

EXAMINATION RESULTS DISTRIBUTION SYSTEM

By:

KIIZA GODWIN .K

REGISTRATION NUMBER: BIT/14287/62/DU

&

MUKAMA AUSTIN

REGISTRATION NUMBER: BIT/7653/51/DU

**A PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT AS A
REQUIREMENT FOR THE AWARD OF A BACHELORS DEGREE IN
INFORMATION TECHNOLOGY OF KAMPALA INTERNATIONAL
UNIVERSITY**

JUNE 2010

DECLARATION

We, **KIIZA GODWIN .K and MUKAMA AUSTIN**, hereby declare that this is our own project report, done to the best of our knowledge and has never been presented to any university for the award of any degree.

Signature.....

KIIZA GODWIN KAMUTERA

Date.....15/05/2010.....

Signature.....

MUKAMA AUSTIN

Date.....16/06/2010.....

APPROVAL

This project report has been submitted for examination with my approval as a university supervisor

Signature.....

MR. KAMPORORO EZRA

KAMPALA INTERNATIONAL UNIVERSITY

Date.....17/08/10.....

DEDICATION

I KIIZA GODWIN KAMUTERA do dedicate this work to my beloved mother Christine Magezi, my uncle Gregory Magezi, my friends Norbert Nuwagaba, Kanyike Ronald, Steven Ojok, my Colleagues at Ahmos Group, Mr. Ezedin Mohamed Ahmed, Mr. Muganzi Herbert, Mr. Hassan Basshir, sisters and brothers, for their unflinching support and encouragement, love and support.

I MUKAMA AUSTIN do dedicate this work to my dear Loving and caring mother Namatovu Imamaculate, my father Moses Nkutu, My uncles Conrad Nkutu, Mugabe Julius, My friends Nakiyingi Jescah, Lubwama Frank, Kairaka Chris, My brothers Nasser Kirunda, Dhakaba Abu-Baker, Nkutu Shaban, & my Sister Namuwaya Sophie.

ACKNOWLEDGEMENT

Our gratitude goes to our *Supervisor* who guided us and corrected our work. If not because of his effort, this work could not have been as it is now.

We also thank all the Lecturers and administrators of Kampala International University for the concern and friendship.

We are grateful to all our colleagues, for their love, concern, support solidarity, the good and sometimes difficult moments we experienced together in the three years at Kampala International University.

We are especially grateful to the following: Norbert Nuwagaba, Ojok Steven, Father Dan, Lora, Asimwe Moses and my Uncle Robert Muganwa, Conrad Nkutu, to mention but a few. We wish them all the best in their lives and place of work.

We are also grateful to our Mothers Christine Magezi & Namatovu Immaculate for their love and support, May the almighty God bless and reward them abundantly.

ABSTRACT

This project report presents the proceedings of the design and implementation of an examination results distribution system for the institute of Certified Public Accountants of Uganda.

The project aimed at facilitating the Examination Results distribution for the Institute of Certified Public Accountants of Uganda.

The data used was obtained through interviews carried out face-to-face, reading through records and observation.

The system was made to operate using the functionality of a database driven website.

Specified PHP commands connect to the MySQL database and request the content that belongs to the Web page.

The implementation and design of this system involved the development of the interface and the database.

The system was able to authenticate every user to the website by using a login form in which a user name and password are entered and making sure the system is secure and functioning to the purpose of its creation.

The evaluation stage involved cross checking the developed system with the real world values that make sure it functions as required in relation to the set objectives.

The recommendations were also suggested for the proper functioning of the system.

It is important to note that the system can only work for the mentioned institute.

ABSTRACT

This project report presents the proceedings of the design and implementation of an examination results distribution system for the institute of Certified Public Accountants of Uganda.

The project aimed at facilitating the Examination Results distribution for the Institute of Certified Public Accountants of Uganda.

The data used was obtained through interviews carried out face-to-face, reading through records and observation.

The system was made to operate using the functionality of a database driven website.

Specified PHP commands connect to the MySQL database and request the content that belongs to the Web page.

The implementation and design of this system involved the development of the interface and the database.

The system was able to authenticate every user to the website by using a login form in which a user name and password are entered and making sure the system is secure and functioning to the purpose of its creation.

The evaluation stage involved cross checking the developed system with the real world values that make sure it functions as required in relation to the set objectives.

The recommendations were also suggested for the proper functioning of the system.

It is important to note that the system can only work for the mentioned institute.

