CHAPTER 4 - ANALYSIS

This chapter takes the form of a narrative reflecting on my learning and the contributions of others improving my learning. The order of it is largely chronological, based on an extensive research diary I kept throughout the process. Where others have made contributions their own words are included. The next chapter deals with the testing conducted with users in relation to the artifact constructed is presented and analyzed. Central to the process even at the initial stages were the other participants in the process.

4.1 Overview

In terms of the AAHE Taxonomy of Electronic Portfolios the artifact created in this study is situated as follows:

- Author – Darragh Power
- Context – A small scale study focused on my own learning, and facilitating the learning of other fellow learners, in an informal way, as part of my training and development practice, and learning as a student in DCU.
- Context - Technology Platform – Open source applications
- Purposes – The purpose of this e-Portfolio system is for self development and documenting learning over time. As the AAHE (2004) taxonomy says:

> **E-Portfolios for Development.**
>
> *Self-assessment*: using portfolios to track an individual's process of reflecting on and analyzing activities and performance; identifying strengths and weaknesses.
>
> *Documenting learning over time*: using portfolios to present evidence about individual or organizational growth and development, and the differentiation and integration of knowledge.
>
> (ibid 2004)

4.2 Initial Stages – The People.

4.2.1 The Participants

David Gallagher and Tariq Ali – third level students and co-workers.

John Flynn – A manager in the organization.

Darragh Power – The researcher.

In keeping with my educational values, the participants in this study are considered as co-researchers. In incorporating the contributions of others in their own words I hope to live my educational values, and allow for multiple voices to emerge. This plurality of voices is a factor in demonstrating rigour in qualitative research according to Winter (1989).

4.2.2 Other contributors and supervisors

Cormac Moran and Cathal Gurrin have provided considerable assistance with database design and SQL. In addition Cathal is a supervisor for my thesis. Margaret Farren is also a supervisor, and has provided tremendous support throughout the study. Realtan Ni Leanain, another supervisor on the course has also provided considerable assistance in relation to the focus of the study. Enda Lydon, Aoife O'Brien, Fiona Williams, Fionabarra Hallissey, Brendan Ryan, Claire Thomas, Trudy Corrigan, and Miriam Fitzpatrick, some of my class mates in the M.Sc in Education and Training in DCU have contributed significantly to the research process. All of the people listed above have at various times influenced my thinking in relation to the study.

A tutorial which I found out about through contact with Bob (not their real name) a student in another Irish University is also a vital contributor to the design process. I met Bob through the Living Action Research Listserv forum run by Rev. Je Kan Adler-Collins, an action research practitioner in the medical profession based in Japan. Bob requested that I maintain their anonymity and the anonymity of the tutorial.

4.3 Knowledge and Experience

None of the participants including myself have any experience with the open source applications Apache Web Server, MySQL or PHP. I have some experience in HTML and have designed five websites in the last 18 months having learned the skills in year one of the H.Dip / M.Sc. in Education and Training Management in Dublin City University (DCU). I have a very basic understanding of JavaScript, in that I can manipulate it, but I don’t understand its syntax. I have no experience of database design theory or practice, having previously attempted to design a database in Microsoft Access for an assignment in DCU, which was a flat tabled database, with no entity-relationship modeling etc (see Appendix A for a glossary of terms). Tariq Ali has some experience in HTML and JavaScript and assisted me in learning HTML. David Gallagher has no experience in programming. John Flynn has experience in Microsoft Access Databases.

4.4 Expectations

> **David Gallagher’s narrative**
>
> I hope that the e-Portfolio enables me to collect, organize and store data in a clear and concise manner. This would be very advantageous as it would enable me to access stored information quicker than I can at the moment. I am particularly looking forward to seeing how the search function operates. The majority of search engines and databases I use are inconsistent and

> incomplete and anything that enables me to accurately search through stored information would be most welcome.
>
> Email excerpt – 31/3/04

Tariq Ali's Narrative

> I would be particularly interested in the diary function. I think it would be ideal if, once all of the diary entries are complete for a particular project they could be printed or exported in formatted manner. This would be ideal for project reviews and to submit as an appendix to project work. This could illustrate that a methodical approach has been taken. Also, if feasible, would it be possible to include some sort of project-planning aspect to the portal? I think it may be useful to have a visible project plan on the customized homepage. This would assist with the time management of a project.
>
> Email excerpt – 31/3/04

John Flynn the third participant did not respond with his reflections on what functions he would like to see as part of the artifact.

4.5 My Expectations

I hope to learn the basics of database design, and apply them in MySQL. I hope to learn the basics of programming in PHP and SQL, in terms of the syntax of the languages and how to apply them. I hope to have a working e-Portfolio system which has been user tested which can be used by learners in their learning and development. My aim is to monitor each contribution made by all the participants in my learning, and make this process of learning and design visible to myself and other learners. I aim to incorporate the design requirements of the participants in the study into the e-Portfolio, in so far as my technical abilities will allow.

4.6 – Cycle One.

4.6.1 Reflections on the Installation – Technology.

Downloading the applications is relatively straightforward if they are downloaded in a bundled package where they are all pre-configured such as EasyPHP. Downloading the individual applications and configuring them on a windows XP platform proved beyond my capabilities as it involved using the Dos shell and configuring the operating system in that way. I could not find any simple explanation which explains to a layman, how this is done on any of the web resources of which I compiled a large number. This may be due to the fact that most developers using these applications develop them on Linux, and already have knowledge of the hardware and how to configure operating systems. It may also be due to the fact that Apache, MySQL and PHP can be located in bundled packages such as EasyPHP. The only disadvantages I can imagine in relation to this approach is that if something goes wrong, not understanding what is happening in the background may mean difficulties in recovering data. The other disadvantage is that the bundled packages are not the latest releases of any of the applications, so for a more advanced user they may lose functionality.

4.6.2 Downloading and installing the applications.

The diary entry below is my diary reflections on the downloading and installation of Apache Web Server, MySQL and PHP, which I had significant difficulty with, primarily due to the fact that I downloaded the individual applications prior to finding a bundled solution.

> **4/3/04 – Diary – Downloaded EasyPHP**
>
> Downloaded EasyPHP and took notes from http://moodle.org regarding installing it. The link for the download was through http://moodle.org also. My initial intention was to use Moodle in some way to support my education and learning practice, and I also wanted to learn how to use Apache Web Server, MySQL and PHP.
>
> **5/3/04 – Diary – Installing EasyPHP - Difficulties**
>
> I had already downloaded and installed individually, PHP4, My SQL and Apache Web server with different configurations, in about January 2004. These I was never able to configure properly as I found that many websites supporting the LAMP platform are pitched at a level above where I am at. IE – The tutorials available from www.webmonkey.com and www.codewalkers.com assumed that the person downloading the tutorials had a reasonable knowledge of object oriented programming terms like, strings, arrays, and were comfortable using the DOS Shell command line in Windows etc. In other words I completely failed to get to grips with the configuration as the supports available for a novice are inadequate. Maybe this is a recommendation for the Open source community. It also reflects the problems with

> the Windows platforms market penetration as the point and shoot and plug and play philosophy doesn't encourage or support adequately the use of open source software such as Apache Web Server, MySQL and PHP.
>
> These initial separate installs that I couldn't get to work, I had to uninstall as I am not technically literate enough to edit the properties of the already installed files. Also on the Moodle installation documents it suggests uninstalling all previous versions of the above packages.
>
> The installation of EasyPHP was considerably easier as it did not involve any configuration editing. The documentation from Moodle was very useful in installing and running it, and pointed to the folder www as being the one in which to store php and html files. It also has already got MySQL set up in a separate folder and pointed out that there is a difficulty in MySQL in setting it up for multiple users and linked tables.

Having downloaded and installed the EasyPHP package I wondered how I was going to configure it to run correctly. I also wondered how I would learn the syntax of the languages SQL and PHP, and how they would interact and work together. I would not have managed to resolve these without the contributions of others. A key example is discussed in the next section.

4.6.3 Social aspect of research is vital – The People.

> **16/3/04 – Diary Reflections – The Tutorial**
>
> As a result of my correspondence with Bob the student from the other that I met through Rev. Je Kan Adler Collins Living Action Research forum, I downloaded a tutorial from a contributor to the course in the other university. I had a look at Bobs own Reflective Practitioner tool which they designed, and it influenced my thinking in terms of what I would like to see possible for my own reflections. The tutorial shaped my thinking around the rough structure of the PHP and HTML and how they interact, and I will probably use much of the source code from this tutorial which I will tailor to suit my own artifact.

This tutorial proved to be an invaluable source of source code for the application and much of the structure for the application is taken from it. Without this overview of how an application might work, this research would be impossible for me as an absolute novice in programming in PHP and application development. I made contact with Bob through Living Action Research List Serv which is run by a Rev. Je Kan Adler-Collins an action researcher in the medical profession based in Japan. I posted a link to a web page I designed with my class mates on the M.Sc in DCU http://student.dcu.ie/~powerd3/classweb which prompted Bob to make contact, and as a result of corresponding about action research methodologies on e-mail Bob suggested I take a look at the tutorial. (Bob and the tutorial remain anonymous at their own request).

Reflecting on this process I am struck by the living interventions of others shaping the course of my learning practice, and that sharing information is a very constructive process. I am also struck by the ideas of Winter (1989) on rigour which suggest that multiple voices and perspectives show rigour in a qualitative research study. Another factor in establishing rigour is the interaction between theory and practice.

4.6.4 The three tier design approach – Technology.

My initial thoughts in relation to the usage of Apache web server, MySQL and PHP were very chaotic. In not understanding the way the applications interact I found it very difficult to conceptualise the design process. An intervention from theory in this regard was Williams and Lane (2003) which I found very helpful. My diary entry below illustrates this.

