidebarText" id="padding"><br />     </td>
        </tr>
    </table>  </td>
    <td width="50" valign="top"><img src="mm_spacer.gif" alt="" width="50"
height="1" border="0" /></td>
    <td width="440" valign="top"><br />
    <br /> </td>
    <td width="40"> </td>
    <td width="100%"><H3><%=p_module_name%></H3>
<%
     p_dis = "True"
    sqlText = "select * " _
          & "from ModuleNotes where " _
          & "Module_ID = " & p_mod_id _
       & " AND Display_doc_link = '" & p_dis & "'"
    set reviewSet = Mycon.Execute(sqlText)
    while not reviewSet.EOF
%>
<h5><A href="/lffc/ModuleNotes/<%Response.write
Server.URLencode(reviewSet("Doclink"))%>"><%Response.write
reviewSet("Title")%></A></h5>
<%
       reviewSet.MoveNext
       wend
       reviewSet.Close
       set reviewSet = Nothing

Mycon.Close
    set Mycon = Nothing
%>
    </td>
    </tr>

    <tr bgcolor="#D3DCE6">
    <td colspan="6"><img src="mm_spacer.gif" alt="" width="1" height="1" border="0"
/></td>
    </tr>

    <tr>
    <td width="15"> </td>
    <td width="215"> </td>
    <td width="50"> </td>
    <td width="440"> </td>
```

```
    <td width="40"> </td>
    <td width="100%"> </td>
    </tr>
</table>
</body>
</html>
```

SA_ADD_ADMIN_ACTION.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<%
    if not Session("Auth")="SA" then
       Response.Redirect"admin.asp"
    end if

       set Mycon = Server.CreateObject("ADODB.Connection")
       Mycon.Open "lffc"

     p_lectname = cstr(Request.form("p_lectname"))
     p_lectpass = cstr(Request.form("p_lectpass"))
     p_lectemail = cstr(Request.form("p_lectemail"))
     p_sa = cstr(Request.form("p_sa"))

       ID_SQL = "Select max(Lecturer_ID) as maxid from Admin"
       set idSet = Mycon.Execute(ID_SQL)
       p_lect_id = idSet("maxid") + 1
       idSet.Close
       set idSet = Nothing

       SQL = "Insert Into Admin(Lecturer_ID, Name, Password, SuperAdmin,
Email)Values('"&p_lect_id&"', '"&p_lectname&"', '"&p_lectpass&"', '"&p_sa&"',
'"&p_lectemail&"')"
       Mycon.Execute(SQL)


       if Err.number = 0 then

          'All is well with the world %>
          <!--#include virtual="lffc/sa_Adminheader.html"-->

          <H2>User Added to the Database</H2>

          Back To<A HREF="sa_admin_main.asp">Main Menu</A>


<%     else %>

          <H2> Problem </H2>
          There was a problem adding the User
          Please go back and try again
          <A HREF="sa_createadmin.asp">ADD Admin/Lecturer</A>

<%      end if
    Mycon.Close
    set Mycon = Nothing

%>
```

SA_ADD_COURSE_ACTION.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<%
    if not Session("Auth")="SA" then
       Response.Redirect"admin.asp"
    end if

       set Mycon = Server.CreateObject("ADODB.Connection")
       Mycon.Open "lffc"

    Set Upload = Server.CreateObject("Persits.Upload")
     Count = Upload.Save("c:\lffc\CourseOverviews")

     p_coursename = cstr(Upload.form("p_coursename"))
     p_stage1 = cstr(Upload.form("p_stage1"))
    p_stage2 = cstr(Upload.form("p_stage2"))
    p_stage3 = cstr(Upload.form("p_stage3"))
    p_stage4 = cstr(Upload.form("p_stage4"))
    p_display = cstr(Upload.form("p_display"))

    if count = 1 then
    p_filename1 = Upload.Files(1).Path
    p_filename = replace(p_filename1, "c:\lffc\CourseOverviews\", "")
    fileuploaded = true
    else
    fileuploaded = false
    end if

     set Upload = Nothing

       ID_SQL = "Select max(Course_ID) as maxid from Courses"
       set idSet = Mycon.Execute(ID_SQL)
       p_course_id = idSet("maxid") + 1
       idSet.Close
       set idSet = Nothing

       SQL = "Insert Into Courses(Course_ID, Course_Name)Values('"&p_course_id&"',
'"&p_coursename&"')"
       Mycon.Execute(SQL)

     if p_stage1 = "yes" then
     p_stageno = 1
     SQL1 = "Insert Into CourseStages(Course_ID, Stage_No)Values('"&p_course_id&"',
'"&p_stageno&"')"
       Mycon.Execute(SQL1)
    end if

     if p_stage2 = "yes" then
     p_stageno = 2
     SQL2 = "Insert Into CourseStages(Course_ID, Stage_No)Values('"&p_course_id&"',
'"&p_stageno&"')"
       Mycon.Execute(SQL2)
    end if

    if p_stage3 = "yes" then
     p_stageno = 3
     SQL3 = "Insert Into CourseStages(Course_ID, Stage_No)Values('"&p_course_id&"',
'"&p_stageno&"')"
       Mycon.Execute(SQL3)
    end if

        if p_stage4 = "yes" then
    p_stageno = 4
    SQL4 = "Insert Into CourseStages(Course_ID, Stage_No)Values('"&p_course_id&"',
'"&p_stageno&"')"
       Mycon.Execute(SQL4)
```

```
    end if

    if fileuploaded = true then
    SQL5 = "Insert Into CourseOV(Course_ID, Course_OV_link,
Displayov)Values('"&p_course_id&"', '"&p_filename&"', '"&p_display&"')"
   Mycon.Execute(SQL5)
   else
   SQL6 = "Insert Into CourseOV(Course_ID, Displayov)Values('"&p_course_id&"',
'"&p_display&"')"
   Mycon.Execute(SQL6)
   end if


       if Err.number = 0 then

           'All is well with the world %>
           <!--#include virtual="lffc/sa_Adminheader.html"-->

           <H2>Course Added to the Database</H2>
   <%
   Response.Write Count & " file(s) uploaded to c:\lffc\CourseOverviews"
   %>
           Back To<A HREF="sa_admin_main.asp">Main Menu</A>


<%     else %>

           <H2> Problem </H2>
           There was a problem adding the Course
           Please go back and try again
           <A HREF="sa_addcourse.asp">ADD Course</A>

<%      end if
   Mycon.Close
   set Mycon = Nothing

%>
```

SA_ADD_MODULENOTES_ACTION.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<%
    if not Session("Auth")="SA" then
       Response.Redirect"admin.asp"
    end if

    set Mycon = Server.CreateObject("ADODB.Connection")
    Mycon.Open "lffc"

    Set Upload = Server.CreateObject("Persits.Upload")
     Count = Upload.Save("c:\lffc\ModuleNotes")

     p_title = cstr(Upload.form("p_title"))
     p_mod_id = cint(Upload.form("p_mod_id"))
    p_display = cstr(Upload.form("p_display"))

    if count = 1 then
    p_filename1 = Upload.Files(1).Path
    p_filename = replace(p_filename1, "c:\lffc\ModuleNotes\", "")
    fileuploaded = true
    else
    fileuploaded = false
    end if

     set Upload = Nothing

        set catSet = Mycon.Execute("select * " _
    & "from Modules " _
    & "Where Module_ID = "&p_mod_id)
    while not catSet.EOF
    p_mod_name = catSet("Module_Name")
         catSet.MoveNext
     wend
     catSet.Close
     set catSet = Nothing

        ID_SQL = "Select max(Entry_No) as maxid from ModuleNotes"
        set idSet = Mycon.Execute(ID_SQL)
        p_entry_no = idSet("maxid") + 1
        idSet.Close
        set idSet = Nothing


    if fileuploaded = true then
     SQL5 = "Insert Into ModuleNotes(Entry_No, Module_ID, Title, DocLink,
Display_Doc_Link)Values('"&p_entry_no&"', '"&p_mod_id&"', '"&p_title&"',
'"&p_filename&"', '"&p_display&"')"
    Mycon.Execute(SQL5)
    else
    SQL6 = "Insert Into ModuleNotes(Entry_No, Module_ID, Title,
Display_Doc_Link)Values('"&p_entry_no&"', '"&p_mod_id&"', '"&p_title&"',
'"&p_display&"')"
    Mycon.Execute(SQL6)
    end if


        if Err.number = 0 then

            'All is well with the world %>
            <!--#include virtual="lffc/sa_Adminheader.html"-->

            <H2>Note Added to the Database</H2>

     <b><h5>Module  <%=p_mod_name%></h5><B>
    <%
```

```
    Response.Write Count & " file(s) uploaded to c:\lffc\ModuleNotes"
    %>
           Back To<A HREF="sa_updatemodules.asp">Main Menu</A>

<%     else %>

           <H2> Problem </H2>
           There was a problem adding the Note
           Please go back and try again
           <A HREF="sa_updatemodules.asp">ADD Notes</A>

<%      end if
Mycon.Close
    set Mycon = Nothing
%>
```

SA_ADDCOURSE.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<%
    if not Session("Auth")="SA" then
       Response.Redirect"admin.asp"
    end if
%>