TABLE OF CONTENTS

1.0	Declaration	(i)
2.0	Approval.....	(i)
3.0	Dedication	(ii)
4.0	Acknowledgement.....	(iii)
5.0	Abstract	(iv)
6.0	Table of contents	(v)

Chapter one: INTRODUCTION

1.0	Background	1
2.0	Statement of the Problem	3
3.0	Objectives of the project	3
4.0	Significance of the project	4
5.0	Scope of the project.....	4
6.0	Methodology	5

Chapter two: LITERATURE REVIEW

1.0	An information system.....	7
2.0	Databases	10
3.0	Database management systems	10
4.0	Database driven website	10
5.0	PHP	11
6.0	MySQL	12

Chapter three: METHODOLOGY

1.0	Introduction	14
2.0	Installation and configuration of PHP and MySQL	14
3.0	Installation process.....	15
4.0	How the MySQL server works	17
5.0	How Apache web Server works.....	17
6.0	Conclusion	17

TABLE OF CONTENTS

1.0	Declaration	(i)
2.0	Approval.....	(i)
3.0	Dedication	(ii)
4.0	Acknowledgement.....	(iii)
5.0	Abstract	(iv)
6.0	Table of contents.....	(v)

Chapter one: INTRODUCTION

1.0	Background	1
2.0	Statement of the Problem	3
3.0	Objectives of the project	3
4.0	Significance of the project	4
5.0	Scope of the project.....	4
6.0	Methodology	5

Chapter two: LITERATURE REVIEW

1.0	An information system.....	7
2.0	Databases	10
3.0	Database management systems.....	10
4.0	Database driven website	10
5.0	PHP	11
6.0	MySQL	12

Chapter three: METHODOLOGY

1.0	Introduction	14
2.0	Installation and configuration of PHP and MySQL.....	14
3.0	Installation process.....	15
4.0	How the MySQL server works	17
5.0	How Apache web Server works.....	17
6.0	Conclusion	17

Chapter four: SYSTEM REQUIREMENTS

1.0 System specification..... 18

2.0 Software specification..... 18

3.0 Hardware specification..... 18

Chapter four: SYSTEM DESIGN

1.0 Designing tables..... 20

2.0 Interface design 26

Chapter five: SYSTEM IMPLEMENTATION

1.0 Creating the database 28

2.0 Creation of the tables 29

3.0 Database integrity..... 30

4.0 Integrity at table level..... 31

5.0 Interface level integrity 31

6.0 Interface implementation 35

Chapter six: EVALUATION, RECOMMENDATION AND CONCLUSION

1.0 Evaluation 38

2.0 Recommendation 38

3.0 Conclusion 39

References 40

Appendices 41

CHAPTER ONE

INTRODUCTION

1.0. Background

Institute of Certified Public Accountants of Uganda (ICPAU) is a member of the international Federation of Accountants (IFSC) and the Eastern Central and Southern African Federation of Accounts (ECSAFA). ICPAU is located on 42nd Bukoto Street-Kololo, P.O.BOX 12461, Kampala, Uganda.

It started courses such as Certified Public Accountants (Uganda) CPA (U), Accounting Technician Certificate (Uganda) ATC (U).

ICPAU is a professional accountancy body established by statute No.12 of 1992. Only members of ICPAU are allowed to practice accountancy, this includes auditing in Uganda. The *mission* of the institute of Certified Public Accountants of Uganda is to develop and promote the accountancy profession in Uganda and beyond through an internationally recognized qualification ensuring high ethical standards and professional accounting and auditing standards.

- To regulate and maintain the Standard of Accountancy in Uganda.
- To prescribe or regulate the conduct of accountants in Uganda.

ICPAU is a member of the international Federation of Accountants (IFSC) and the Eastern Central and Southern African Federation of Accounts (ECSAFA)

It founded courses such as CPA (U) Certified Public Accountants (Uganda), ATC (U) Accounting Technician Certificate (Uganda).

ICPAU is not engaged in the provision of tuition. This function is carried out by the various institutions. Students are advised of the institutions teaching CPA (U) and ATC (U).

It admits any person with a Ugandan certificate of education (UACE) with three principal passes where one must be of grade “D” or the equivalent, or two principal passes of at least grade “D” or the equivalent plus a Uganda certificate of education (UCE) with at least 5 credits including English language and mathematics.

It also admits any person with a degree from any recognized university. If the qualification is obtained from outside Uganda, it must be accredited by the respective government.

Exemptions may, on application be registered students holding recognized degrees, other certificates or passes of subjects of other recognized accountancy bodies, be granted on a subject for subject basis in level one to two of CPA (U).