> **17/3/04 – Diary Reflections - Hugh Williams and David Lane – Web Database Applications, with MySQL and PHP – Book**
>
> I've just purchased the book which seems to be the definitive text on the creation of web database applications with MySQL and PHP. I found many of the tutorials on the web pitched at too high a level in that I can understand the logic of object oriented programming, but the syntax and language of strings, arrays, variables and objects is a difficult concept as I don't have a background in C or Java, which is assumed by many tutorials. Over 60% of web based applications are on the MySQL and PHP platform. This absolute novice explanation may be a recommendation for the open source community in the future in their construction of tutorials.
>
> The three tier architecture structure – namely MySQL database – PHP tier – Client Tier – fundamentally shaped my thinking about what needs to be programmed, and more importantly what the design structure should be like.

I was going to dive in and try to get a feel for programming and I started constructing a frontpage, and login screen using PHP and HTML, however the complexity of the structures involved made me re-think.

4.6.5 Designing the Database

22/3/04 – 24/3/04 - Diary Reflections – Tables – Database design

I drafted a number of basic database structures to try and conceptualise the database design for the artifact. Cormac Moran made a valuable intervention here as well in his advice to plan, and diagram the process. Cormacs' inputs can be seen on Database tables revision tables2basic1.xls and in particular from tables2basic2.xls to tables2basic4.xls. I also distributed copies via e-mail of the table revisions to each person involved in the design process.

Cathal Gurrin also suggested that I add a first name and last name field to the user table, in addition to the username, which will be important should the artifact develop to a wider user base.

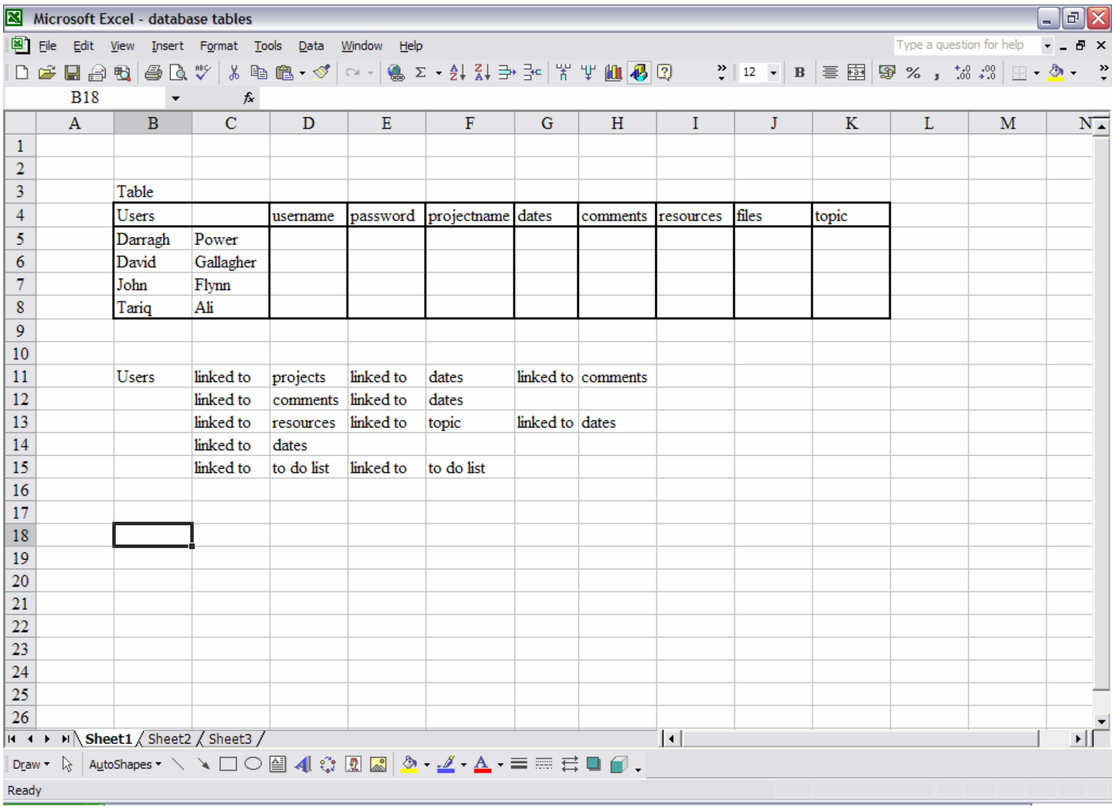
4.7 Excel Files:

Each Excel file represents a change in my thinking and the learning process as I modify the design of the artifact with the help of Cormac, Cathal, David and Tariq. A brief synopsis of the changes is outlined below. The full files can be viewed on the CD accompanying this study in the folder marked databasetablerevisions. You will note that my reflections on the three tier design structure refer mainly to the database tables. My concern at this time was that if the database tables were not correct it would be problematic later in the process, particularly in terms of querying the database using SQL. I will take you through the development of the design process now showing you the various changes in the design process and explain the rationale for the amendments, based on the contributions from each of the participants. This stage of the process is focused almost exclusively on the database table revisions, which took place in eight steps.

4.7.1 STEP 1 - Database tables.xls – Initial concept plan for the artifact.

Note that this essentially a concept document, an attempt to get a handle on what users will use. You will notice from **Fig. 4.1 database tables.xls** below that this is a single tab excel spreadsheet, with a view to figuring out the structures of the tables of the database which will underpin the application. It will be evident to those who are experienced database designers that there is no attempt at figuring out the tables in relation to an entity relationship model. The entity relationship model of database design is the standard model for database design. It involves identifying the separate entities involved, in this case, users and the projects they have to do, and the files, comments and resources that constitute the project. The ways separate entities relate to each other are called relationships.

STEP 1 - Fig. 4.1 – database tables.



Microsoft Excel - database tables
File Edit View Insert Format Tools Data Window Help
Type a question for help
B18
Sheet1 Sheet2 Sheet3
Draw AutoShapes
Ready

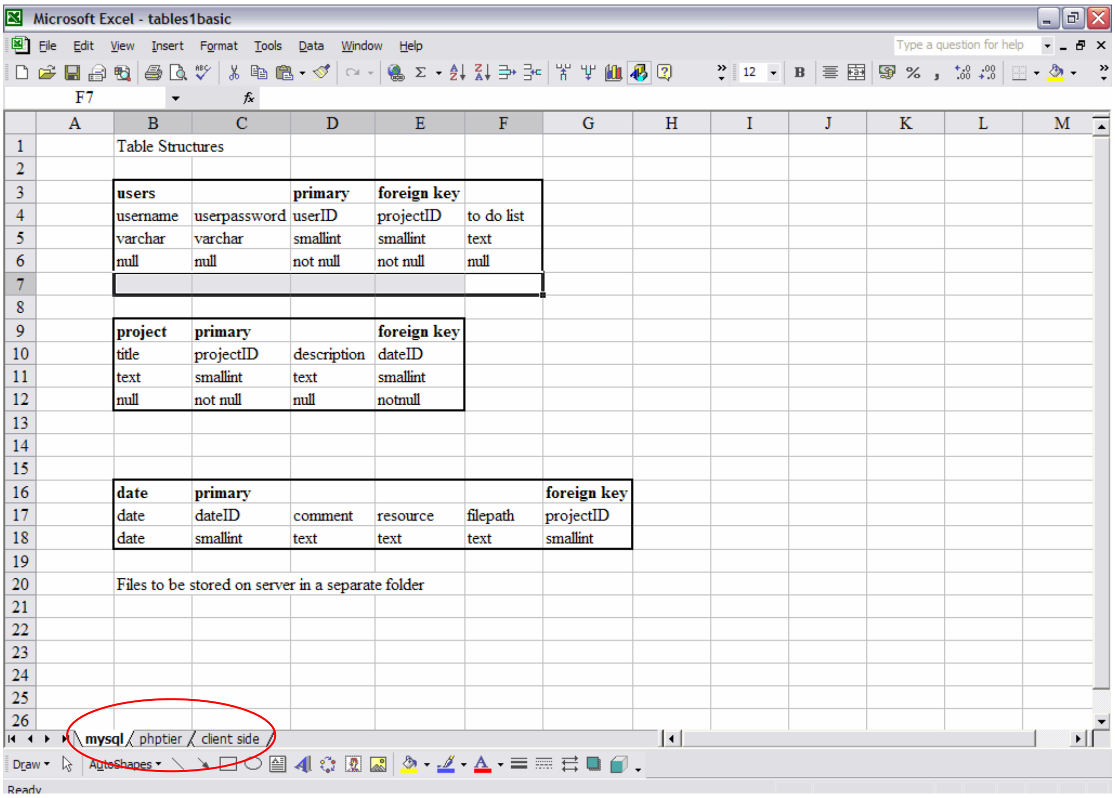
	A	B	C	D	E	F	G	H	I	J	K
3		Table									
4		Users		username	password	projectname	dates	comments	resources	files	topic
5		Darragh	Power								
6		David	Gallagher								
7		John	Flynn								
8		Tariq	Ali								
11		Users	linked to	projects	linked to	dates	linked to	comments			
12			linked to	comments	linked to	dates					
13			linked to	resources	linked to	topic	linked to	dates			
14			linked to	dates							
15			linked to	to do list	linked to	to do list					

4.7.2 STEP 2 - Tables1basic.xls - Initial design following three tier approach as a result of reading Williams and Lane (2002) – Note the introduction of the three tabs:

- MySQL – Database Structures
- PHP Tier – Server Sided Scripting
- Client Side – What the user will see

This intervention from the literature fundamentally shaped my thinking on what was needed for the design. The database needs to be interpreted by the PHP tier in order to be sent out to Internet Explorer or an equivalent browser.