<html>
<head><SCRIPT LANGUAGE = "JAVASCRIPT">
    function validate() {
       if (document.form1.p_coursename.value.length < 5) {
          alert("Please enter a name for the course.");
          return false;
          }
          return true;
       }
    </SCRIPT></head>
<!--#include virtual="lffc/sa_Adminheader.html"-->
<body>
<p><font size="5" color="#000080">Add a new Course</font></p>
<form method="POST" enctype="multipart/form-data" name="form1"
action="sa_add_course_action.asp" onSubmit="return validate();">
<p>
  <p>Course Name        <input type="text"
name="p_coursename" size="20"></p>
  <p><font color="#000080">Please Select Which Stage/Year's apply to the
Course:</font></p>
  <p>Stage/Year1:   <INPUT TYPE="checkbox" NAME="p_stage1"
VALUE="yes"></p>
  <p>Stage/Year2:   <INPUT TYPE="checkbox" NAME="p_stage2"
VALUE="yes"></p>
  <p>Stage/Year3:   <INPUT TYPE="checkbox" NAME="p_stage3"
VALUE="yes"></p>
  <p>Stage/Year4:   <INPUT TYPE="checkbox" NAME="p_stage4"
VALUE="yes"></p>
  <p>Upload Course Overview Document: <input type="file" name="p_filename"></p>
  <p>Display Course Overview on Website:<SELECT NAME="p_display"><OPTION
VALUE="False">False<OPTION value="True">True</p>
  <p><input type="submit" value="Add to Database"
name="B1">       
  <input type="reset" value="Clear" name="B2"></p>
</form>
</body>
</html>
```

SA_ADDMODULE.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<%
    if not Session("Auth")="SA" then
       Response.Redirect"admin.asp"
    end if

    set Mycon = Server.CreateObject("ADODB.Connection")
    Mycon.Open "lffc"

%><head><SCRIPT LANGUAGE = "JAVASCRIPT">
    function validate() {
       if (document.form2.p_mod_name.value.length < 5) {
          alert("Please enter a name for the Module.");
          return false;
          }
          return true;
       }
    </SCRIPT></head>

    <!--#include virtual="lffc/sa_Adminheader.html"-->
    <h3><b>Adding a New Module</b></h3>
     <form method="POST" action="sa_addmodule_action.asp" name="form2"
onSubmit="return validate();">
    <TABLE width="788">
    <TR><td><h5>Module Name:</h5></td><td><INPUT TYPE="Text"
NAME="p_mod_name"></TD></TR></TABLE>
<table width="473"><tr><td width="271"><h5><b>
Select a Lecturer for the Module:</h5></b></td><td>
    <SELECT NAME="p_lect_id">
<%
       set catSet2 = Mycon.Execute("select * from Admin")
       while not catSet2.EOF
%>
       <OPTION VALUE="<%=catSet2("Lecturer_ID")%>"><%
          Response.Write catSet2("Name")%> <%
       catSet2.MoveNext
    wend
    catSet2.Close
    set catSet2 = Nothing
%>
    </SELECT>
</td></tr></table>
<table width="785">
    <TR><TD width="202"><INPUT TYPE="submit" VALUE="Add Module to
Database"></form></td><td><A HREF="sa_admin_main.asp">Back TO Main
Menu</A></b></td></TR>
    </TABLE>

<%

      Mycon.Close
      set Mycon = Nothing


%>
```

SA_ADDMODULE_ACTION.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<%
    if not Session("Auth")="SA" then
       Response.Redirect"admin.asp"
    end if

       set Mycon = Server.CreateObject("ADODB.Connection")
       Mycon.Open "lffc"


     p_mod_name = Request.form("p_mod_name")
    p_lect_id = Request.form("p_lect_id")

       ID_SQL = "Select max(Module_ID) as maxid from Modules"
       set idSet = Mycon.Execute(ID_SQL)
       p_mod_id = idSet("maxid") + 1
       idSet.Close
       set idSet = Nothing

       SQL = "Insert Into Modules(Module_ID, Module_Name)Values('"&p_mod_id&"',
'"&p_mod_name&"')"
       Mycon.Execute(SQL)

     SQL2 = "Insert Into ModuleLecturer(Module_ID,
Lecturer_ID)Values('"&p_mod_id&"', '"&p_lect_id&"')"
    Mycon.Execute(SQL2)



       if Err.number = 0 then

          'All is well with the world %>
          <!--#include virtual="lffc/sa_Adminheader.html"-->

          <H2>Module Added to the Database</H2>

          Dont Forget To Associate the New Module with a Course.

          Back To<A HREF="sa_updatemodules.asp">Main Menu</A>


<%     else %>

          <H2> Problem </H2>
          There was a problem adding the Module
          Please go back and try again
          <A HREF="sa_updatemodules.asp">ADD Course</A>

<%      end if
    Mycon.Close
    set Mycon = Nothing

%>
```

SA_ADDMODULENOTES.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<%
    if not Session("Auth")="SA" then
       Response.Redirect"admin.asp"
    end if

   p_mod_id = Request.querystring("p_mod_id")

   set Mycon = Server.CreateObject("ADODB.Connection")
   Mycon.Open "lffc"

   set catSet = Mycon.Execute("select * " _
   & "from Modules " _
   & "Where Module_ID = "&p_mod_id)
   while not catSet.EOF
   p_mod_name = catSet("Module_Name")
        catSet.MoveNext
    wend
    catSet.Close
    set catSet = Nothing


%>

<html>
<!--#include virtual="lffc/sa_Adminheader.html"-->
<body>
<b><h4>Module: <%=p_mod_name%></h4><B>
<p><font size="5" color="#000080">Add Module Notes</font></p>
<form method="POST" enctype="multipart/form-data"
action="sa_add_modulenotes_action.asp">
<table width="837">
  <TR><TD><INPUT TYPE="hidden" NAME="p_mod_id" VALUE="<%=p_mod_id%>"></TD></TR>
  <tr><td width="529">Title: (Name of hyperlink on website e.g. Lecture 1 - wood
    turning basics)</td><td><input type="text" name="p_title" size="31"></td></tr>
  <tr><td width="529">Upload Associated Document:</td><td><input type="file"
name="p_file"></td></tr>
  <tr><td width="529">Display Course Overview on Website:</td><td><SELECT
NAME="p_display"><OPTION VALUE="False">False<OPTION value="True">True</td></tr>
  <tr><td width="529">This determines if a kink to the material appears on the
website,
    False means the link/information will not be visible. This may be changed
    later by editing the Notes</td><td></td></tr>
</table>
  <p><input type="submit" value="Add to Database"
name="B1">        
  <input type="reset" value="Clear" name="B2"></p>
</form>
<%
Mycon.Close
set Mycon = Nothing
%>
</body>
</html>
```

SA_ADMIN_MAIN.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<%
    if not Session("Auth")="SA" then
        Response.Redirect"admin.asp"
    end if
%>
<html>
<!--#include virtual="lffc/sa_Adminheader.html"-->
<center>
<font color="#FF0000">
<b><font size="5">ADMINISTRATION MENU</font></b>
</center>
<table border="1" width="100%" height="195">
    <tr>
        <td height="189" width="415"><h4><b><a href="sa_updatecourses.asp">Update
Courses:</a></h4></B>
        <h4><b><p><a href="sa_updatemodules.asp">Update Modules:</a><h4><b><p><a
href="sa_createadmin.asp">Create
        Admin/Lecturer Account:</a></h4></a></b></td>
        <td height="189" width="555"><h4><b><a href="sa_admininfo.asp">View
Administrators Information:</a></h4></b><p><h4><b>
        <a href="sa_dbtables.asp">View Database Tables:</a></b></h4><p><h4><b>
        <a href="sa_lecturer.asp">Lecturer Account Maintenance:</a></b></h4></td>
    </tr>
</table>

</html>
```

SA_ADMININFO.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<%
    if not Session("Auth")="SA" then
        Response.Redirect"admin.asp"
    end if

    p_table_name = "Admin"

    Const adOpenForwardOnly = 0
    Const adLockOptimistic = 3
    Const adCmdText = &H0001
%>
<!--#include virtual="lffc/sa_Adminheader.html"-->
<H2> </H2>

<TABLE BORDER="1">

<%
    set Mycon = Server.CreateObject("ADODB.Connection")
    Mycon.Open "lffc"

    set tableSet = Server.CreateObject("ADODB.RecordSet")
    tableSet.Open p_table_name, Mycon, adOpenForwardOnly, adLockOptimistic,
adCmdTable
%>
    <TR>
<%      p_numberOfColumns = tableSet.fields.Count
        for x = 0 to (p_numberOfColumns - 1) %>
            <TH><%= tableSet.Fields(x).Name %></TH>
<%      next %>
    </TR>
<% while not tableSet.EOF %>
        <TR>
<%      for each col in tableSet.Fields %>
            <TD><%= col.Value %></TD>
<%          next       %>
        </TR>
<%      tableSet.MoveNext
    wend
    tableSet.Close
%>

</TABLE>
<%
    Mycon.Close
    set Mycon = Nothing
%>
</TABLE>
```