This institute recommends four distinct methods of study;

- A full time training at the training institution.
- Part-time at the training institution
- Private study
- Correspondence

2.0. Problem statement

For almost 12 years, ICPAU has been using manual methods to distribute examination results to thousands of students. The number of students by then was less than 50 and so manual method was convenient by then to the institute. The number of students registering with the institute has gradually been increasing over the past years. The institute registers over 8000 students every academic year. It can no longer be efficient using the manual method. Also, students wait unnecessarily long before obtaining their results. Summing the whole matter up, the present system as a whole is time consuming and error prone.

3.0. Objectives of the project

With the above mentioned problems faced by the students of the institute, the study is intended to develop an information system that will attempt to correctly perform the following tasks;-

- Make every student's results available on the internet for easy access.
- Attend to students examination complaints as fast as possible.
- To ensure that examination results are received by the right person through proper authentication.
- To provide a way for the administrator to carry out administrative tasks such as updating the database, deleting and adding students to the database.
- To ensure that the system allows students to log onto the system, then carries out authentication to decide whether or not the user should be granted access to the system.
- To ensure that the student's complaints are received and responded to in time.

4.0. Significance of the project

The project intended to develop a system that would greatly improve the image of the institute as an examination body towards its students in the following areas.

Fast and accurate results capture will be realized together with reliable data retrieval where necessary.

Timely, correct and regular updates about the changes in the institute's policies.

This study is also aimed at cutting on the expenditure incurred in distributing examination results to the various students distributed all over the country. This process is currently done manually by distributing mails to each student. Some mails end up getting to wrong destination.

There is also congestion at the institute whenever examination results are released as a result of many students storming the institute to collect their results.

The study is intended to devise a system that will quickly respond and resolve to most students complaints in the shortest time possible.

5.0. Scope of the project

There are many areas in the students' affairs management process of the institute that are still manual and need to be computerized. However in this study, concentration was put on the process of examination results distribution. Examination results for both CPA (U) and ATC - (U) have been put into consideration in the project.

6.0. Methodology

The remedy of all the problems sighted in the problem statement earlier on will be through a design of a system to help automate the examination results distribution

The system will be in form of a database driven website. This website will be programmed to generate dynamic web pages depending on the requests issued by either the student or the administrator.

This is what will happen when a user visits a page on the icpau website:

- The visitor's Web browser requests the Web page using a standard URL
- The Web server software (Apache, IIS, or whatever) recognizes that the requested file is a PHP script, and so the server interprets the file using its PHP plug-in, before responding to the page request.
- Specified PHP commands connect to the MySQL database and request the content that belongs in the Web page.
- The MySQL database responds by sending the requested content to the PHP script.
- The PHP script stores the content into one or more PHP variables, and then uses the echo function to output the content as part of the Web page.
- The PHP plug-in finishes up by handing a copy of the HTML it has created to the Web server.
- The Web server sends the HTML to the Web browser as it would a plain HTML file, except that instead of coming directly from an HTML file, the page is the output provided by the PHP plug-in.

The tasks that each visitor can carryout will be determined by whether the visitor is an administrator or a student. On opening the web page, each user will be required to log in using their username and password which on submission will be verified and authenticated by the server.

The administrator will be able to carry out the following tasks:

- Adding users
- Deleting users
- Viewing all users details
- Editing user's details

Each student will be able to carry out the following tasks:

- Log in and view only his examination results.
- Change their password

In summary, the system will be able to perform the following:

- Accept new users
- Decide whether or not to allow a student to access the website
- Collect submitted data from the forms and store them in the database
- Allow each user to carryout operations they are entitled to depending on their privileges.

With the mentioned procedures followed and implemented as required, the problem at hand will significantly be solved.

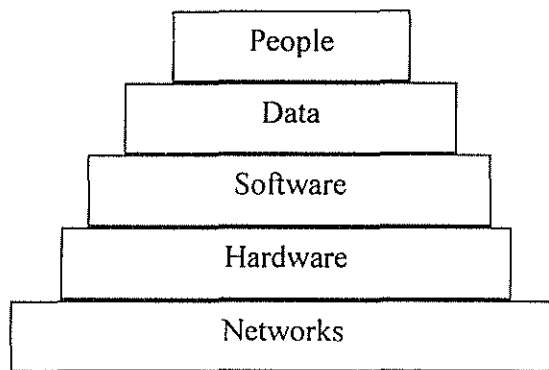
CHAPTER TWO

LITERATURE REVIEW

As explained in the previous chapter, this project is an information system which automates the examination results distribution.

2.0 An information system

According to James [1], an information system is an organized combination of people, hardware, software, communication networks and data resources that collects, transforms and disseminates information in an organization.