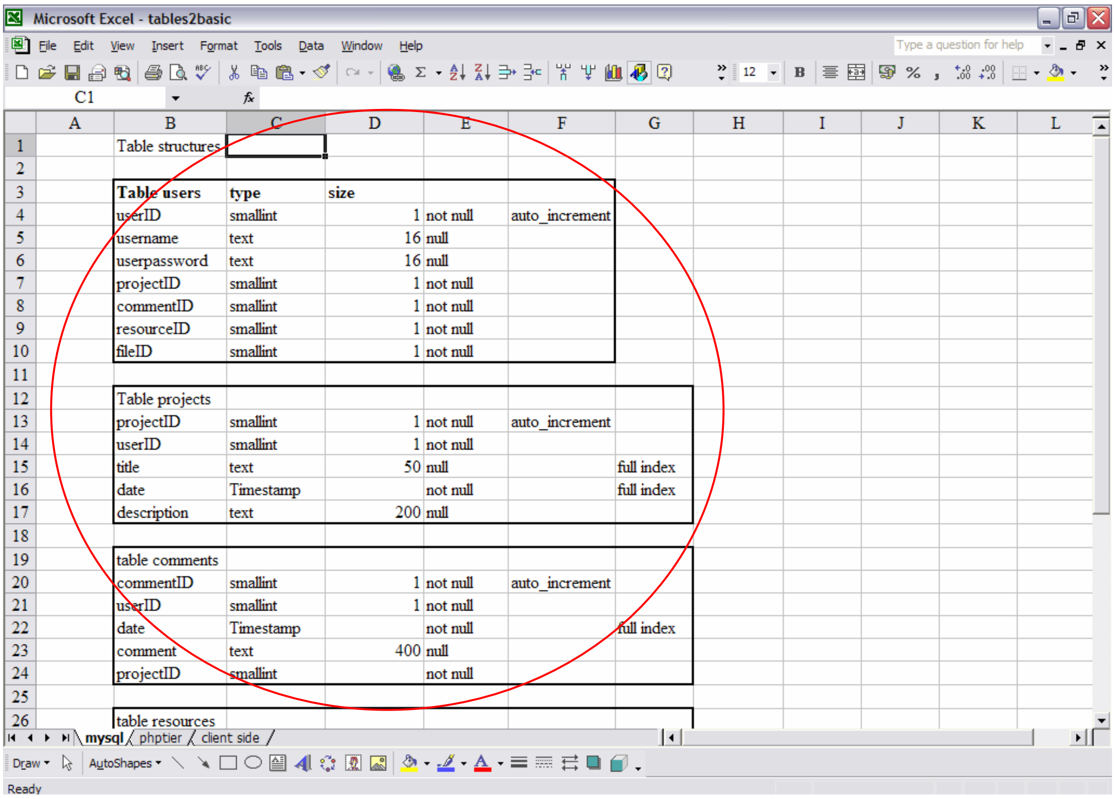
STEP 2 - Fig. 4.2 – tables1basic.xls


Microsoft Excel - tables1basic
File Edit View Insert Format Tools Data Window Help
Type a question for help
F7
mysql
phptier
client side
Draw
AutoShapes
Ready

	A	B	C	D	E	F	G
1		Table Structures					
2							
3		users		primary	foreign key		
4		username	userpassword	userID	projectID	to do list	
5		varchar	varchar	smallint	smallint	text	
6		null	null	not null	not null	null	
7							
8							
9		project	primary		foreign key		
10		title	projectID	description	dateID		
11		text	smallint	text	smallint		
12		null	not null	null	notnull		
13							
14							
15							
16		date	primary				foreign key
17		date	dateID	comment	resource	filepath	projectID
18		date	smallint	text	text	text	smallint
19							
20		Files to be stored on server in a separate folder					

4.7.3 STEP 3 - Tables2basic.xls – Developing design through contact with Cathal and Cormac. I am learning the entity and relationship model of database design through trial and error with the help of Cathal and Cormac, with the result that the number of tables and fields in each table, and how these relate to each other is changing. Another key contribution I made by the book 'Database Design using Entity-Relationship Diagrams' by Bagui and Earp (2003), which was very helpful with regard to learning the basics of database design. There is an increase in the numbers of tables here and in addition the types of information stored in the columns, which has changed from having the table fields across the columns to having the table fields in the rows instead. This was a direct result of Cormacs intervention and helped the design to correspond to the conventions of database design. At this stage I am very much trying to conceptualise what the database will look like.

STEP 3 - Fig 4.3 tables2basic.xls – MySQL tab

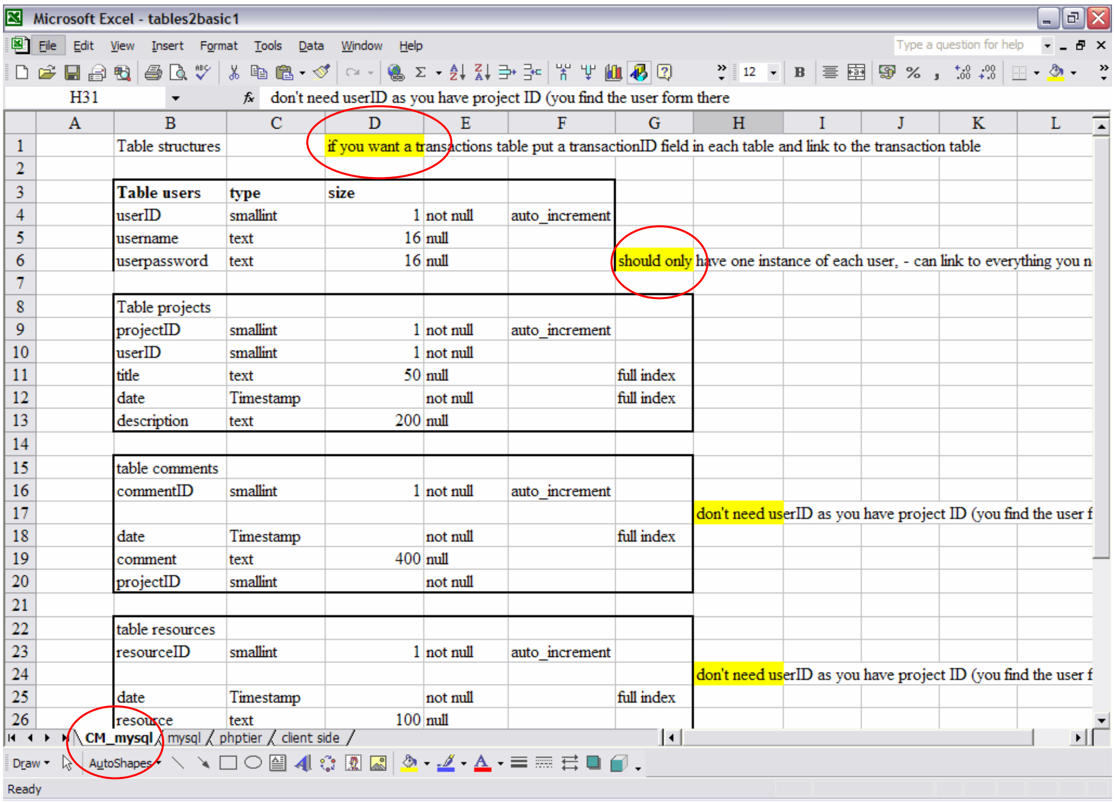


Microsoft Excel - tables2basic
File Edit View Insert Format Tools Data Window Help
Type a question for help
C1
mysql
phptier
client side
Draw
AutoShapes
Ready

	A	B	C	D	E	F	G
1		Table structures					
2							
3		**Table users**	**type**	**size**			
4		userID	smallint	1	not null	auto_increment	
5		username	text	16	null		
6		userpassword	text	16	null		
7		projectID	smallint	1	not null		
8		commentID	smallint	1	not null		
9		resourceID	smallint	1	not null		
10		fileID	smallint	1	not null		
11							
12		Table projects					
13		projectID	smallint	1	not null	auto_increment	
14		userID	smallint	1	not null		
15		title	text	50	null		full index
16		date	Timestamp		not null		full index
17		description	text	200	null		
18							
19		table comments					
20		commentID	smallint	1	not null	auto_increment	
21		userID	smallint	1	not null		
22		date	Timestamp		not null		full index
23		comment	text	400	null		
24		projectID	smallint		not null		
25							
26		table resources					

4.7.4 STEP 4 - Tables2basic1.xls – Cormac contributed a new tab pointing out problems with the initial database design tables. The tab CM_msql is Cormacs work and was a major factor in my thinking about how entity's and relationships work. Note Cormacs contributions highlighted in yellow. These contributions again point out how crucial the contributions of others are for me in the process of my learning. Cormac in knowing the way queries are constructed in SQL had a good understanding of the way a query in SQL would work. In particular Cormacs intervention can be seen in the way the primary and foreign keys are changing. The keys define the relationships between entities. So a user doing a project would need the users ID to appear in the project table of the database. This is called a foreign key. The primary key in the user table is the users ID. As the users ID appears in the project table, it shows the link between a user and the project that the user does. This learning is evident in **Fig 4.4 – Tables2basic1.xls.**