SA_CHANGEMODULENOTES.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<%
    if not Session("Auth")="SA" then
       Response.Redirect"admin.asp"
    end if

   p_mod_id = Request.querystring("p_mod_id")

   set Mycon = Server.CreateObject("ADODB.Connection")
   Mycon.Open "lffc"

   set catSet1 = Mycon.Execute("select * " _
       & "from Modules " _
       & "Where Module_ID = "&p_mod_id)
   while not catSet1.EOF

   p_mod_name = catset1("Module_Name")
           catSet1.MoveNext
    wend
    catSet1.Close
    set catSet1 = Nothing
%>
<html>
<!--#include virtual="lffc/sa_Adminheader.html"-->
<body>
<p><font size="4" color="#000080">Module Name: <%=p_mod_name%></font></p>
<%
sqlText = "select * " _
          & "from ModuleNotes where " _
          & "Module_ID = " & p_mod_id
    set reviewSet = Mycon.Execute(sqlText)
    while not reviewSet.EOF
%>
<form method="POST" action="sa_changemodulenotes_list.asp">
<table width="837">
  <TR><TD><h5><INPUT TYPE="hidden" NAME="p_entry_no"
VALUE="<%=reviewSet("Entry_No")%>"></h5></TD></tr>
  <tr><td><h5>Title:<% Response.Write reviewSet("Title") %></h5></td>
<td><h5>Associated Document Name:<% Response.Write reviewSet("DocLink")
%></h5></td>
<td><h5>Display Content on Website:<% Response.Write reviewSet("Display_Doc_Link")
%></h5></td></tr>
<tr><td><input type="submit" value="Edit Note"></td></tr>
</form>
</table>
<%
reviewSet.MoveNext
       wend
       reviewSet.Close
       set reviewSet = Nothing

Mycon.Close
set Mycon = Nothing
%>
</body>
</html>
```

SA_CHANGEMODULENOTES_ACTION.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<%
    Const adOpenKeyset = 1
    Const adLockOptimistic = 3
    Const adCmdText = &H0001

     set Mycon = Server.CreateObject("ADODB.Connection")
    Mycon.Open "lffc"


     Set Upload = Server.CreateObject("Persits.Upload")
     Count = Upload.Save("c:\lffc\ModuleNotes")

     p_title = cstr(Upload.form("p_title"))
     p_entry_no = cint(Upload.form("p_entry_no"))
    p_display1 = cstr(Upload.form("p_display1"))

    if count = 1 then
    p_filename1 = Upload.Files(1).Path
    p_filename = replace(p_filename1, "c:\lffc\ModuleNotes\", "")
    fileuploaded = true
    else
    fileuploaded = false
    end if

     set Upload = Nothing

    SQL2 = "select * from ModuleNotes where Entry_No=" & p_entry_no

    if fileuploaded = true then
       set myRS2 = Server.CreateObject("ADODB.RecordSet")
        myRS2.Open SQL2, Mycon, adOpenKeyset, adLockOptimistic, adCmdText
        myRS2("Title") = p_title
        myRS2("DocLink") = p_filename
        myRS2("Display_Doc_Link") = p_display1
        myRS2.Update
        myRS2.Close
      Set myRS2 = Nothing
     else
        set myRS3 = Server.CreateObject("ADODB.RecordSet")
        myRS3.Open SQL2, Mycon, adOpenKeyset, adLockOptimistic, adCmdText
        myRS3("Title") = p_title
        myRS3("Display_Doc_Link") = p_display1
        myRS3.Update
        myRS3.Close
      Set myRS3 = Nothing
    end if

       if Err.number = 0 then

          'All is well with the world %>
          <!--#include virtual="lffc/sa_Adminheader.html"-->

          <H2>Note Edited</H2>

     <%
    Response.Write Count & " file(s) uploaded to c:\lffc\ModuleNotes"
    %>
          Back To<A HREF="sa_updatemodules.asp">Main Menu</A>

<%     else %>

          <H2> Problem </H2>
          There was a problem editing the Note
          Please go back and try again
```

```
            <A HREF="sa_updatemodules.asp">ADD Notes</A>

<%       end if

Mycon.Close
    set Mycon = Nothing
%>
```

SA_CHANGEMODULENOTES_LIST

```
<%@ LANGUAGE="VBSCRIPT" %>
<%
    if not Session("Auth")="SA" then
       Response.Redirect"admin.asp"
    end if

   p_entry_no = Request.form("p_entry_no")

   set Mycon = Server.CreateObject("ADODB.Connection")
   Mycon.Open "lffc"

%>

<html>
<!--#include virtual="lffc/sa_Adminheader.html"-->
<body>
<%
sqlText = "select * " _
          & "from ModuleNotes where " _
          & "Entry_No = " & p_entry_no
    set EditSet = Mycon.Execute(sqlText)
    while not EditSet.EOF
%>
<p><font size="5" color="#000080">Edit Note <% Response.Write EditSet("Title")
%></font></p>
<form method="POST" enctype="multipart/form-data"
action="sa_changemodulenotes_action.asp">
<table width="837">
  <TR><TD><h5><INPUT TYPE="hidden" NAME="p_entry_no"
VALUE="<%=EditSet("Entry_No")%>"></h5></TD></tr>
 <tr><td width="529"><h5>Title: (Name of hyperlink on website e.g. Lecture 1 - wood
   turning basics)</td><td><input type="text" name="p_title"
VALUE="<%=EditSet("Title")%>"></h5></td></tr>
<td><h5>Upload Associated Document: (Please leave blank if you wish to keep
existing document)</h5></td><td><input type="file" name="p_file"></td></tr>
<tr><td width="458"><h5><b>Display Link to Note on Website: (True means the
   link is displayed)</b></h5></td><td><SELECT NAME="p_display1"><OPTION
VALUE="<%=EditSet("Display_Doc_Link")%>"><%=EditSet("Display_Doc_Link")%><OPTION
VALUE="False">False<OPTION value="True">True</td></tr>
<tr><td><input type="submit" value="Edit
Note">           Back To<A
HREF="admin_main.asp">Main Menu</A></td></tr>
</form>
</table>
<%
EditSet.MoveNext
       wend
       EditSet.Close
       set EditSet = Nothing

Mycon.Close
set Mycon = Nothing
%>
</body>
</html>
```

SA_COURSEMODULES.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<!--#include virtual="lffc/sa_AdminHeader.html"-->
<%
    if not Session("Auth")="SA" then
       Response.Redirect"admin.asp"
    end if

    set Mycon = Server.CreateObject("ADODB.Connection")
    Mycon.Open "lffc"
%>
<H3>Select the Course you wish to associate the Module with</H3>
<h5>Repeat this process if you wish to associate the module with multiple
courses</h5>
<FORM METHOD="post" ACTION="sa_update_coursemodules.asp">
<BR>
    Select Course: 
    <SELECT NAME="p_course_id">
<%
       set catSet = Mycon.Execute("select * from Courses")
       while not catSet.EOF
%>
       <OPTION VALUE="<%= catSet("Course_ID") %>"><%
          Response.Write catSet("Course_Name")%> <%
       catSet.MoveNext
    wend
    catSet.Close
    set catSet = Nothing
%>
    </SELECT>
<br>
     Select Module:
    <SELECT NAME="p_mod_id">
<%
       set modSet = Mycon.Execute("select * from Modules")
       while not modSet.EOF
%>
       <OPTION VALUE="<%= modSet("Module_ID") %>"><%
          Response.Write modSet("Module_Name")%> <%
       modSet.MoveNext
    wend
    modSet.Close
    set modSet = Nothing
%>
    </SELECT>
    <P>
    <INPUT TYPE="submit" value="Continue">

</FORM>
<%
    Mycon.Close
    set Mycon = Nothing
%></body>
</html>
```

SA_CREATEADMIN.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<%
    if not Session("Auth")="SA" then
       Response.Redirect"admin.asp"
    end if
%>

<html>
<head><SCRIPT LANGUAGE = "JAVASCRIPT">
    function validate() {
       if (document.form3.p_lectname.value.length < 5) {
          alert("Please enter a username for the Lecturer.");
          return false;
          }
       if (document.form3.p_lectpass.value.length < 5) {
          alert("Please enter a Password for the Lecturer.");
          return false;
          }
       if (document.form3.p_lectemail.value.length < 5) {
          alert("Please enter the Lecturer's Email Address.");
          return false;
          }
          return true;
       }
    </SCRIPT></head>
<!--#include virtual="lffc/sa_Adminheader.html"-->
<body>
<p><font size="5" color="#000080">Add a New Admin/Lecturer to the System</font></p>
<form method="POST" action="sa_add_admin_action.asp" name="form3" onSubmit="return
validate();">
<p>
  <p>Name:       
            <input
type="text" name="p_lectname" size="20"></p>
<p>Password:             
<input type="text" name="p_lectpass" size="20"></p>
<p>Email Address:       <input type="text"
name="p_lectemail" size="20"></p>
  <p><font color="#000080">Please Select If Lecturer is a Super
Administrator:</font></p>
  <p>SuperAdmin:   <SELECT NAME="p_sa"><OPTION
VALUE="False">False<OPTION value="TRUE">True</p>
  <p><input type="submit" value="Add to Database"
name="B1">        
  <input type="reset" value="Clear" name="B2"></p>
</form>
</body>
</html>
```

SA_DBTABLES.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<%
    if not Session("Auth")="SA" then
       Response.Redirect"admin.asp"
    end if
%>
<html>
<!--#include virtual="lffc/sa_Adminheader.html"--><html>
<p>List of Tables Contained in The Leterfrack Furniture College Database,<BR>
To view a Table just Pick it from the List:
<FORM ACTION="sa_viewtable_action.asp" METHOD="post">
    <SELECT NAME="p_table_name">
        <OPTION VALUE="Courses">Courses
         <OPTION VALUE="CourseStages">CourseStages
         <OPTION VALUE="Modules">Modules
         <OPTION VALUE="ModuleNotes">ModuleNotes
         <OPTION VALUE="CourseModules">CourseModules
         <OPTION VALUE="ModuleLecturer">ModuleLecturer
         <OPTION VALUE="CourseOV">CourseOV
         </SELECT>
    <P>
    <INPUT TYPE="submit">