People have relied on information systems to communicate with each other using a variety of physical devices (hardware), information processing instructions and procedures (software), communication channels (networks), and stores data (data resource), since the dawn of civilization.

Today's end users rely on many types of information systems; including simple manual (paper and pencil) hardware devices and informal (word-of-mouth) communication channels. However with the advance of computers, information system refers to computer based information systems that use hardware, software, telecommunication networks, computer based data management techniques and other forms of information technology to transform data resource into a variety of information products.

There are several factors that may lead to the success or failure of an information system development.

The top five factors that lead to the success of an information system include:

- User involvement in the development(Joint Application Development)
- Executive management involvement
- Clear statement of requirements
- Proper planning
- Realistic expectations

The top five factors that may lead to the failure of an information system include:

- Lack of user input
- Incomplete requirements and specifications
- Changing requirements and specifications
- Lack of executive support
- Technological incompetence

Companies will too connect to the internet as a marketing channel, a place to publish information about themselves and their products as well as communicate with their customers and business partners. But the internet is capable of far more.

Viewing the internet as a merely a gigantic bulletin board or electronic mail system barely misses the point. With its extraordinary scope and growth, this global network of networks, true future will be to support distributed applications across companies and geographical boundaries.

Process of building an information system

According to Igor [2], there are a number of steps involved in developing an information system. These include:

1. **Development process**, which is what we actually do to build a system.

2. **Management process**, which concerns itself with organizing work, availing resources and monitoring progress of work.
3. **Support processes**, provide developers with computers to manage documentation and facilitate communication among people in various groups on the project.

Development process

Methods of building computer based information systems vary from adhoc to highly structured. Adhoc approach is to build simple personal support system; for instance, a spread sheet budget for a manager.

A more formal approach is needed to build a larger system which is becoming increasingly complex.

There is also an increasing demand to build systems that are correct and do what is expected of them. A formal rather than adhoc approach is needed to achieve correctness.

System requirements must be exactly defined before construction is started. Thus there is similarity between computer based systems and other kinds of systems.

Firstly, there is the concept of a dream system to solve a business problem. This is discussed and analyzed to clearly identify the user requirements while taking into account various needs and constraints.

Requirements are then used to produce a system specification which has the same purpose as a house plan, to specify how the future information system will satisfy user requirements. It's important to distinguish between the initial concept and the resulting system specifications; both play an important role in the system development.

The first comes up with an idea and the second is the practical realization of that idea.

In between there may be other processes like carrying out detailed analysis of the system and identifying detailed user requirements.

There are however some differences to distinguish the building of an information system from other systems.

There are no universal standards for an information system. Information system requirements tend to change as development proceeds, such changes because of a deadline, can bypass the original plan which can lead to eventual breakdown of the whole process.

There is however, increasing emphasis on developing universal standards and managing the process of change when building computer systems.

2.0 Databases

According to Wikipedia [4], a database is an organized collection of data.

It also defines a database as a collection of records stored in a computer in a systematic way, so that a computer program can consult it to answer questions.

Databases are an important area in the information technology field today. As the amount of data to be stored and managed grows, efficiency is only possible if the data is stored in a database.

According to Cary N. Prague and Micheal R. Irwin [3], a database is a computer term for a collection of information concerning a particular topic. Databases help you organize this related information in a logical fashion for easy access and retrieval."

3.0 Database management system

According to Cary N. Prague and Micheal R. Irwin [3], a database management system is a system which stores, retrieves information from a database by querying and helps automate repetitive tasks such as performing inventory control and scheduling.

In summary a database management system is simply a program used to manage and query a database.

4.0 Database driven website

Louis and Peter [8] defined a database driven approach to web development simply as the use of a database for storing a web site's content. It involves the separation of a site's content (text and images) from the design of the website.

The design is most typically stored in template files. When a site visitor requests a page by clicking a link or a button, a script runs and populates a template with the appropriate information from the database before displaying the populated template as an HTML page to the customer. Therefore database driven websites are referred to as dynamic websites.

On the other hand a static website is comprised of a series of HTML files. These files are self contained documents that appear the same to all users, regardless of where, how, when or why they are viewing the document.

A single static file holds all of the data for a particular page and the design and content are intertwined. When a site visitor requests a page, the HTML file is passed from the server to the visitor's browser. On a static site, visitors are simply viewing documents over the internet.

The need for dynamic content

The Web is no longer static; it's dynamic. As the information content of the Web grows, so does the need to make Web sites more dynamic. Think of an e-shop that has 1,000 products. The owner has to create 1,000 Web pages (one for each product), and whenever anything changes, the owner has to change all those pages.