STEP 4 - Fig 4.4 – Tables2basic1.xls – CM_mysql tab



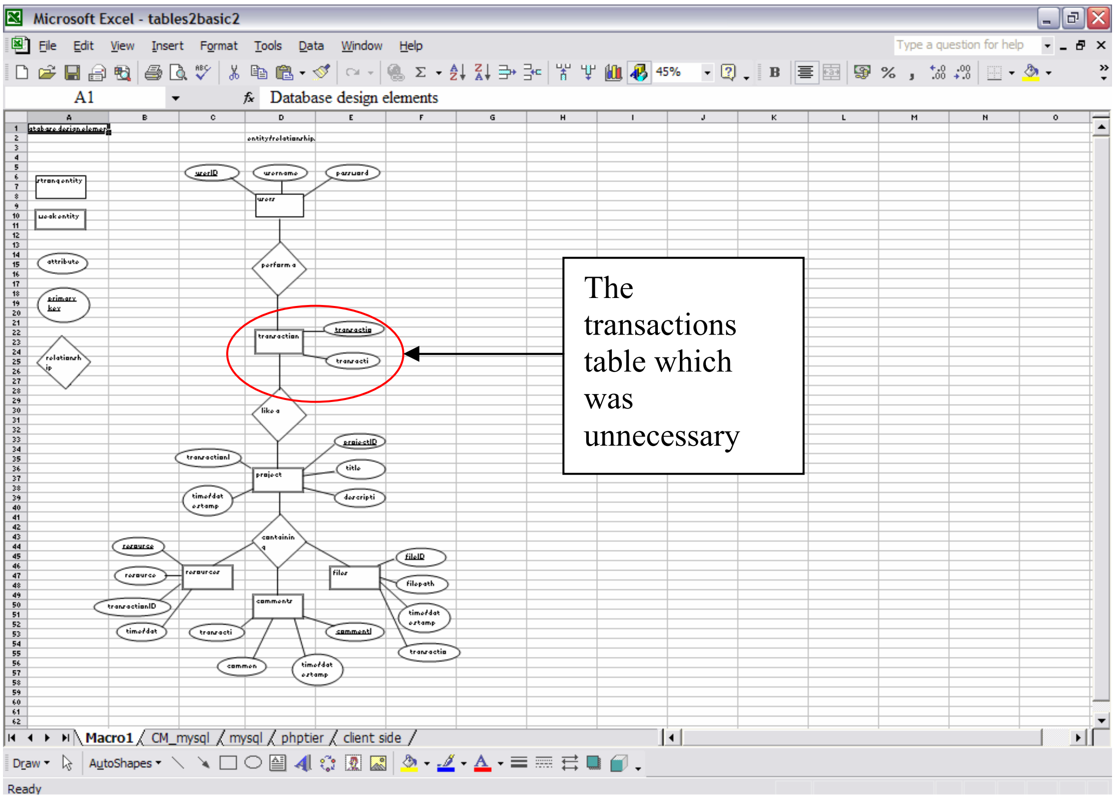
H31 don't need userID as you have project ID (you find the user form there

	A	B	C	D	E	F	G	H	I	J	K	L
1		Table structures		if you want a transactions table put a transactionID field in each table and link to the transaction table								
2												
3		**Table users**	**type**	**size**								
4		userID	smallint	1	not null	auto_increment						
5		username	text	16	null							
6		userpassword	text	16	null		should only have one instance of each user, - can link to everything you n					
7												
8		Table projects										
9		projectID	smallint	1	not null	auto_increment						
10		userID	smallint	1	not null							
11		title	text	50	null		full index					
12		date	Timestamp		not null		full index					
13		description	text	200	null							
14												
15		table comments										
16		commentID	smallint	1	not null	auto_increment						
17								don't need userID as you have project ID (you find the user f				
18		date	Timestamp		not null		full index					
19		comment	text	400	null							
20		projectID	smallint		not null							
21												
22		table resources										
23		resourceID	smallint	1	not null	auto_increment						
24								don't need userID as you have project ID (you find the user f				
25		date	Timestamp		not null		full index					
26		resource	text	100	null							

4.7.5 STEP 5 - Tables2basic2.xls – Following a suggestion from Cormac on an e-mail a new TAB Macro1 has been added to provide a visual representation of the database structures, to help to get an overview of how the database will look. This was done with help from Bagui and Earp (2003). The learning involved in doing the visual representation of the database involved learning how to use database design entity relationship visual diagram models, for strong and weak entities, attributes, relationships, primary keys and foreign keys. (See Appendix A – for a glossary of terms)

Note the addition of a transactions table which I thought would be necessary for querying the database, but as the structure emerged I found to be irrelevant, as each user, will have projects and the projects will have files, comments and resources. Therefore queries can be constructed using the user and project ID’s. In database language this would be that the foreign key in the project table would be the userID, and the files, comments and resources tables which are in relationship to the project table, have the projectID as the foreign key.

STEP 5 - Fig 4.5 – tables2basic2.xls – Macro1 tab



The transactions table which was unnecessary

4.7.6 STEP 6 - Tables2basic3.xls

4.7.6 STEP 6 - Tables2basic3.xls – Note the addition of a project ID field in the tables files, resources, comments which is for querying in SQL. This reflects the learning from the previous table. For a full visual representation of these tables please see the file **Tables2basic3.xls** on the accompanying CD.

STEP 6 – Fig 4.6 tables2basic3.xls – The MySQL tab

Table users	**Type**	**Size**			
userID	Smallint	1	not null	Auto_increment	primary key
Username	Text	16	Null		
Userpassword	Text	16	Null		

Table transactions	**Type**	**Size**			
transactionID	Smallint	1	not null	Auto_increment	primary key
userID	Smallint	1	not null		foreign key
Transaction	Text	10	Null		

Table projects	**Type**	**Size**			
Projected	Smallint	1	not null	Auto_increment	
transactionID	Smallint	1	not null		
Title	Text	50	Null		full index
Date	Timestamp		not null		full index
Description	Text	200	Null		

table comments	**Type**	**Size**			
Commented	Smallint	1	not null	Auto_increment	
Date	Timestamp		not null		full index
comment	Text	400	Null		
transactionID	Smallint		not null		

table resources	**Type**	**Size**			
resourceID	Smallint	1	not null	Auto_increment	
Date	Timestamp		not null		full index
Resource	Text	100	Null		
transactionID	Smallint		not null		

table files	**Type**	**Size**			
fileID	Smallint	1	not null	Auto_increment	
Date	Timestamp		not null		full index
Filepath	Text	100	Null		
transactionID	Smallint		not null		

STEP 7 - Tables2basic4.xls – Addition of first and last name fields in the user table following a contribution from Cathal. Note the removal of the transaction table which Cormac suggested is unnecessary if the projectID and userIDs are to be used for querying the database.

STEP 7 – Fig 4.7 – tables2basic4.xls – MySQL tab

Table users	**Type**	**Size**			
userID	Smallint	1	not null	Auto_increment	primary key
Username	Text	16	Null		
Userpassword	Text	16	Null		
Firstname	Text	16	Null		
Lastname	Text	16			

Table projects	**Type**	**Size**			
Projected	Smallint	1	not null	Auto_increment	
Title	Text	50	Null		full index
Description	Text	200	Null		
userID	Smallint	1	not null		
Date	Timestamp		not null		full index

Table comments	**Type**	**Size**			
Commented	Smallint	1	not null	Auto_increment	
Date	Timestamp		not null		full index
Comment	Text	400	Null		
Projected	Smallint		not null		

Table resources	**Type**	**Size**			
resourceID	Smallint	1	not null	Auto_increment	
Date	Timestamp		not null		full index

Resource	Text	100	Null		
Projected	Smallint		not null		

Table files	**Type**	**Size**			
fileID	Smallint	1	not null	Auto_increment	
Date	Timestamp		not null		full index
Filepath	Text	100	Null		
Projected	Smallint		not null		

4.7.8 STEP 8 - Tables3basic1.xls – Following Tariq Alis narrative the to-do-list table has been added to allow for the project management element of the portfolio system.

STEP 8 – Fig 4.8 – tables3basic1.xls

Table users	**Type**	**Size**			
userID	Smallint	1	not null	Auto_increment	primary key
Username	Text	16	Null		
Password	Text	16	Null		
Firstname	Text	16	Null		
Lastname	Text	16			

Table projects	**Type**	**Size**			
Projected	Smallint	1	not null	Auto_increment	
Title	Text	50	Null		full index
Description	Text	200	Null		
userID	Smallint	1	not null		
Date	Timestamp		not null		full index

Table to do list	**Type**	**Size**		
todolistID	Smallint	1	not null	Auto_increment
list item	Text		Null	
due date	Date		Null	
userID	Smallint	1	not null	
Projected	Smallint	1	not null	

Table comments	**Type**	**Size**

Commented	Smallint	1	not null	Auto_increment	
Date	Timestamp		not null		full index
Comment	Text	400	Null		
Projected	Smallint		not null		

Table resources	**Type**	**Size**			
resourceID	Smallint	1	not null	Auto_increment	
Date	Timestamp		not null		full index
Resource	Text	100	Null		
Projected	Smallint		not null		

Table files	**Type**	**Size**			
fileID	Smallint	1	not null	Auto_increment	
Date	Timestamp		not null		full index
Filepath	Text	100	Null		
Projected	Smallint		not null		

4.8 Database

Using MyPHPAdmin which is a web interface to access MySQL I constructed the database “thesis” which contains all the tables as per **Fig 4.8.**

4.9 Reflections on Politics

My diary entry below indicates some of the issues I was reflecting on, around the management of my time and resources while working in a commercial organization. I was hoping to be able to use some of the computer facilities in work, and also avail of some time in work to develop the artifact as part of my training role. As my organization is developing a culture of reflective practice for managers involving a construction of a portfolio of their work, I thought that some of the programming I was doing might be useful in helping with this, and also when the artifact is finished, I will be making the research available to some of my colleagues. My diary entry at the time illustrates some issues that arose in terms of support. Fionnuala Flanagan a contributor to the H.Dip / M.Sc in Education and Training Management in DCU suggested that these issues may occur in a similar way for other people in their own practice contexts, and may offer a transferable aspect of the research.

> **26/3/04 - Diary**
>
> The development of the artefact in work is problematic for a number of reasons.
>
> 1. The technical support company for my employer have specific commercial tie ins with a number of other vendors, which means that open source material such as PHP and EasyPHP cannot be installed on PC's in the office. This means that I need to have my own computer.
>
> 2. There is a politics in the use of open source, with the LAMP (Linux, Apache Web Server, MySQL and PHP) platform being a low cost competitor of the commercial software vendors. The aspects of the platform that are transferable to a windows platform are difficult to install, and are subject to an environment where the commercial vendors are openly hostile to the use of open source software. This reflects the potential loss of earnings that they represent to a commercial vendor. It also means that the artefact I develop is necessarily focussed on individual learners rather than towards organisational learning.
>
> 3. There is also a parallel perception in the open source community in terms of using MySQL in a commercial environment, whereby it is strongly advised on the MySQL website that commercial users purchase a commercial license for the product.
>
> 4. This dichotomy would be significantly more problematic for a larger scale project in a commercial environment, and may make the use of open source software prohibitive for other trainers and developers of learning objects. A recommendation may be that some middle ground be found which reduces the conflict between the open source community and commercial vendors. This may have to be negotiated at a local level within the users context.