</FORM></td>

</body>
</html>
```

SA_DEL_LECTURER.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<!--#include virtual="lffc/sa_AdminHeader.html"-->
<%
    if not Session("Auth")="SA" then
        Response.Redirect"admin.asp"
    end if

    set Mycon = Server.CreateObject("ADODB.Connection")
    Mycon.Open "lffc"
%>
<H3>Select the Lecturer who you wish to Remove from the system</H3>
<FORM ACTION="sa_dellect_action.asp" METHOD="post">
<BR>
    Lecturer:
    <SELECT NAME="p_lect_id">
<%
        set catSet = Mycon.Execute("select * from Admin")
        while not catSet.EOF
%>
        <OPTION VALUE="<%=catSet("Lecturer_ID")%>"><%
            Response.Write catSet("Name")%> <%
        catSet.MoveNext
    wend
    catSet.Close
    set catSet = Nothing
%>
    </SELECT>
    <P>
    <INPUT TYPE="submit" value="Delete Lecturer's Details">

</FORM>
<%
    Mycon.Close
    set Mycon = Nothing
%></body>
</html>
```

SA_DELCOURSE.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<!--#include virtual="lffc/sa_AdminHeader.html"-->
<%
    if not Session("Auth")="SA" then
       Response.Redirect"admin.asp"
    end if

    set Mycon = Server.CreateObject("ADODB.Connection")
    Mycon.Open "lffc"
%>
<H3>Select the Course You wish to Delete</H3>
<FORM ACTION="sa_delcourse_action.asp" METHOD="post">
<BR>
    Course:
    <SELECT NAME="p_course_id">
<%
       set catSet = Mycon.Execute("select * from Courses")
       while not catSet.EOF
%>
       <OPTION VALUE="<%= catSet("Course_ID") %>"><%
          Response.Write catSet("Course_Name")%> <%
       catSet.MoveNext
    wend
    catSet.Close
    set catSet = Nothing
%>
    </SELECT>
    <P>
    <INPUT TYPE="submit" value="Delete Course">

</FORM>
<%
    Mycon.Close
    set Mycon = Nothing
%></body>
</html>
```

SA_DELCOURSE_ACTION.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<%
    if not Session("Auth")="A" then
       Response.Redirect"admin.asp"
    end if

    p_lect_id = Session("Lecturer")

    set Mycon = Server.CreateObject("ADODB.Connection")
    Mycon.Open "lffc"

    set catSet = Mycon.Execute("select * " _
    & "from Admin " _
    & "Where Lecturer_ID = "&p_lect_id)
    while not catSet.EOF
    p_name = catSet("Name")
         catSet.MoveNext
     wend
     catSet.Close
     set catSet = Nothing


    Set Upload = Server.CreateObject("Persits.Upload")
     Count = Upload.Save("c:\lffc\ModuleNotes")

     p_title = cstr(Upload.form("p_title"))
     p_mod_id = cint(Upload.form("p_mod_id"))
    p_display = cstr(Upload.form("p_display"))

    if count = 1 then
    p_filename1 = Upload.Files(1).Path
    p_filename = replace(p_filename1, "c:\lffc\ModuleNotes\", "")
    fileuploaded = true
    else
    fileuploaded = false
    end if

     set Upload = Nothing

       ID_SQL = "Select max(Entry_No) as maxid from ModuleNotes"
       set idSet = Mycon.Execute(ID_SQL)
       p_entry_no = idSet("maxid") + 1
       idSet.Close
       set idSet = Nothing


    if fileuploaded = true then
     SQL5 = "Insert Into ModuleNotes(Entry_No, Module_ID, Title, DocLink,
Display_Doc_Link)Values('"&p_entry_no&"', '"&p_mod_id&"', '"&p_title&"',
'"&p_filename&"', '"&p_display&"')"
    Mycon.Execute(SQL5)
    else
    SQL6 = "Insert Into ModuleNotes(Entry_No, Module_ID, Title,
Display_Doc_Link)Values('"&p_entry_no&"', '"&p_mod_id&"', '"&p_title&"',
'"&p_display&"')"
    Mycon.Execute(SQL6)
    end if


       if Err.number = 0 then

          'All is well with the world %>
          <!--#include virtual="lffc/sa_Adminheader.html"-->

          <H2>Note Added to the Database</H2>
```

```
    <b><h5>Logged in as: <%=p_name%></h5><B>
    <%
    Response.Write Count & " file(s) uploaded to c:\lffc\ModuleNotes"
    %>
           Back To<A HREF="admin_main.asp">Main Menu</A>

<%     else %>

           <H2> Problem </H2>
           There was a problem adding the Note
           Please go back and try again
           <A HREF="a_addmodnotes.asp">ADD Notes</A>

<%      end if
Mycon.Close
    set Mycon = Nothing
%>
```

SA_DELETEMODULE.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<!--#include virtual="lffc/sa_AdminHeader.html"-->
<%
    if not Session("Auth")="SA" then
        Response.Redirect"admin.asp"
    end if

    set Mycon = Server.CreateObject("ADODB.Connection")
    Mycon.Open "lffc"
%>
<H3>Select the Module You wish to Delete</H3>
<FORM ACTION="sa_delmodule_action.asp" METHOD="post">
<BR>
    Course:
    <SELECT NAME="p_mod_id">
<%
        set catSet = Mycon.Execute("select * from Modules")
        while not catSet.EOF
%>
        <OPTION VALUE="<%= catSet("Module_ID") %>"><%
            Response.Write catSet("Module_Name")%> <%
        catSet.MoveNext
    wend
    catSet.Close
    set catSet = Nothing
%>
    </SELECT>
    <P>
    <INPUT TYPE="submit" value="Delete Module">

</FORM>
<%
    Mycon.Close
    set Mycon = Nothing
%></body>
</html>
```

SA_DELLECT_ACTION.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<!--#include virtual="lffc/sa_adminheader.html"-->
<%
    if not Session("Auth")="SA" then
       Response.Redirect"admin.asp"
    end if

     p_lect_id = Request.form("p_lect_id")

    set Mycon = Server.CreateObject("ADODB.Connection")
    Mycon.Open "lffc"

    mySQL = "DELETE FROM Admin WHERE Lecturer_ID = " &p_lect_id
     Mycon.Execute(mySQL)

      Mycon.Close
      set Mycon = Nothing
%>
<P>Lecturer's Information Has been Deleted <BR></P>
<P><a href="sa_admin_main.asp">Return To Main Menu</a></P>
</BODY>
</HTML>
```

SA_DELMODULE_ACTION.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<!--#include virtual="lffc/sa_adminheader.html"-->
<%
    if not Session("Auth")="SA" then
       Response.Redirect"admin.asp"
    end if

     p_mod_id = Request.form("p_mod_id")

    set Mycon = Server.CreateObject("ADODB.Connection")
    Mycon.Open "lffc"


    mySQL = "DELETE FROM Modules WHERE Module_ID = " &p_mod_id
     Mycon.Execute(mySQL)

     mySQL2 = "DELETE FROM ModuleLecturer WHERE Module_ID = " &p_mod_id
     Mycon.Execute(mySQL2)

     mySQL3 = "DELETE FROM CourseModules WHERE Module_ID = " &p_mod_id
     Mycon.Execute(mySQL3)


      Mycon.Close
      set Mycon = Nothing
%>
<P> Module Has been Deleted <BR></P>
<P><a href="sa_admin_main.asp">Return To Main Menu</a></P>
</BODY>
</HTML>
```

SA_EDITCOURSE.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<!--#include virtual="lffc/sa_AdminHeader.html"-->
<%
    if not Session("Auth")="SA" then
       Response.Redirect"admin.asp"
    end if

    set Mycon = Server.CreateObject("ADODB.Connection")
    Mycon.Open "lffc"
%>
<H3>Select the Course You wish to Edit</H3>
<FORM ACTION="sa_update_course.asp" METHOD="post">
<BR>
    Course:
    <SELECT NAME="p_course_id">
<%
       set catSet = Mycon.Execute("select * from Courses")
       while not catSet.EOF
%>
       <OPTION VALUE="<%= catSet("Course_ID") %>"><%
          Response.Write catSet("Course_Name")%> <%
       catSet.MoveNext
    wend
    catSet.Close
    set catSet = Nothing
%>
    </SELECT>
    <P>
    <INPUT TYPE="submit" value="Get Course Details">

</FORM>
<%
    Mycon.Close
    set Mycon = Nothing
%></body>
</html>
```

SA_EDITMODULE.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<!--#include virtual="lffc/sa_AdminHeader.html"-->
<%
    if not Session("Auth")="SA" then
       Response.Redirect"admin.asp"
    end if

    set Mycon = Server.CreateObject("ADODB.Connection")
    Mycon.Open "lffc"
%>
<H3>Select the Module whos Name or Lecturer You wish to Edit</H3>
<FORM ACTION="sa_editmoduledetails.asp" METHOD="post">
<BR>
    Module:
    <SELECT NAME="p_mod_id">
<%
       set catSet = Mycon.Execute("select * from Modules")
       while not catSet.EOF
%>
       <OPTION VALUE="<%=catSet("Module_ID")%>"><%
          Response.Write catSet("Module_Name")%> <%
       catSet.MoveNext
    wend
    catSet.Close
    set catSet = Nothing
%>
    </SELECT>
    <P>
    <INPUT TYPE="submit" value="Get Module Details">