Wouldn't it be easier to have only one page that created and served the content on the fly from the information about the products stored in a database, depending on the client request?

Nowadays sites have to change constantly and provide up-to-date news, information, stock prices, and customized pages. PHP and MySQL are two ways to make the site dynamic.

5.0 PHP

According to Matt [6] PHP is a recursive word abbreviated as PHP: HyperText Preprocessor. It is a server-side scripting language written in an HTML context. Unlike an ordinary HTML page, a PHP script is not sent directly to a client by the server; instead, it is parsed by the PHP binary or module. HTML elements in the script are left alone, but PHP code is interpreted and executed. PHP code in a script can query databases, create images, read and write files and talk to remote servers.

Features of PHP?

- ✓ PHP4 provides native support for user sessions, using both cookies and the query string. You can "register" a variable with a session, and then access the same variable name and value in subsequent user requests.

- ✓ A new comparison operator (===) has been introduced that tests for equivalence of type as well as equivalence of value.
- ✓ New associative arrays containing server and environmental variables have been made available, as well as a variable that holds information about uploaded files.
- ✓ PHP4 provides built-in support for both Java and XML.
- ✓ The language now includes the Boolean data type.
- ✓ A particularly useful feature of PHP3 was the capability to name form elements as if they were elements in an array. The elements' names and values are then made available to the code in array form. This feature has been extended to support multidimensional arrays.
- ✓ Support for object-oriented programming was somewhat rudimentary in PHP.
This is significantly extended in PHP4; for example, it is now possible to call an overridden method from a child class.

Why choose PHP?

In summary, these are the reasons as to why PHP was chosen to implement the project.

There are some compelling reasons as why PHP was used to implement this project.

As an open source product, PHP is well supported by a talented production team and a committed user community. Furthermore, PHP can be run on all the major operating systems with most servers.

Further more PHP allows you to separate HTML code from scripted elements.

In many instances, you will be able to separate the coding stage of a project from the design and build stages. Not only can this make life easier for you as a programmer, it also can remove obstacles that stand in the way of effective and flexible design.

PHP is Portable. By this I mean that PHP is designed to run on many operating systems and to cooperate with many servers and databases. You can build for a UNIX environment and shift your work to NT without a problem. You can test a project with Personal Web Server and install it on a UNIX system running on PHP as an Apache module.

PHP is designed to run on many operating systems and to cooperate with many servers and databases. You can build for a UNIX environment and shift your work to NT without a problem. You can test a project with Personal Web Server and install it on a UNIX system running on PHP as an Apache module.

6.0 MySQL

MySQL is a small, compact database server ideal for moderate applications. In addition to supporting standard SQL (ANSI), it compiles on a number of platforms and has multithreading abilities on Unix servers, which make for great performance. According to Md. Ashraful Anam [8] MySQL is the standard query language for interacting with databases. MySQL is an open source, SQL database server that is fast.

CHAPTER THREE

METHODOLOGY

1.0 Introduction

This chapter explains out the procedures and methods used to implement the project and ensuring that the project meets the intended functions.

The chapter also gives the prerequisites for implementing the project and the requirements for the smooth running of the system

The following were required in the implementation of the system.

- A web server

The web server used was Apache HTTP server.

- A database server

The database server used was MySql server.

The above mentioned prerequisites were acquired by the installation of the WAMP package which contains both the Apache HTTP server and MySql server.

2.0 Installation and configuration of PHP and MySQL

WAMP5 package needs to be installed on the server machine.

WAMP5 is an installer for Windows. It installs a complete WAMP solution on your computer.

WAMP stands for: Windows - Apache - MySQL- PHP5

WAMP5 automatically configures them to work together, wherever you install them on the local drive.

The following software is installed:

Apache 1.3.x.

- PHP 5.x.x
- MySQL 4.x.x
- PHPmyadmin
- SQLitemanager
- Wampserver service manager

WAMP5 comes with a service manager as an icon tray. This manager allows you to fully control the server and local projects.

WAMP5 is not designed for production but as a development environment. With WAMP5, you are able to create scripts locally on the computer, test them and then upload them on the production server.

WAMP5 is configured with Apache and MySQL servers with default configuration files to have the most standard platform. However the configurations can be customized to meet the programmers requirements.

During installation, a "www" directory is created in WAMP5's directory. The script files (PHP files) in this directory and access your localhost threw an Internet Windows or the service manager (icon tray).