As this diary entry indicates the fact that the computer support function in my organisation is outsourced. This outsourced organisation has tie-in connections with other commercial database and server sided scripting vendors there is a significant problem in terms of support in my organisation. The use of commercial products could not be countenanced in terms of the cost of buying licenses and also in terms of the ongoing support for such applications. This financial cost in addition to the costs in terms of time of learning to program in server sided scripting languages and SQL is potentially prohibitive for anyone seeking to go down this route. In addition the recurring fees in terms of licenses etc are an additional annuity cost.

These reasons are my motivation for using open source software which being financially cost free provide an opportunity for low cost development for me. I also felt that the open source community being one where collaboration is in evidence through the releasing of source code, would provide reasonable support through online tutorials etc. One of the difficulties as a result of this approach however is that I am relatively unsupported in terms of bringing the research into my daily practice as a trainer, so the development is taking place outside work on my own computer, which puts the users in a situation where they cannot see the development process except periodically.

4.10 Configuration Details

In order to start the development process I needed to be able to set up my own local computer to run the Apache Web Server, MySQL and PHP applications and configure them to run simultaneously and allow MySQL and PHP to read each other. This difficulty arose later at the stage where I was uploading the artefact to a server in order to allow for user testing. I will discuss this further later, but my reflections on the initial configuration on my local computer are outlined below.

> **29/3/04 – Diary Reflections - Technical Difficulties**
>
> I am struggling to configure MyPHPAdmin and MySQL for multiple users which is very frustrating. At this point I keep getting the error message:
>
> *"Error*
> *The additional Features for working with linked Tables have been deactivated. To find out why click here.*
>
> Clicking here brings me into a manual that is written for someone with considerably more technical knowledge than I currently possess. In other words it's not particularly user friendly – an example is:
>
> *"• HTTP and cookie authentication modes are recommended in a* ***multi-user environment*** *where you want to give users access to their own database and don't want them to play around with others.*
> *Nevertheless be aware that MS Internet Explorer seems to be really buggy about cookies, at least till version 6. And PHP 4.1.1 is also a bit buggy in this area!*
> *Even in a* ***single-user environment****, you might prefer to use HTTP or cookie mode so that your user/password pair are not in clear in the configuration file.*

- *HTTP and cookie authentication modes are more secure: the MySQL password does not need to be set in the phpMyAdmin configuration file (except for the "controluser" -see the Configuration section). However, keep in mind that the password travels in plain text, unless you are using the HTTPS protocol.*
In cookie mode, the password is stored, encrypted with the blowfish algorithm, in a temporary cookie."

```
GRANT USAGE ON MySQL.* TO 'pma'@'localhost' IDENTIFIED BY
'pmapass';
GRANT SELECT (
    Host, User, Select_priv, Insert_priv, Update_priv,
Delete_priv,
    Create_priv, Drop_priv, Reload_priv, Shutdown_priv,
Process_priv,
    File_priv, Grant_priv, References_priv, Index_priv,
Alter_priv,
    Show_db_priv, Super_priv, Create_tmp_table_priv,
Lock_tables_priv,
    Execute_priv, Repl_slave_priv, Repl_client_priv
    ) ON mysql.user TO 'pma'@'localhost';
GRANT SELECT ON mysql.db TO 'pma'@'localhost';
GRANT SELECT ON mysql.host TO 'pma'@'localhost';
GRANT SELECT (Host, Db, User, Table_name, Table_priv,
Column_priv)
    ON mysql.tables_priv TO 'pma'@'localhost'
```

This is definitely geared to a more technically literate user.

This type of error message I think could be explained in significantly easier terms for a layman, and this use of technical language is something that maybe the open source community could try to simplify for inexperienced users. I've included it in full to illustrate how difficult it is to understand. After a lot of trial and error and frustration over about five days I managed to configure the applications as my diary entry below shows.

3/4/04 Diary Reflection – Breakthrough in Technical Knowledge

I finally managed to configure the database and login successfully. The steps to this are as follows:

Step 1
In myPHPAdmin – the config.php file the line:

```
$cfg['Servers'][$i]['auth_type']      = 'config';
```

Needs to be changed to:

```
$cfg['Servers'][$i]['auth_type']      = 'http';
```

So that apache can process the http requests.

Step 2
A new user needs to be created on the specific database, to allow the user to access this database only, and not the entire database library. In my case I created a new user in myPHPAdmin, called thesis with a password thesis, and ran the following SQL query to grant this user all privileges on the database.

```
GRANT ALL PRIVILEGES ON thesis.* TO thesis@localhost IDENTIFIED BY 'thesis';
```

Step 3

The db.inc.php file which contains all the details of the database I edited to read.

```
<?php
   //This file defines the database settings. It can be included in all your php scripts that
interact with
   //your database. Then if you need to change the passwd or the databaseServer, etc you will
only have to change it once

   //define some parameters for the connection
   $databaseServer="localhost";       //specifies the database server. Can be changed if the
database moves
   $user="thesis";                    //specifies a username to log on to the database with.
   $passwd="thesis";                  //specifies a password for the username.
   $database="thesis";                //specifies the database on the server you wish to connect to.
   //NOTE: in this case user, passwd and database are all set to the same value.
   //This is for simplicity, but it won't always be the case when using databases on other
servers.
?>
```

This combination of factors allows the user to access the database. The register.html form processed by register.php inserts the new users details into the table on the database ‘thesis.’ This is a very satisfying breakthrough as it has taken me a long time to figure out these steps, as I could not find any simple explanations of how to do this.

Note any line beginning with // is not read by the browser, only the lines ending in “;” are PHP lines. As you will see I have just named everything “thesis” for convenience sake. This is the source code from the tutorial Bob introduced me to, which I modified to suit my needs, and to read my database “thesis.”

4.11 Reflections on the Design process to date.

As is evident from the chronology above it is now a month since the start of the project and it has taken four weeks of hard work to get to the point where I could start programming. One of my commitments with this research was to be very clear and honest with the difficulties in relation to the study. I do not want to underplay how difficult it has been in learning:

- the database structures
- how to draw an entity relationship diagram correctly
- how to put the design elements together in a group of tables that are workable
- how to download the correct software applications
- how to install and configure the software applications
- how to learn how to do these technical tasks and research the process at the same time

The result to date is workable but is by no means a definitive way of doing this kind of project, and as with the entire project it would not be possible to have got this far as a novice in the field without the assistance and contributions of other people in the process, in particular Cormacs input on the database was crucial to building a working model. At this stage in reflecting on the process to date I realise that most of my learning has been in relation to the database design theory and the configuration of MySQL to allow for multiple users to access the database. The full source code for each page is viewable in Appendix D, and the actual files are available on the CD attached.

4.12 Cycle Two

4.12.1 Starting to Program

Gary Greenberg cited in the NLII Annual Review (2003) discusses e-Portfolios and:

> ..advised session attendees to get started before waiting for the time when they think they can get it right." (ibid 2003: 28)

Given the length of time for this study I found the quote above to be a pragmatic cue to learning the programming by doing. Reflecting on how I learned how to programme in HTML I found that learning by trial and error and by actually doing the programming was the only way I could get to grips with the syntax of the language. With this in mind I started to learn to programme with PHP.

4.12.2 Initial steps in PHP

The first step in the process of learning to program in PHP was to copy all of the files from the tutorial that Bob introduced me to, into the folder C:\Program Files\EasyPHP1-7\www and seeing how they worked. I became aware that this is the folder to use for web files in PHP from reading through documentation at the http://moodle.org site which uses EasyPHP to support Moodle. Moodle is the open source course management system (CMS) that DCU use.

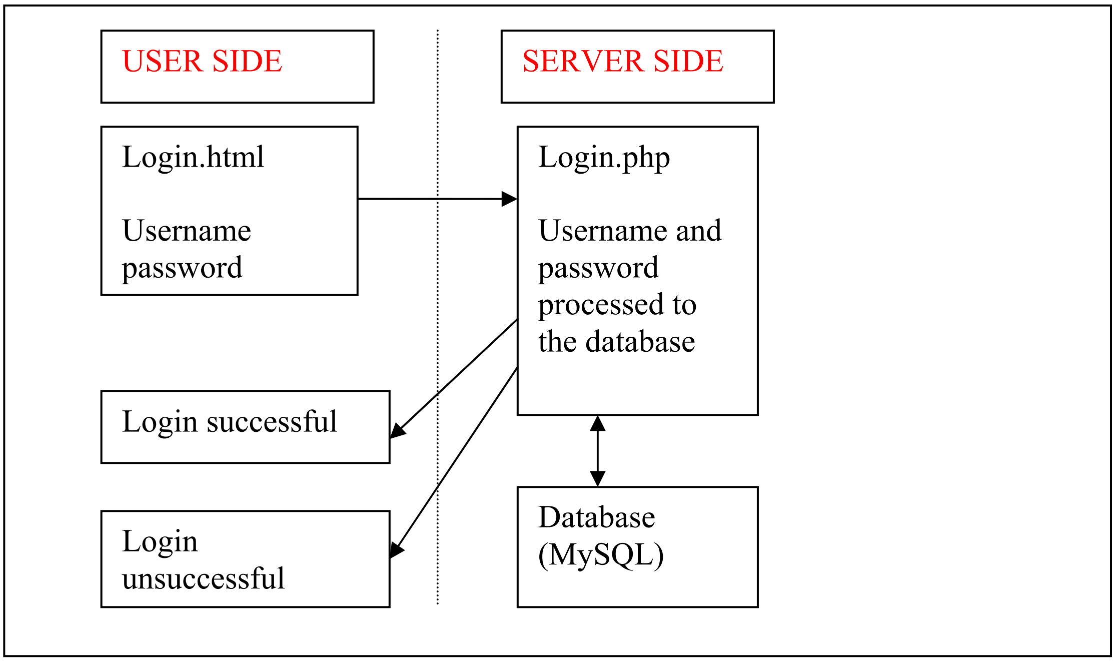
At this point I do not understand the technical syntax of the programming language PHP in terms of variables and arrays. As with HTML programming I adopted the view that why learn to re-create the wheel if I can pick up what I need from other websites which is a feature of open source development. Many open source sites exist which make source code available such as www.phpfreaks.com and www.php.net which is a significant reason in my choice of the applications. The result of this view was I decided to take as much of the source code as possible from the tutorial Bob introduced me to, and configure it where possible to suit my own needs. As my database is unique to my application much of the source code from the tutorial had to be re-written and re-configured, and in some cases added to. After the initial configuration of db.inc.php I moved onto the files register.html and CreateUser.php, and login.html and login.php. A major learning point for me was the realization that these files work as a pair. The file login.html creates a set of variables which are processed by login.php.