</FORM>
<%
    Mycon.Close
    set Mycon = Nothing
%></body>
</html>
```

SA_EDITMODULEDETAILS.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<%
    if not Session("Auth")="SA" then
       Response.Redirect"admin.asp"
    end if

     Const adOpenKeyset = 1
    Const adLockOptimistic = 3
    Const adCmdText = &H0001

    set Mycon = Server.CreateObject("ADODB.Connection")
    Mycon.Open "lffc"

    p_mod_id = Request.Form("p_mod_id")

   set catSet = Mycon.Execute("select * " _
   & "from Modules " _
   & "Where Module_ID = "&p_mod_id)
    while not catSet.EOF
%>
     <!--#include virtual="lffc/sa_Adminheader.html"-->
    <h3><b>Modules Details:<%=catSet("Module_Name")%></b></h3>
     <form method="POST" action="sa_editmoduledetails_action.asp">
    <TABLE width="788">
    <TR><TD width="438"><INPUT TYPE="hidden" NAME="p_mod_ID"
VALUE="<%=catSet("Module_ID")%>"></TD></TR>
    <TR><TD width="438"><h5><b>Name:</h5></b></TD><TD>
       <INPUT TYPE="text" NAME="p_name" VALUE="<%=catSet("Module_Name")%>"
size="47"></TD></TR>
    <TR><td>
<%
strSQL2 = "SELECT * FROM ModuleLecturer Where Module_ID = " &p_mod_id
 set currentlect = Mycon.Execute(strSQL2)
   while not currentlect.EOF
   p_clect_id = currentlect("Lecturer_ID")
           currentlect.MoveNext
    wend
   currentlect.Close
   set currentlect = Nothing
strSQL3 = "SELECT * FROM Admin Where Lecturer_ID = " &p_clect_id
 set displect = Mycon.Execute(strSQL3)
   while not displect.EOF
   p_current_id = displect("Lecturer_ID")
   p_current_name = displect("Name")
           displect.MoveNext
    wend
   displect.Close
   set displect = Nothing

%>
    Module Lecturer:</TD><td>
    <SELECT NAME="p_lect_id">
    <OPTION VALUE="<%= p_current_id %>"><%= p_current_name %>
<%
       set catSet1 = Mycon.Execute("select * from Admin")
          while not catSet1.EOF
 %>
       <OPTION VALUE="<%=catSet1("Lecturer_ID")%>"><%
          Response.Write catSet1("Name")%> <%
       catSet1.MoveNext
    wend
    catSet1.Close
    set catSet1 = Nothing
%>
    </SELECT></td>
```

```
    <TR><TD width="438"><INPUT TYPE="submit" VALUE="Update Modules
Details"></form>   
        <b><A HREF="sa_updatemodules.asp"><font size="4" color="#000000">Back TO Main
Menu</font></A></b></td></TR>
    </TABLE>
<%
    catSet.MoveNext
    wend
    catSet.Close
    set catSet = Nothing

     Mycon.Close
     set Mycon = Nothing
%>
```

SA_EDITMODULEDETAILS_ACTION.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<%
    if not Session("Auth")="SA" then
       Response.Redirect"admin.asp"
    end if

    Const adOpenKeyset = 1
    Const adLockOptimistic = 3
    Const adCmdText = &H0001

    p_mod_id = Request.Form("p_mod_id")

    set Mycon = Server.CreateObject("ADODB.Connection")
    Mycon.Open "lffc"

    mySQL = "select * from Modules where Module_ID= " & p_mod_id

    set myRS = Server.CreateObject("ADODB.RecordSet")
    myRS.Open mySQL, Mycon, adOpenKeyset, adLockOptimistic, adCmdText
    myRS("Module_Name") = Request.form("p_name")
    myRS.Update

    myRS.Close
    Set myRS = Nothing

    mySQL2 = "select * from ModuleLecturer where Module_ID= " & p_mod_id
    set myRS2 = Server.CreateObject("ADODB.RecordSet")
    myRS2.Open mySQL2, Mycon, adOpenKeyset, adLockOptimistic, adCmdText
    myRS2("Lecturer_ID") = Request.form("p_lect_id")
    myRS2.Update

    myRS2.Close
    Set myRS2 = Nothing

    Mycon.Close
    set Mycon = Nothing
%>
<!--#include virtual="lffc/sa_Adminheader.html"-->
<P> Modules Details Updated <BR></P>
<b><A HREF="sa_admin_main.asp"><font size="4" color="#000000">Back TO Main
Menu</font></A>
</BODY>
</HTML>
```

SA_EDITMODULENOTES.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<!--#include virtual="lffc/sa_AdminHeader.html"-->
<%
    if not Session("Auth")="SA" then
       Response.Redirect"admin.asp"
    end if

    set Mycon = Server.CreateObject("ADODB.Connection")
    Mycon.Open "lffc"
%>
<H3>Select the Module whose Notes you Wish to Edit</H3>
<FORM METHOD="post" ACTION="sa_editmodulenotesmain.asp">
<BR>
    Course:
    <SELECT NAME="p_mod_id">
<%
       set catSet = Mycon.Execute("select * from Modules")
       while not catSet.EOF
%>
       <OPTION VALUE="<%= catSet("Module_ID") %>"><%
          Response.Write catSet("Module_Name")%> <%
       catSet.MoveNext
    wend
    catSet.Close
    set catSet = Nothing
%>
    </SELECT>
    <P>
    <INPUT TYPE="submit" value="Get Module Update Menu">

</FORM>
<%
    Mycon.Close
    set Mycon = Nothing
%></body>
</html>
```

SA_EDITMODULENOTESMAIN.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<%
    if not Session("Auth")="SA" then
       Response.Redirect"admin.asp"
    end if
    p_lect_id = Session("Lecturer")

     set Mycon = Server.CreateObject("ADODB.Connection")
     Mycon.Open "lffc"

%>
<html>
<!--#include virtual="lffc/sa_Adminheader.html"-->
<body>
<table border="0" width="100%" height="82">
    <tr>
       <td height="82" width="100%"><b><center><h3><font color="#FF0000">Edit Module
Notes Menu</b></h3>
</center></td>
    </tr>
</table>
<%
         p_mod_id = Request.Form("p_mod_id")
          set catset2 = Mycon.Execute("select * " _
          & "from Modules " _
          & "where Module_ID = " &p_mod_id)
          while not catSet2.EOF
%>
<table border="1" width="984" height="120">
    <tr>
       <td height="0" width="509">
<h4><%= catSet2("Module_Name") %></h4>
</td>
       <td height="0" width="0"><h5><font color="#000066"><a
href="sa_addmodulenotes.asp?p_mod_id=<%=p_mod_id%>">Add New
Notes:</a></font></h5><p>
       <h5><font color="#000066"><a
href="sa_removemodulenotes.asp?p_mod_id=<%=p_mod_id%>">Remove
Notes:</a></font></h5></p>
       <p><h5><font color="#000066"><a
href="sa_changemodulenotes.asp?p_mod_id=<%=p_mod_id%>">Edit
Notes:</a></font></h5></td>
    </tr>
</table>
<%

                catset2.MoveNext
       wend
       catset2.Close
       set catSet2 = Nothing

    Mycon.Close
    set Mycon = Nothing

%>
</body>
</html>
```

SA_LECTURER.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<%
   if not Session("Auth")="SA" then
      Response.Redirect"admin.asp"
   end if

%>

<html>
<!--#include virtual="lffc/sa_Adminheader.html"-->
<p><b><font color="#0000FF" size="5">Lecturer Account Maintenance</font></b></p>
<p>
<H3><b><a href="sa_reset_admin.asp">Update an Exsisting Lecturer's
Information</a></b></H3>
</p>
<p>
<H3><b><a href="sa_del_lecturer.asp">Remove a Lecturer from the System</a></b></H3>
</p>
<p>
</p>
</html>
```

SA_REMOVEMODULENOTES.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<%
    if not Session("Auth")="SA" then
       Response.Redirect"admin.asp"
    end if

   p_mod_id = Request.querystring("p_mod_id")

   set Mycon = Server.CreateObject("ADODB.Connection")
   Mycon.Open "lffc"

   set catSet1 = Mycon.Execute("select * " _
       & "from Modules " _
       & "Where Module_ID = "&p_mod_id)
   while not catSet1.EOF

   p_mod_name = catset1("Module_Name")
           catSet1.MoveNext
    wend
    catSet1.Close
    set catSet1 = Nothing
%>
<html>
<!--#include virtual="lffc/sa_Adminheader.html"-->
<body>
<p><font size="4" color="#000080">Module Name: <%=p_mod_name%></font></p>
<%
sqlText = "select * " _
          & "from ModuleNotes where " _
          & "Module_ID = " & p_mod_id
    set reviewSet = Mycon.Execute(sqlText)
    while not reviewSet.EOF
%>
<form method="POST" action="sa_removemodulenotes_action.asp">
<table width="837">
  <TR><TD><h5><INPUT TYPE="hidden" NAME="p_entry_no"
VALUE="<%=reviewSet("Entry_No")%>"></h5></TD></tr>
  <tr><td><h5>Title:<% Response.Write reviewSet("Title") %></h5></td>
<td><h5>Associated Document Name:<% Response.Write reviewSet("DocLink")
%></h5></td>
<td><h5>Display Content on Website:<% Response.Write reviewSet("Display_Doc_Link")
%></h5></td></tr>
<tr><td><input type="submit" value="Delete Note"></td></tr>
</form>
</table>
<%
reviewSet.MoveNext
       wend
       reviewSet.Close
       set reviewSet = Nothing