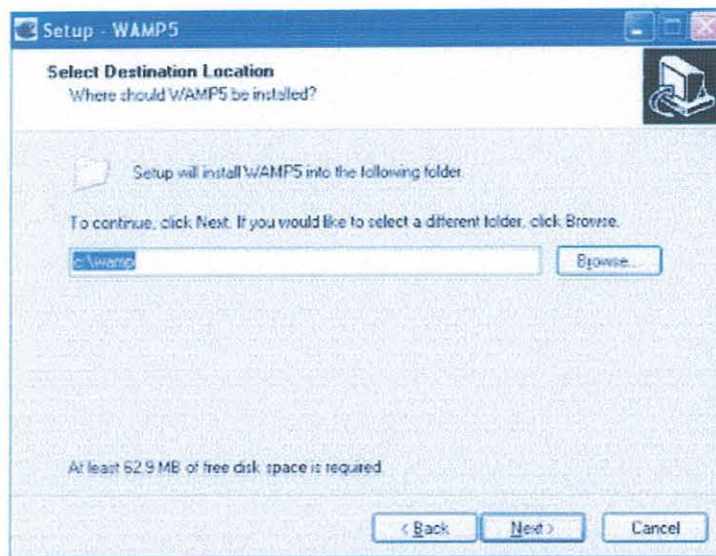
3.0 Installation process

Download the latest release of WAMP5 from the following WAMPSEVER site:

<http://www.en.wampserver.com> (English Website)

Double-click on the file to start installing.

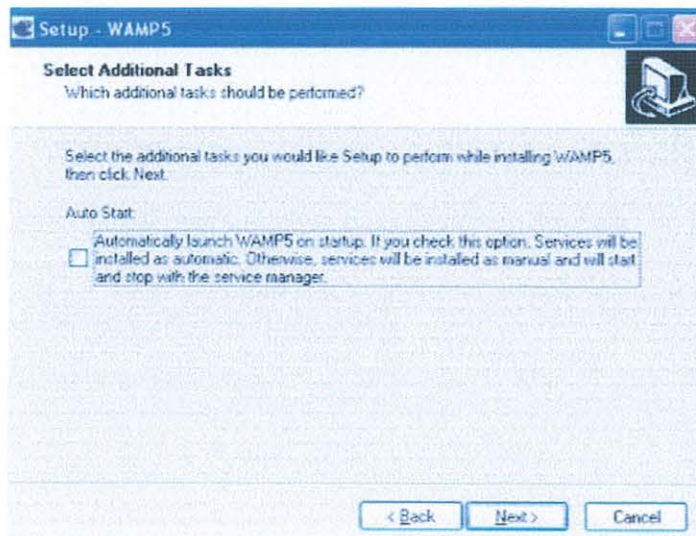
WAMP5 will ask you where to install. You can install wherever you want, WAMP5 builds configuration files to reflect your choice. It is important to note that some scripts and modules do not like long names and special characters like spaces. It is recommended to leave the default directory c:\wamp



WAMP5 will then ask you if want to install the option "Auto Start".

If you check this option, services will be installed as automatic and a link to the service manager will be copied to your startup menu.

Otherwise, services will be installed as manual and will start and stop with the service manager.



After installation, you can directly start using your server. You should first change MySQL's root default password. You can do it through PHPmyadmin.

4.0 How the MySql Server works

The database server (MySql) obtains SQL statements supplied by the Apache server, executes them and returns the results back to the apache server. It is only the Apache server that can dispatch these SQL statements to the MySql server. The MySql server in return queries the database in accordance with the request of the Apache server. This prevents malicious scripts or unauthorized SQL commands from outside the system.

The Apache server and MySql server work as one entity. This is because the details of the various operations are hidden from the programmer. The programmer only works with the Apache server which then communicates with the database server.

5.0 How Apache web server works

The web server will parse PHP statements found in the web pages and then dispatch HTML tags and contents to the web browser that requests a specified page.

According to Oxford Advanced learner's dictionary [9], the term PARSE means to divide a statement into parts and then studying each part critically.

Apache web server is able to carry out database manipulations tasks by making use of a set of inbuilt functions of the MySql family. These functions usually take MySql query statements as arguments and they simply pass these queries to the database server which then executes them like it would to normal SQL statements.

6.0 Conclusion

All the above mentioned components were integrated to come up with a system that was used to fulfill the specified purpose.

CHAPTER THREE

SYSTEM REQUIREMENTS

1.0. Systems Specification

This chapter contains the requirements necessary for running this information system. It includes the minimum and recommended requirements but its' advisable to choose the recommended rather than the minimum to ensure optimum performance.

2.0 Software specifications

The server should have the following software packages installed in order to facilitate the proper working of the information system.

- MySql software
- Apache server software
- PHP software
- Java so
- Windows NT operating system

The Client should have the following software packages installed in order to access the information system.