The following excerpt from login.html illustrates this. Note the action is to send the username and password to login.php which then processes it.

```
<form name="form1" method="post" action="login.php">
```

The page login.php then processes the form by sending the information to the MySQL database, and depending on whether the username and password are recognized by the database sends out a message to the user that their login was successful, or unsuccessful. This is illustrated in **Fig 4.9 login.html / login.php below.**

Fig 4.9 login.html / login.php



It is the same situation for the register.html and CreateUser.php files which also work in pairs. “Include(db.inc.php)” tells PHP to include the database “thesis,” the “$query” section is an SQL query which posts the username and password to the database and if the user is registered will collect their userID and allow them to open a session on the database. You will notice that there is a section at the top of the login.php page which mentions a teacher and admin log in which is a relic from the tutorial, but as it didn’t effect the workings of the script I’ve left it in. It also gives me the option to setup an administrator function as my technical skills develop. The code that I can amend in the future is quoted below. Currently my technical knowledge is not sufficient to process this but I may gain these skills in the future and it is with a view to leaving the option open that this code remains rather than being deleted.

```
     if($username=="teacher"){
         //if they did we will have to check that they got the right
password
         if($password=="admin"){
             //if all is correct we can set the teacher session
variable
             //and direct the teacher to the teacher menu
             $_SESSION['teacher']=true;
             echo "Successfully logged in <br>";
             echo "<a href=\"teacher.php\">Go to the teacher home
page</a>";
         }else{
             //if they got the password wrong we can print out the
form again
         ?>
```

It is through understanding that these files work in pairs and that a variable can be set on one page (login.html) and processed by another page (login.php) that I began to understand variables. Any piece of code with a "$" in front of it is a variable. In other words $username is a variable. The combination of username and password variables in the one piece of code are an array, and the array then is posted to the database. In other words the table columns in the database "thesis" correspond to the arrays created in PHP in the login.html and login.php are posted to the database. Through conversations with Cormac I began to make the connections between the data structures of the database tables and the way these data structures are reflected in the PHP use of arrays. Specifically I began to realize that the column titles in the database needed to be matched exactly by the data being submitted through an array (collection of variables) in PHP.

Another amendment to the tutorial files is the inclusion of the headers for the portfolio system with the images and navigation menu. This section is:

```
echo "<html>

<head>
<title>Welcome to Your Portfolio / Research Manager portal
page</title>
```

```
<link rel="stylesheet" type="text/css" href="style.css">

</head>
<table width=\"100%\">
<tr>
<td width=\"25%\">

<img    src=\"images/fractal.jpg\"    width=\"118\"    height=\"118\"
alt=\"Welcome to portfolio building!\" align=\"middle\">

</td>

<td width=\"50%\" valign=\"top\" align=\"middle\">

<h1 align=\"center\" valign=\"top\">
<h1>You have logged in successfully
</h1>

<a name=\"top\"></a></td>

<td width=\"25%\">
</td>


</tr>
<tr      colspan=\"3\"><td      colspan=\"3\"      align=\"center\"><a
href=\"template.php\">To My Projects page</a></td></tr>
</table></body></html>";
```

Note the difference between HTML and PHP is that when embedding a HTML page in PHP the backslash character needs to be included before any " otherwise PHP will not read it correctly.

4.12.3 Cascading Style Sheet – Style.css

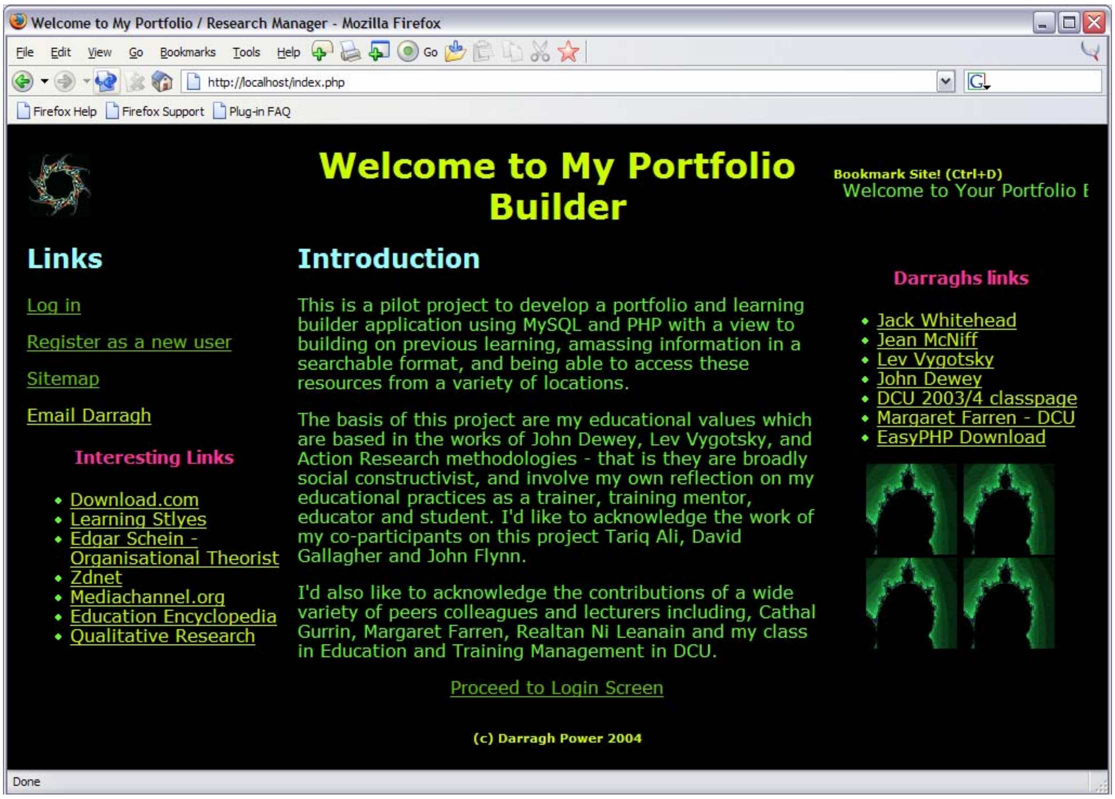
A cascading style sheet defines the font sizes, and styles for all the HTML code on the site. It can be embedded (included in a HTML page), or it can be external, which means that it is stored as a separate page and then linked to each HTML page. I set up an external cascading style sheet at this stage with a view to managing all the styles on the site to ensure consistency and be visually appealing. I have included the file style.css on the accompanying CD. If I had more developed technical skills I would have liked to give users an option on what colour schemes and font / text sizes they would like to see on the site, so they could choose for themselves. This would be in keeping with my values, though my current skill level doesn't allow for it. I would not rule it out in the future as an option though. The cascading style sheet is one of the standard ones on Microsoft FrontPage which is installed on my local computer, and I use as a HTML editor. One of the functions available in Microsoft FrontPage is the generation of standardized external cascading style sheets. I have just copied a standard cascading style sheet into my own site as style.css (.css is the extension for a cascading style sheet. This sets the standard for each of the HTML tags such as the <a> tags for links and <h> tags for text etc. In the future I would hope to offer users the capability to choose their own style sheet, which reflects their own taste, but this is currently beyond my technical abilities. The style sheet is incorporated into each page on the site with this code:

```
<link rel="stylesheet" type="text/css" href="style.css">
```

This appears between the <head></head> tags in the html code.

An example of how the homepage looks with the cascading style sheet is below in **Fig 4.10 Cascading Style Sheet – Home (Index.php)**

Fig 4.10 Cascading Style Sheet – Home (Index.php)


Welcome to My Portfolio / Research Manager - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
Go
http://localhost/index.php
Firefox Help Firefox Support Plug-in FAQ
Done

Welcome to My Portfolio Builder

Bookmark Site! (Ctrl+D)
Welcome to Your Portfolio E

Links

Log in

Register as a new user

Sitemap

Email Darragh

Interesting Links

- Download.com
- Learning Stlyes
- Edgar Schein - Organisational Theorist
- Zdnet
- Mediachannel.org
- Education Encyclopedia
- Qualitative Research

Introduction

This is a pilot project to develop a portfolio and learning builder application using MySQL and PHP with a view to building on previous learning, amassing information in a searchable format, and being able to access these resources from a variety of locations.

The basis of this project are my educational values which are based in the works of John Dewey, Lev Vygotsky, and Action Research methodologies - that is they are broadly social constructivist, and involve my own reflection on my educational practices as a trainer, training mentor, educator and student. I'd like to acknowledge the work of my co-participants on this project Tariq Ali, David Gallagher and John Flynn.