Mycon.Close
set Mycon = Nothing
%>
</body>
</html>
```

SA_REMOVEMODULENOTES_ACTION.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<!--#include virtual="lffc/sa_adminheader.html"-->
<%
    if not Session("Auth")="SA" then
       Response.Redirect"admin.asp"
    end if

   p_entry_no = Request.Form("p_entry_no")

   set Mycon = Server.CreateObject("ADODB.Connection")
   Mycon.Open "lffc"

    mySQL = "DELETE FROM ModuleNotes WHERE Entry_No = " &p_entry_no
     Mycon.Execute(mySQL)

      Mycon.Close
      set Mycon = Nothing
%>
<P><h4>Note Deleted</H4></P>
<P><a href="sa_updatemodules.asp">Return To Main Menu</a></P>
</BODY>
</HTML>
```

SA_RESET_ADMIN.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<!--#include virtual="lffc/sa_AdminHeader.html"-->
<%
    if not Session("Auth")="SA" then
        Response.Redirect"admin.asp"
    end if

    set Mycon = Server.CreateObject("ADODB.Connection")
    Mycon.Open "lffc"
%>
<H3>Select the Lecturer whos Information You wish to Edit</H3>
<FORM ACTION="sa_update_lect.asp" METHOD="post">
<BR>
    Lecturer:
    <SELECT NAME="p_lect_id">
<%
        set catSet = Mycon.Execute("select * from Admin")
        while not catSet.EOF
%>
        <OPTION VALUE="<%=catSet("Lecturer_ID")%>"><%
            Response.Write catSet("Name")%> <%
        catSet.MoveNext
    wend
    catSet.Close
    set catSet = Nothing
%>
    </SELECT>
    <P>
    <INPUT TYPE="submit" value="Get Lecturer's Details">

</FORM>
<%
    Mycon.Close
    set Mycon = Nothing
%></body>
</html>
```

SA_UPDATE_COURSE.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<%
    if not Session("Auth")="SA" then
       Response.Redirect"admin.asp"
    end if

    set Mycon = Server.CreateObject("ADODB.Connection")
    Mycon.Open "lffc"

    p_course_id = Request.Form("p_course_id")

    strSQL = "SELECT * FROM Courses WHERE Course_ID = " & p_course_id
    strSQL1 = "SELECT * FROM CourseStages WHERE Course_ID = " &p_course_id
    strSQL2 = "SELECT * FROM CourseOV Where Course_ID = " &p_course_id

    set rstDBEdit = Mycon.Execute(strSQL)

    while not rstDBEdit.EOF
    %>
    <!--#include virtual="lffc/sa_Adminheader.html"-->
    <h3><b>Updating Course: <%= rstDBEdit("Course_Name") %></b></h3>
     <form method="POST" enctype="multipart/form-data"
action="sa_update_course_action.asp">
    <TABLE width="788">
    <TR><TD width="203"><INPUT TYPE="hidden" NAME="Course_ID" VALUE="<%=
rstDBEdit("Course_ID") %>"></TD></TR>
    <TR><TD width="203"><h5><b>Course Name:</h5></b></TD><TD width="575">
       <INPUT TYPE="text" size="44" NAME="Course_Name" VALUE="<%=
rstDBEdit("Course_Name") %>"></TD size="39"></TR>
     </table>
<table><tr><td width="205"><h5><b>Please Select Which Stage/Year's you wish to
apply to the Course from now on:</h5></b></td>
<%
p_stage1 = 0
p_stage2 = 0
p_stage3 = 0
p_stage4 = 0

set catSet = Mycon.Execute(strSQL1)
     while not catSet.EOF

     if catSet("Stage_No") = 1 then
     p_stage1 = 1
     end if

     if catSet("Stage_No") = 2 then
     p_stage2 = 2
     end if

     if catSet("Stage_No") = 3 then
     p_stage3 = 3
     end if

     if catSet("Stage_No") = 4 then
     p_stage4 = 4
     end if

         catSet.MoveNext
     wend
     catSet.Close
     set catSet = Nothing

    if p_stage1 = 1 then
        %><td width="101"><h5><b>Stage/Year1:<INPUT TYPE="checkbox" NAME="p_stage1"
VALUE="yes" checked><br><%
```

```
        else
        %><td><h5><b>Stage/Year1:<INPUT TYPE="checkbox" NAME="p_stage1"
VALUE="yes"><br><%
     end if

    if p_stage2 = 2 then
        %><td><h5><b>Stage/Year2:<INPUT TYPE="checkbox" NAME="p_stage2" VALUE="yes"
checked><br><%
        else
        %><td><h5><b>Stage/Year2:<INPUT TYPE="checkbox" NAME="p_stage2"
VALUE="yes"><br><%
     end if

     if p_stage3 = 3 then
        %><td><h5><b>Stage/Year3:<INPUT TYPE="checkbox" NAME="p_stage3" VALUE="yes"
checked><br><%
        else
        %><td><h5><b>Stage/Year3:<INPUT TYPE="checkbox" NAME="p_stage3"
VALUE="yes"><br><%
     end if

    if p_stage4 = 4 then
        %><td><h5><b>Stage/Year4:<INPUT TYPE="checkbox" NAME="p_stage4" VALUE="yes"
checked><br><%
        else
        %><td><h5><b>Stage/Year4:<INPUT TYPE="checkbox" NAME="p_stage4"
VALUE="yes"><br><%
     end if

%>
</tr></table><table><tr>
<td><h5><b>Upload Course Overview Document: (Please leave blank if you wish to keep
existing document)</h5></b></td>
<td width="236"><input type="file" name="p_filename"></td></tr>


<% set displaySet = Mycon.Execute(strSQL2)
   while not displaySet.EOF
   p_display = displaySet("Displayov")
           displaySet.MoveNext
    wend
   displaySet.Close
   set displaySet = Nothing
%>
  <tr><td width="205"><h5><b>Display Course Overview on Website: (True Means Link
will be Displayed)</h5></b></td>
   <td width="236"><SELECT NAME="p_display1"><OPTION VALUE="<%= p_display %>"><%=
p_display %><OPTION VALUE="False">False<OPTION value="True">True</td></tr>
</table>
<table width="785">
   <TR><TD width="202"><INPUT TYPE="submit" VALUE="Update
Course"></form></td><td><A HREF="sa_admin_main.asp">Back TO Main
Menu</A></b></td></TR>
   </TABLE>

<%

   rstDBEdit.movenext
   wend
   rstDBEdit.Close
   set rstDBEdit = Nothing

     Mycon.Close
     set Mycon = Nothing


%>
```

SA_UPDATE_COURSE_ACTION.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<%
    if not Session("Auth")="SA" then
       Response.Redirect"admin.asp"
    end if

    Const adOpenKeyset = 1
    Const adLockOptimistic = 3
    Const adCmdText = &H0001

    set Mycon = Server.CreateObject("ADODB.Connection")
       Mycon.Open "lffc"

    Set Upload = Server.CreateObject("Persits.Upload")
     Count = Upload.Save("c:\lffc\CourseOverviews")

     p_course_id = cint(Upload.form("Course_ID"))
     p_coursename = cstr(Upload.form("Course_Name"))
     p_stage1 = cstr(Upload.form("p_stage1"))
    p_stage2 = cstr(Upload.form("p_stage2"))
    p_stage3 = cstr(Upload.form("p_stage3"))
    p_stage4 = cstr(Upload.form("p_stage4"))
    p_display1 = cstr(Upload.form("p_display1"))

    if count = 1 then
    p_filename1 = Upload.Files(1).Path
    p_filename = replace(p_filename1, "c:\lffc\CourseOverviews\", "")
    fileuploaded = true
    else
    fileuploaded = false
    end if

     set Upload = Nothing

     mySQL = "select * from Courses where Course_ID= " & p_course_id

     set myRS = Server.CreateObject("ADODB.RecordSet")
     myRS.Open mySQL, Mycon, adOpenKeyset, adLockOptimistic, adCmdText
     myRS("Course_Name") = p_coursename
    myRS.Update

     myRS.Close
    Set myRS = Nothing

    SQLCOV = "select * from CourseOV where Course_ID=" & p_course_id

    set myRS1 = Server.CreateObject("ADODB.RecordSet")
    myRS1.Open SQLCOV, Mycon, adOpenKeyset, adLockOptimistic, adCmdText
    myRS1("Displayov") = p_display1
    myRS1.Update

    myRS1.Close
    Set myRS1 = Nothing

    SQLCOV2 = "select * from CourseOV where Course_ID=" & p_course_id

    if fileuploaded = true then
        set myRS2 = Server.CreateObject("ADODB.RecordSet")
        myRS2.Open SQLCOV2, Mycon, adOpenKeyset, adLockOptimistic, adCmdText
        myRS2("Course_Ov_link") = p_filename
        myRS2.Update

       myRS2.Close
      Set myRS2 = Nothing
    end if
```

```
    removeSet = "DELETE FROM CourseStages WHERE Course_ID = " & p_course_id
     Mycon.Execute(removeSet)

     if p_stage1 = "yes" then
     p_stageno = 1
     SQL1 = "Insert Into CourseStages(Course_ID, Stage_No)Values('"&p_course_id&"',
'"&p_stageno&"')"
        Mycon.Execute(SQL1)
     end if

     if p_stage2 = "yes" then
     p_stageno = 2
     SQL2 = "Insert Into CourseStages(Course_ID, Stage_No)Values('"&p_course_id&"',
'"&p_stageno&"')"
        Mycon.Execute(SQL2)
     end if

     if p_stage3 = "yes" then
     p_stageno = 3
     SQL3 = "Insert Into CourseStages(Course_ID, Stage_No)Values('"&p_course_id&"',
'"&p_stageno&"')"
        Mycon.Execute(SQL3)
     end if

         if p_stage4 = "yes" then
     p_stageno = 4
     SQL4 = "Insert Into CourseStages(Course_ID, Stage_No)Values('"&p_course_id&"',
'"&p_stageno&"')"
        Mycon.Execute(SQL4)
     end if