- Windows NT operating system
- Java enabled browser like Netscape

3.0. Hardware specifications

The Server should hardware requirements: -

- Minimum of 256MB RAM
- Minimum, of 40GB Hard drive server
- Minimum of Pentium IV or equivalent processor

A connection to a network that spans the globe, which is currently the internet

The Clients should have the following hardware requirements

- Minimum of 128MB RAM for optimal performance
- Standard keyboard and mouse
- Coloured monitor

CHAPTER FOUR

SYSTEM DESIGN

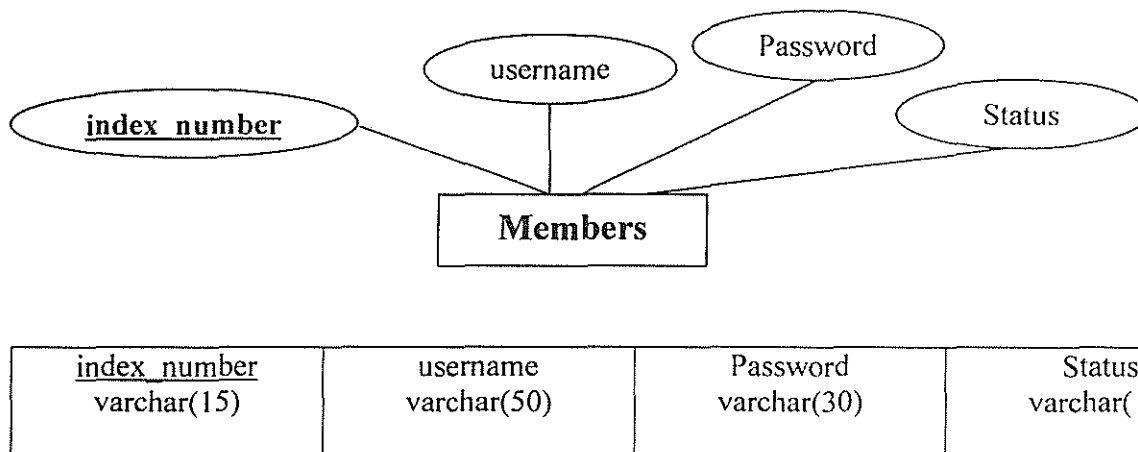
This chapter will in detail explain the system design, the various components of the system and how the system is generally expected to fulfill its functions.

1.0. Designing Tables

The icpau_database database consists of a total of 35 tables. These tables were all created at design time.

The members table

This table generally consists of the users log in information.



The figure above explains the design of the members table

The members table will record the students' username and passwords. Each username and password has a unique index number. Index number is the primary key in all the tables. The system bases on the index number to retrieve a specific student's information.

Status is a field that checks whether the visitor to the site is a student or an administrator.

You can have as many administrators as possible in order to quicken processes.

User rights are assigned to the visitor depending on the status of the visitor.

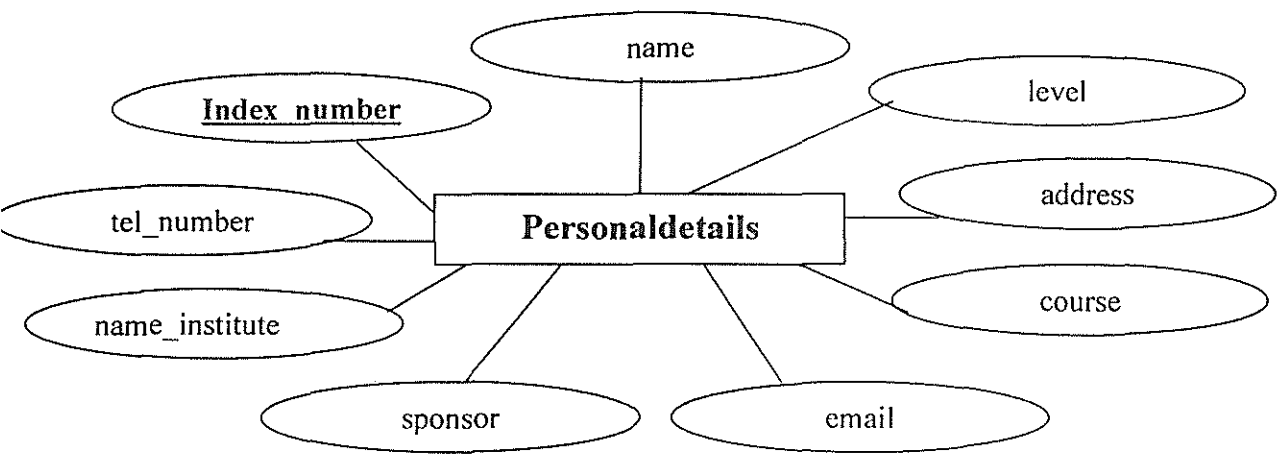
A student can only view his specific information, he or she cant edit, delete, view other students' results, update and creating new entries.