I'd also like to acknowledge the contributions of a wide variety of peers colleagues and lecturers including, Cathal Gurrin, Margaret Farren, Realtan Ni Leanain and my class in Education and Training Management in DCU.

Proceed to Login Screen

Darraghs links

- Jack Whitehead
- Jean McNiff
- Lev Vygotsky
- John Dewey
- DCU 2003/4 classpage
- Margaret Farren - DCU
- EasyPHP Download

(c) Darragh Power 2004

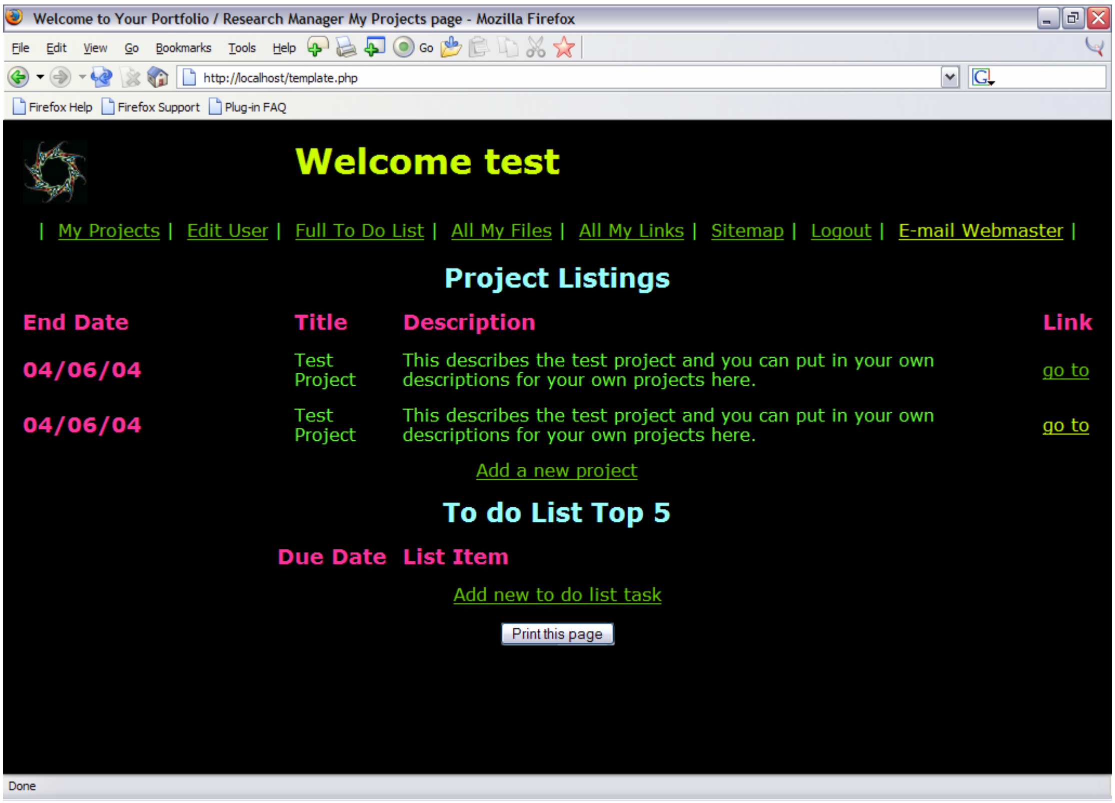
4.12.4 The Design Process

The user manual in Appendix C gives a comprehensive breakdown of each of the pages in the site and how users can access each page and the features of each. The next section focuses on the design process involved in the pages and in particular the technical issues in relation to the design process.

4.12.5 My Projects (Template.php)

To a user the title of this page is My Projects page. Following a successful log in this is the first page that a user will see. The example below shows a user with two projects already uploaded.

Fig 4.11 – Template.php – cascading style sheet


Welcome to Your Portfolio / Research Manager My Projects page - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
Go
http://localhost/template.php
Firefox Help
Firefox Support
Plug-in FAQ
Print this page
Done

Welcome test

| My Projects | Edit User | Full To Do List | All My Files | All My Links | Sitemap | Logout | E-mail Webmaster |

Project Listings

End Date	Title	Description	Link
04/06/04	Test Project	This describes the test project and you can put in your own descriptions for your own projects here.	go to
04/06/04	Test Project	This describes the test project and you can put in your own descriptions for your own projects here.	go to

Add a new project

To do List Top 5

Due Date	List Item

Add new to do list task

The features of this page are that a user when they login can see when their project is due – End Date column. The Title of the project, and a Description of it. The user then has an option to click on the link to the project. Originally this page had three columns on it and a similar layout to index.php. I had to change this for technical reasons rather than design reasons. My technical skills are insufficient to embed the query results from the database within <td></td> tags that are already in existence, and retain the side columns which would have allowed for the 'To Do List' to appear in it. This 'To Do List' is another SQL query, and I had tremendous difficulty in putting more than one SQL query on a page with the page returning blank results sets for the to do list repeatedly. I found this very difficult to resolve. The resolution came by placing the query for the 'to do list' I placed between the <body></body> tags in the HTML code, instead of the <head></head> tags as I had it initially.

The SQL syntax of the query for the projects includes the variable $userID which is set earlier on the page:

```
    $userID=$_SESSION['userid'];
```

This sets the session ID (the logged in user) as the user whose records are to be searched in the query. The query reads:

```
    $query = "SELECT * FROM project WHERE userID='$userID'";
```

This queries the database and the results are processed as follows:

```
<?php
    //loop until we are out of rows
    while ($row = mysql_fetch_array($result, MYSQL_ASSOC)) {
        //$row will contain the current row of the table so we print
out the values of it
        print "<tr>\n";
            //this will get the date but it will be in the format
YYYY-MM-DD
        $dbDate=$row['enddate'];
        //to put it in the format d/m/y we do
        $niceDate=date("d/m/y",strtotime($dbDate));
        print "  <td><h3>$niceDate</h3></td>\n";
        print "  <td>".$row['title']."</td>\n";
        print "  <td>".$row['description']."</td>\n";
            print               "                              <td><a
href=\"project.php?id=".$row['projectID']."\">go to</a></td>\n";
        print "</tr>\n";
    }
?>
```

Again this is an amendment of the source code from the tutorial. The database stores the date in the form Year – Month – Day and the PHP code transforms this to Day – Month – Year before printing it on screen. The results of the query from the database are printed as table rows <tr> and table data <td> within table rows. This query result structure meant that I had to restructure my initial design template as I didn't understand how to put the query into an existing <td> or <tr> tag at this point in time.

4.12.6 Tariq's influence

The ‘To Do List’ / project management aspect was developed with through reflecting on Tariq Ali’s narrative where he states:

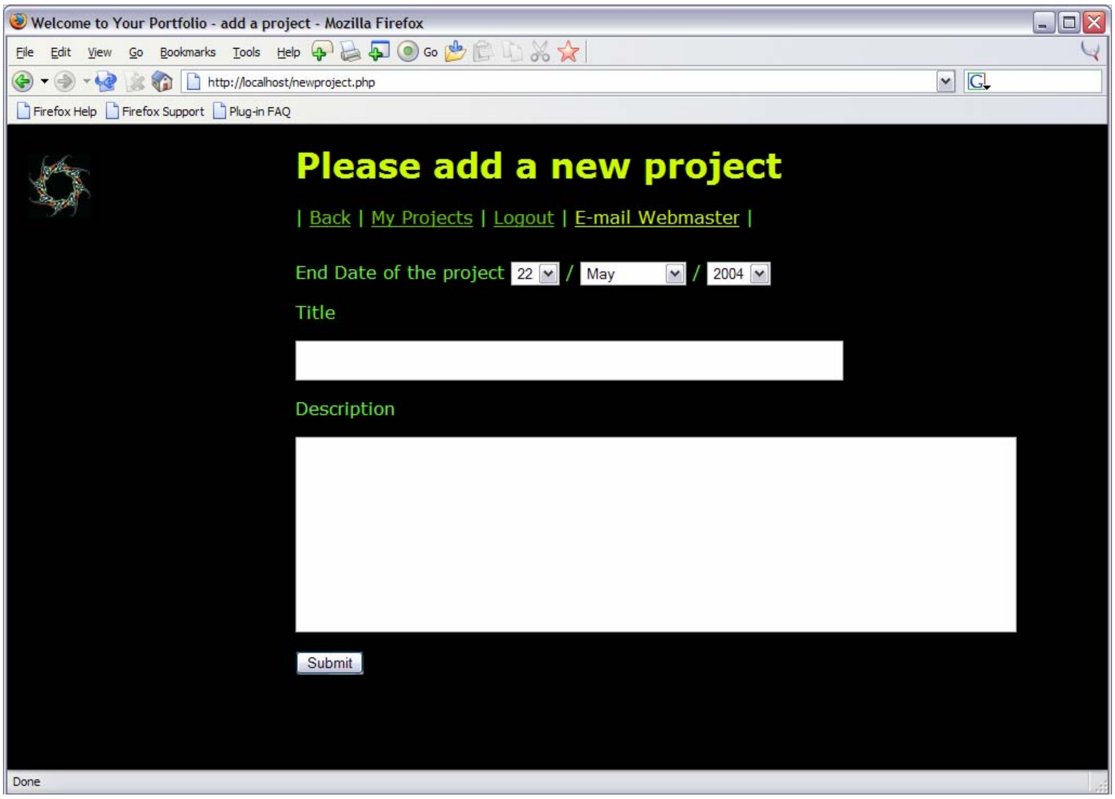
> Also, if feasible, would it be possible to include some sort of project-planning aspect to the portal? I think it may be useful to have a visible project plan on the customized homepage.
>
> (e-mail extract – 31/3/04)

Initially the plan was for the ‘To Do List’ to appear on each page in the same place but the technical skills needed to program this are currently beyond my skills to do. The chief problem I am finding is in running two SQL queries side by side on the one page, with both sets of results being output into a table. The best compromise I can currently come up with is to have the projects shown by their end date.