    Mycon.Close
    set Mycon = Nothing
%>
<!--#include virtual="lffc/sa_Adminheader.html"-->
<P> Course Details have been Updated <BR></P>
<P><a href="sa_admin_main.asp">Return To Main Menu</a></P>
</BODY>
</HTML>
```

SA_UPDATE_COURSEMODULES.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<%
    if not Session("Auth")="SA" then
       Response.Redirect"admin.asp"
    end if

    set Mycon = Server.CreateObject("ADODB.Connection")
    Mycon.Open "lffc"

    p_course_id = Request.Form("p_course_id")
    p_mod_id = REquest.Form("p_mod_id")

%>
<!--#include virtual="lffc/sa_Adminheader.html"-->
<FORM METHOD="post" ACTION="sa_update_coursemodules_action.asp">
<table><tr><td width="205"><h5><b>Please Select Which Stage/Year's you wish to
apply to the Module to:</h5></b></td>
<input type="hidden" name="p_mod_id" value="<%=p_mod_id%>">
<input type="hidden" name="p_course_id" value="<%=p_course_id%>">

<%
p_stage1 = 0
p_stage2 = 0
p_stage3 = 0
p_stage4 = 0

    set catSet1 = Mycon.Execute("select * " _
    & "from Coursestages " _
    & "Where Course_ID = "&p_course_id)
     while not catSet1.EOF

     if catSet1("Stage_No") = 1 then
     p_stage1 = 1
     end if

     if catSet1("Stage_No") = 2 then
     p_stage2 = 2
     end if

     if catSet1("Stage_No") = 3 then
     p_stage3 = 3
     end if

     if catSet1("Stage_No") = 4 then
     p_stage4 = 4
     end if

         catSet1.MoveNext
     wend
     catSet1.Close
     set catSet1 = Nothing



     p_stage1_checked = ""
     p_stage2_checked = ""
     p_stage3_checked = ""
     p_stage4_checked = ""

    sqlText = "select * from CourseModules where Module_ID = " & p_mod_id & " AND
Course_ID = " & p_course_id
    set reviewSet = Mycon.Execute(sqlText)
    while not reviewSet.EOF

     if reviewSet("Stage_ID") = 1 then
```

```
    p_stage1_checked = "checked"
    end if

    if reviewSet("Stage_ID") = 2 then
    p_stage2_checked = "checked"
    end if

    if reviewSet("Stage_ID") = 3 then
    p_stage3_checked = "checked"
    end if

    if reviewSet("Stage_ID") = 4 then
    p_stage4_checked = "checked"
    end if

       reviewSet.MoveNext
   wend
   reviewSet.Close
   set reviewSet = Nothing


  if p_stage1 = 1 then
       %><td><h5><b>Stage/Year1:<INPUT TYPE="checkbox" NAME="p_stage1" VALUE="yes"
<%=p_stage1_checked%>><br><%
    end if

   if p_stage2 = 2 then
       %><td><h5><b>Stage/Year2:<INPUT TYPE="checkbox" NAME="p_stage2" VALUE="yes"
<%=p_stage2_checked%>><br><%
    end if

    if p_stage3 = 3 then
       %><td><h5><b>Stage/Year3:<INPUT TYPE="checkbox" NAME="p_stage3" VALUE="yes"
<%=p_stage3_checked%>><br><%
    end if

   if p_stage4 = 4 then
       %><td><h5><b>Stage/Year4:<INPUT TYPE="checkbox" NAME="p_stage4" VALUE="yes"
<%=p_stage4_checked%>><br><%
    end if

%>



</tr></table>
<table width="785">
   <TR><TD width="202"><INPUT TYPE="submit" VALUE="Associate Module with
Course"></form></td><td><A HREF="sa_admin_main.asp">Back To Main
Menu</A></b></td></TR>
   </TABLE>

<%

    Mycon.Close
    set Mycon = Nothing
%>
```

SA_UPDATE_COURSEMODULES_ACTION.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<%
    if not Session("Auth")="SA" then
       Response.Redirect"admin.asp"
    end if

       set Mycon = Server.CreateObject("ADODB.Connection")
       Mycon.Open "lffc"


     p_course_id = Request.form("p_course_id")
     p_stage1 = Request.form("p_stage1")
    p_stage2 = Request.form("p_stage2")
    p_stage3 = Request.form("p_stage3")
    p_stage4 = Request.form("p_stage4")
    p_mod_id = Request.form("p_mod_id")

    DeleteSQL = "DELETE FROM CourseModules WHERE Course_ID = " &p_course_id & " and
Module_ID = " &p_mod_id
     Mycon.Execute(DeleteSQL)

    if p_stage1 = "yes" then
     p_stageno = 1
          ID_SQL1 = "Select max(Entry_No) as maxid from CourseModules"
       set idSet1 = Mycon.Execute(ID_SQL1)
       p_entry_id1 = idSet1("maxid") + 1
       idSet1.Close
       set idSet1 = Nothing
     SQL1 = "Insert Into CourseModules(Entry_No, Module_ID, Course_ID,
Stage_ID)Values('"&p_entry_id1&"', '"&p_mod_id&"', '"&p_course_id&"',
'"&p_stageno&"')"
       Mycon.Execute(SQL1)
     end if

     if p_stage2 = "yes" then
     p_stageno = 2
          ID_SQL2 = "Select max(Entry_No) as maxid from CourseModules"
       set idSet2 = Mycon.Execute(ID_SQL2)
       p_entry_id2 = idSet2("maxid") + 1
       idSet2.Close
       set idSet2 = Nothing
     SQL2 = "Insert Into CourseModules(Entry_No, Module_ID, Course_ID,
Stage_ID)Values('"&p_entry_id2&"', '"&p_mod_id&"', '"&p_course_id&"',
'"&p_stageno&"')"
       Mycon.Execute(SQL2)
     end if

     if p_stage3 = "yes" then
     p_stageno = 3
               ID_SQL3 = "Select max(Entry_No) as maxid from CourseModules"
       set idSet3 = Mycon.Execute(ID_SQL3)
       p_entry_id3 = idSet3("maxid") + 1
       idSet3.Close
       set idSet3 = Nothing
     SQL3 = "Insert Into CourseModules(Entry_No, Module_ID, Course_ID,
Stage_ID)Values('"&p_entry_id3&"', '"&p_mod_id&"', '"&p_course_id&"',
'"&p_stageno&"')"
       Mycon.Execute(SQL3)
     end if

         if p_stage4 = "yes" then
     p_stageno = 4
          ID_SQL4 = "Select max(Entry_No) as maxid from CourseModules"
       set idSet4 = Mycon.Execute(ID_SQL4)
       p_entry_id4 = idSet4("maxid") + 1
```

```
       idSet4.Close
       set idSet4 = Nothing
     SQL4 = "Insert Into CourseModules(Entry_No, Module_ID, Course_ID,
Stage_ID)Values('"&p_entry_id4&"', '"&p_mod_id&"', '"&p_course_id&"',
'"&p_stageno&"')"
    Mycon.Execute(SQL4)
     end if


       if Err.number = 0 then

           'All is well with the world %>
           <!--#include virtual="lffc/sa_Adminheader.html"-->

           <H2>Course Associated with Module successfully</H2>

           Back To<A HREF="sa_admin_main.asp">Main Menu</A>


<%     else %>

           <H2> Problem </H2>
           There was a problem adding the Course
           Please go back and try again
           <A HREF="sa_updatemodules.asp">ADD Course</A>

<%      end if
    Mycon.Close
    set Mycon = Nothing

%>
```

SA_UPDATE_LECT.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<%
    if not Session("Auth")="SA" then
       Response.Redirect"admin.asp"
    end if

     Const adOpenKeyset = 1
    Const adLockOptimistic = 3
    Const adCmdText = &H0001

    set Mycon = Server.CreateObject("ADODB.Connection")
    Mycon.Open "lffc"

    p_lect_id = Request.Form("p_lect_id")

   set catSet = Mycon.Execute("select * " _
   & "from Admin " _
   & "Where Lecturer_ID = "&p_lect_id)
    while not catSet.EOF
%>
     <!--#include virtual="lffc/sa_Adminheader.html"-->
    <h3><b>Lecturer's Details:<%=catSet("Name")%></b></h3>
     <form method="POST" action="sa_update_lect_action.asp">
    <TABLE width="788">
    <TR><TD width="458"><INPUT TYPE="hidden" NAME="p_lect_ID"
VALUE="<%=catSet("Lecturer_ID")%>"></TD></TR>
    <TR><TD width="458"><h5><b>Name:</h5></b></TD><TD>
       <INPUT TYPE="text" NAME="p_name" VALUE="<%=catSet("Name")%>"></TD></TR>
     <TR><TD width="458"><h5><b>Password:</h5></b></TD><TD>
       <INPUT TYPE="text" NAME="p_pass" VALUE="<%=catSet("Password")%>"></TD></TR>
   <TR><TD width="458"><h5><b>Email:</h5></b></TD><TD>
    <INPUT TYPE="text" NAME="p_email" VALUE="<%=catSet("Email")%>"></TD></TR>
   <tr><td width="458"><h5><b>Is The Lecturer a SuerAdmin:</b></h5></td><td><SELECT
NAME="p_sadmin"><OPTION VALUE="False">False<OPTION value="True">True</td></tr>

    <TR><TD width="458"><INPUT TYPE="submit" VALUE="Update Lecturer's
Details"></form>   
       <b><A HREF="sa_admin_main.asp"><font size="4" color="#000000">Back TO Main
Menu</font></A></b></td></TR>
    </TABLE>
<%
    catSet.MoveNext
    wend
    catSet.Close
    set catSet = Nothing

     Mycon.Close
     set Mycon = Nothing
%>
```

SA_UPDATE_LECT_ACTION.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<%
    if not Session("Auth")="SA" then
       Response.Redirect"admin.asp"
    end if

    Const adOpenKeyset = 1
    Const adLockOptimistic = 3
    Const adCmdText = &H0001

    p_lect_id = Request.Form("p_lect_id")

    set Mycon = Server.CreateObject("ADODB.Connection")
    Mycon.Open "lffc"

    mySQL = "select * from Admin where Lecturer_ID= " & p_lect_id

    set myRS = Server.CreateObject("ADODB.RecordSet")
    myRS.Open mySQL, Mycon, adOpenKeyset, adLockOptimistic, adCmdText
    myRS("Name") = Request.form("p_name")
    myRS("Password") = Request.form("p_pass")
    myRS("SuperAdmin") = Request.form("p_sadmin")
    myRS("Email") = Request.form("p_email")
    myRS.Update

    myRS.Close
    Set myRS = Nothing

    Mycon.Close
    set Mycon = Nothing
%>
<!--#include virtual="lffc/sa_Adminheader.html"-->
<P> Lecturer's Details Updated <BR></P>
</BODY>
</HTML>
```

SA_UPDATECOURSES.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<%
   if not Session("Auth")="SA" then
      Response.Redirect"admin.asp"
   end if

%>

<html>
<!--#include virtual="lffc/sa_Adminheader.html"-->
<p><b><font color="#0000FF" size="5">Updateing Courses</font></b></p>
<p>
<H3><b><a href="sa_addcourse.asp">Add a New Course</a></b></H3>
</p>
<p>
<H3><b><a href="sa_delcourse.asp">Delete an Exsisting Course</a></b></H3>
</p>
<p>
<H3><b><a href="sa_editcourse.asp">Edit Exsisting Course Details</a></b></H3>
</p>
</html>
```

SA_UPDATEMODULES.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<%
    if not Session("Auth")="SA" then
       Response.Redirect"admin.asp"
    end if

%>

<html>
<!--#include virtual="lffc/sa_Adminheader.html"-->
<p><b><font color="#0000FF" size="5">Updateing Modules</font></b></p>
<p>
<H3><b><a href="sa_addmodule.asp">Add a New Module</a></b></H3>
</p>
<p>
<H3><b><a href="sa_coursemodules.asp">Associate Modules with Courses</a></b></H3>
</p>
<p>
<H3><b><a href="sa_editmodule.asp">Edit Module Name and Lecturer</a></b></H3>
</p>
<p>
<H3><b><a href="sa_deleteModule.asp">Delete an Exsisting Module</a></b></H3>
</p>
<p>
<H3><b><a href="sa_editmodulenotes.asp">Edit an Exsisting Modules
Notes</a></b></H3>
</p>
</html>
```

SA_VIEWTABLE_ACTION.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<%
    if not Session("Auth")="SA" then
       Response.Redirect"admin.asp"
    end if

    p_table_name = Request.form("p_table_name")

    Const adOpenForwardOnly = 0
    Const adLockOptimistic = 3
    Const adCmdText = &H0001
%>
<!--#include virtual="lffc/sa_Adminheader.html"-->
<H2><%= p_table_name %></H2>

<TABLE BORDER="1">

<%
    set Mycon = Server.CreateObject("ADODB.Connection")
    Mycon.Open "lffc"

    set tableSet = Server.CreateObject("ADODB.RecordSet")
    tableSet.Open p_table_name, Mycon, adOpenForwardOnly, adLockOptimistic,
adCmdTable
%>
    <TR>
<%      p_numberOfColumns = tableSet.fields.Count
        for x = 0 to (p_numberOfColumns - 1) %>
            <TH><%= tableSet.Fields(x).Name %></TH>
<%      next %>
    </TR>
<% while not tableSet.EOF %>
        <TR>
<%      for each col in tableSet.Fields %>
            <TD><%= col.Value %></TD>
<%          next      %>
        </TR>
<%      tableSet.MoveNext
    wend
    tableSet.Close
%>

</TABLE>
<%
    Mycon.Close
    set Mycon = Nothing
%>
</TABLE>
```

SEARCH.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<%
 set Mycon = Server.CreateObject("ADODB.Connection")
     Mycon.Open "lffc"
%>
<!--#include virtual="lffc/Header.html"-->
<head>
<meta http-equiv="Content-Language" content="en-ie">
<title></title>
<link rel="stylesheet" href="mm_training.css" type="text/css" />
</head>
<body bgcolor="#64748B">
<table width="100%" border="0" cellspacing="0" cellpadding="0">
    <tr bgcolor="#D3DCE6">
    <td width="230" colspan="2" valign="top" bgcolor="#26354A"><img
src="webimages\mm_training_photo.jpg" alt="Home Page Image" width="230"
height="286" border="0" /><br />
    <table border="0" cellspacing="0" cellpadding="0" width="230">
        <tr>
        <td width="230" class="sidebarText" id="padding"><br />     </td>
        </tr>
    </table>  </td>
    <td width="50" valign="top"><img src="mm_spacer.gif" alt="" width="50"
height="1" border="0" /></td>
    <td width="440" valign="top"><br />
    <br /> </td>
    <td width="40"> </td>
    <td width="100%"><H3> Search For Notes</H3>
<FORM ACTION="search_action.asp" METHOD="post">


    Look For:
    <INPUT TYPE="text" NAME="p_text">
    <BR>
    In Module:
    <SELECT NAME="p_mod_id">
<%
        set catSet = Mycon.Execute("select * from Modules")
        while not catSet.EOF
%>
        <OPTION VALUE="<%= catSet("Module_ID") %>"><%
           Response.Write catSet("Module_Name")%> <%
        catSet.MoveNext
    wend
    catSet.Close
    set catSet = Nothing
%>
    </SELECT>
    <P>
    <INPUT TYPE="submit">

</FORM></td>
    </tr>

    <tr bgcolor="#D3DCE6">
    <td colspan="6"><img src="mm_spacer.gif" alt="" width="1" height="1" border="0"
/></td>
    </tr>

    <tr>
    <td width="15"> </td>
    <td width="215"> </td>
    <td width="50"> </td>
    <td width="440"> </td>
    <td width="40"> </td>
```

```
    <td width="100%"> </td>
    </tr>
</table>
<%
    Mycon.Close
    set Mycon = Nothing
%>
</body>
</html>
```

SEARCH_ACTION.ASP

```
<%@ LANGUAGE="VBSCRIPT" %>
<!--#include virtual="lffc/Header.html"-->
<head>
<%
    p_text = cstr(Request.form("p_text"))
    p_text = uCase(p_text)
    p_mod_id = Request.form("p_mod_id")

    set Mycon = Server.CreateObject("ADODB.Connection")
    Mycon.Open "lffc"
%>
<meta http-equiv="Content-Language" content="en-ie">
<title></title>
<link rel="stylesheet" href="mm_training.css" type="text/css" />
</head>
<body bgcolor="#64748B">
<table width="100%" border="0" cellspacing="0" cellpadding="0">
    <tr bgcolor="#D3DCE6">
    <td width="230" colspan="2" valign="top" bgcolor="#26354A"><img
src="webimages\mm_training_photo.jpg" alt="Home Page Image" width="230"
height="286" border="0" /><br />
    <table border="0" cellspacing="0" cellpadding="0" width="230">
        <tr>
        <td width="230" class="sidebarText" id="padding"><br />     </td>
        </tr>
    </table>  </td>
    <td width="50" valign="top"><img src="mm_spacer.gif" alt="" width="50"
height="1" border="0" /></td>
    <td width="440" valign="top"><br />
    <br /> </td>
    <td width="40"> </td>
    <td width="100%">
<%
    set catset2 = Mycon.Execute("select * " _
     & "from Modules " _
     & "where Module_ID = " &p_mod_id)
        while not catSet2.EOF
%>
<H3><b><%= catSet2("Module_Name") %></b></H3>
<%                  catset2.MoveNext
       wend
       catset2.Close
       set catSet2 = Nothing

    sqlText = "SELECT * FROM ModuleNotes WHERE Module_ID = " & p_mod_id & " and
(uCase(Title) like '%"&p_text&"%')"


    set reviewSet = Mycon.Execute(sqlText)
    while not reviewSet.EOF

%>
<h5><A href="/lffc/ModuleNotes/<%Response.write
Server.URLencode(reviewSet("Doclink"))%>"><%Response.write
reviewSet("Title")%></A></h5>
<%
    reviewSet.MoveNext
       wend
       reviewSet.Close
       set reviewSet = Nothing

      Mycon.Close
      set Mycon = Nothing
%>
   </td>
```

```
    </tr>

    <tr bgcolor="#D3DCE6">
    <td colspan="6"><img src="mm_spacer.gif" alt="" width="1" height="1" border="0"
/></td>
    </tr>

    <tr>
    <td width="15"> </td>
    <td width="215"> </td>
    <td width="50"> </td>
    <td width="440"> </td>
    <td width="40"> </td>
    <td width="100%"> </td>
    </tr>
</table>
</body>
</html>
```