4.12.7 Adding A Project – newproject.php and addnewproject.php

In a similar way to the login.html and login.php pages the pages newproject.php and addproject.php act as a pair. The newproject.php page sets the variables, and the addproject.php puts the variables in an array and submits them to the database. I adapted this from the tutorial.

Fig 4.12 newproject.php


Welcome to Your Portfolio - add a project - Mozilla Firefox

http://localhost/newproject.php

Please add a new project

| Back | My Projects | Logout | E-mail Webmaster |

End Date of the project 22 / May / 2004

Title

Description

This page allows the user to set the end date for the project which will hopefully work for the users in helping them with their project management. These pages work in the same way as login.html and login.php in that the submission form on login.html sets the variables and the login.php page processes the submission of the variables to the database. The code for the submission in this case is as follows:

```
$query = "INSERT INTO project VALUES(NULL,'$userID','$date','$title',
'$description')";
```

As you can see the array details match the column names on the table ‘projects’ in the database.

This paired approach is evident throughout the design of the artifact which is again a feature of the original tutorial design structure. The same structures that newproject.php and addproject.php use to process data are found in the files, newlink.php and addnewlink.php, newcomment.php and addnewcomment.php, newtodolistitem.php and addnewnotodolistitem.php. The first page sets the variables and the second page processes the details. The one difference is that the links and comments are attached to a project rather than to a user. In other words for the purposes of querying the database the user accesses their comments and links to a particular project. (See Macro1 - tables3basic1.xls for an overview of the database structures)

4.12.8 Uploading Files – myupload.php and uploadmyFile.php

Like the resources and comments that are added by users, files are added to a project. The myupload.php and uploadmyFile.php were slightly different from adding comments or resources, in that the files that are uploaded onto the web are actually stored in a separate folder on the server called userfiles. The database stores the Filepath for the uploaded file. The file size is limited to about 1.5 MB. All of this is similar to the tutorial that Bob introduced me to. The chief difference is in the types of extensions allowable for uploading which are:

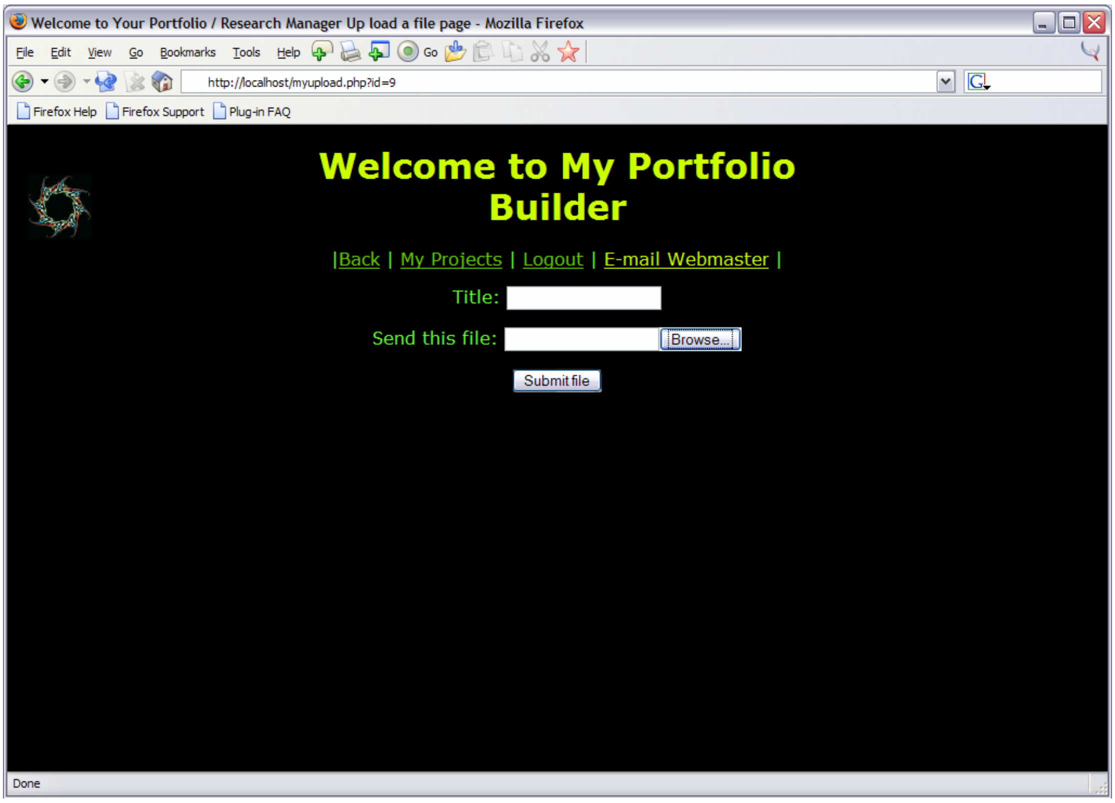
```
.rtf, .pdf, .doc, .jpg, .gif, .xls, .ppt, .pps, .mp3, or .wav
```

The original tutorial allowed:

```
.rtf, .pdf or .txt
```

The page the user sees is:

Fig 4.13 – myupload.php



Welcome to Your Portfolio / Research Manager Up load a file page - Mozilla Firefox

http://localhost/myupload.php?id=9

Welcome to My Portfolio Builder

|Back | My Projects | Logout | E-mail Webmaster |

Title:

Send this file:

Done

I struggled with getting these files uploaded into the correct directory for a number of days. I discussed this with Cathal on the 4/6/04 and as a result of this conversation I edited the following piece of code in uploadmyfile.php which processes the form shown above:

```
$uploaddir="C:\Program Files\EasyPHP1-7\www\userfiles\";
```

So that it read:

```
$uploaddir="C:\Program Files\EasyPHP1-7\www\userfiles\\";
```

The addition of the second backslash is necessary for the upload to take place correctly.

4.13 Cycle Three

4.13.1 Becoming more technically proficient

The stage of learning PHP is one which is characterized by trial and error, and a lot of time spent trying to get the pages to work properly. As is evident from the progress above from March to early June I was learning a number of things simultaneously, in terms of database design, SQL queries and also PHP language and syntax.

4.13.2 Passing projectID numbers through URLs

At this stage I embarked on another cycle of learning how to query the database properly and ensure that multiple queries could appear on a page. Up to this point just getting the results of HTML forms passed to the database has been difficult, particularly for the add files, comments and resources. The chief reason for this difficulty is that I didn't know how to pass the project ID around from page to page once a user had selected that particular project. On the template.php page the query result I discussed earlier makes the projectID a link. The section in question is highlighted.

```
<?php
    //loop until we are out of rows
    while ($row = mysql_fetch_array($result, MYSQL_ASSOC)) {
        //$row will contain the current row of the table so we print
out the values of it
        print "<tr>\n";

            //this will get the date but it will be in the format
YYYY-MM-DD
        $dbDate=$row['enddate'];
        //to put it in the format d/m/y we do
        $niceDate=date("d/m/y",strtotime($dbDate));
        print "  <td><h3>$niceDate</h3></td>\n";
        print "  <td>".$row['title']."</td>\n";
        print "  <td>".$row['description']."</td>\n";
            print               "                              <td><a
href=\"project.php?id=".$row['projectID']."\">go to</a></td>\n";
        print "</tr>\n";
    }
?>
```

This makes the URL to the linked page read as:

http://localhost/project.php?id=9

The link therefore incorporates the projectID. Now the next thing I needed to understand how to do was to pick up the ID and incorporate it into queries in the linked page. Again I discussed this with Cathal on the 4/6/04 and through trial and error we figured out that in the page that is being linked, in this case the project.php page. The key to collecting the ID number in the target page is the PHP code:

```
$projectID = $_GET['id'];
```

This code takes the projectID number from the URL and sets it as the variable $projectID, which then allows this variable to be used in all the links from the project.php page. In other words the files, resources and comments are linked through the URL to the projectID in the same way as the link to add a comment from the project.php page demonstrates.

```
<?php echo "<a href=\"newcomment.php?id=".$projectID."\">Add a
comment</a>"; ?>
```

I then had to understand how to incorporate this projectID into the SQL syntax in order to update and read the database.

4.13.3 Query construction in SQL

This is not actually that difficult as long as the database structures are known and clearly understood. An example from addnewcomment.php which processes the form newcomment.php is:

```
$query = "INSERT INTO comments VALUES(NULL,'$projectID','$comment',
'$date')";
```

This means that the projectID now accompanies the comment when it is submitted to the database. Now that the projectID is linked correctly to the comment it allows for this query to appear on the project.php?id=(whatever the projectID is) which brings up all the comments in relation to this project, ordered by date. (The section “$query” is the SQL for this).

```
$query = "SELECT * FROM comments WHERE projectID='$projectID' ORDER
BY date";
```

4.13.4 Formatting the output of the queries and multiple queries per page.

At this time (6/6/04) the artifact functioned reasonably well in that users could add projects, within projects they could add resources, files and comments, and the additions would be correctly stored in the database under the correct user and projectIDs. I could now turn my attention to getting the queries constructed and formatting the output of the queries. At this time I am conscious of the fact that the Tariq and David both expected formatted output and clear search results. Through experimenting with using various tags in a trial and error way I managed to figure out that the query results could be printed within <p> (paragraph) tags and also <ul> (Un ordered list) tags, which allowed me to display query results within table structures like a row or a column, without having to restructure the design of the page.

The artifact was now developed to the point where the ‘alpha’ version could be user tested. David Dick (2002) suggests that many products are developed to a basic level, an ‘alpha’ version, which the designers are aware will have design flaws, but which are sufficiently functional to be tested.