An administrator has all privileges to the database. He or she can delete, update, view students' information and add new entries.

Personaldetails table

This table consists of the students general information such as the name, address, name of the institute, telephone contacts, course offered, email address and the sponsor of the student.

Course field is very vital because it determines which course tables are to be retrieved. There are two courses put into consideration. That is CPA(U) and ATC(U).



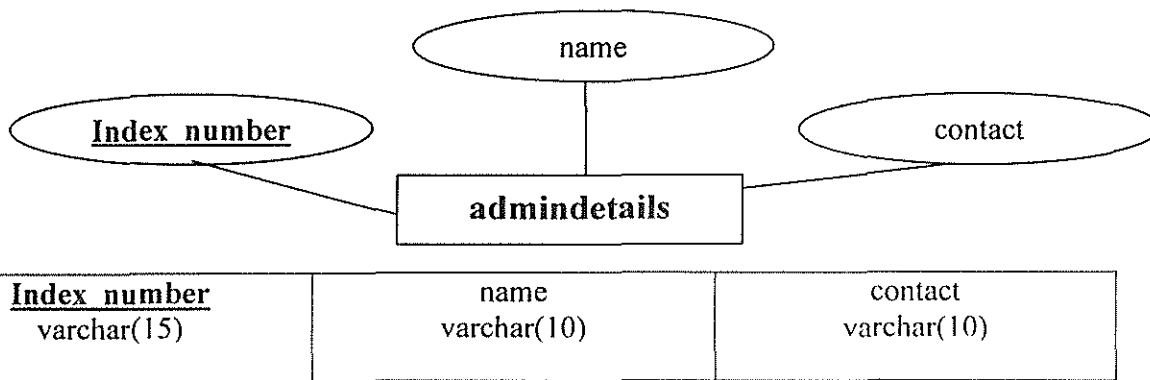
<u>Index number</u>	name	level	address	course	email	sponsor	name_institute	tel_number
varchar(15)	varchar(10)	varchar(10)	varchar(10)	varchar(10)	varchar(10)	varchar(10)	varchar(10)	varchar(10)

The figure above explains the design of the personaldetails table

admindetails table

This table consists of the administrators name and contacts. The details of this table are retrieved depending on the status of the visitor to the site. They are only retrieved if the administrator logs in.

The design of this table is as shown below.



ATC(U) tables

ATC as a course consists of 12 course units. Each course unit has a particular table which stores the students' index number who sat for that course unit, the mark obtained by the student and the date of sitting of the paper.

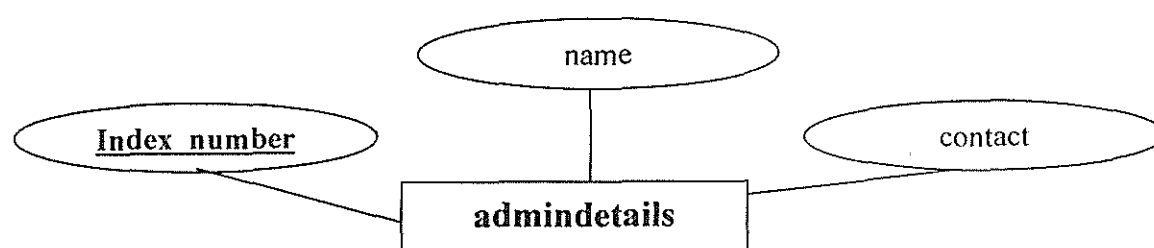
Each course unit has a table assigned to it.

Table	Course unit
atc1	Principles of Accounting I
atc2	Principles of Law I
atc3	Business Math & Statistics
atc4	Commercial Environment
atc5	Principles of Accounting II
atc6	Principles of Law II
atc7	Economics
atc8	Information Systems
atc9	Introduction to Mgt Accounting
atc10	Principles of Taxation
atc11	Business Management
atc12	Introduction to Fin.Reporting

admindetails table

This table consists of the administrators name and contacts. The details of this table are retrieved depending on the status of the visitor to the site. They are only retrieved if the administrator logs in.

The design of this table is as shown below.



<u>Index number</u> varchar(15)	name varchar(10)	contact varchar(10)
------------------------------------	---------------------	------------------------

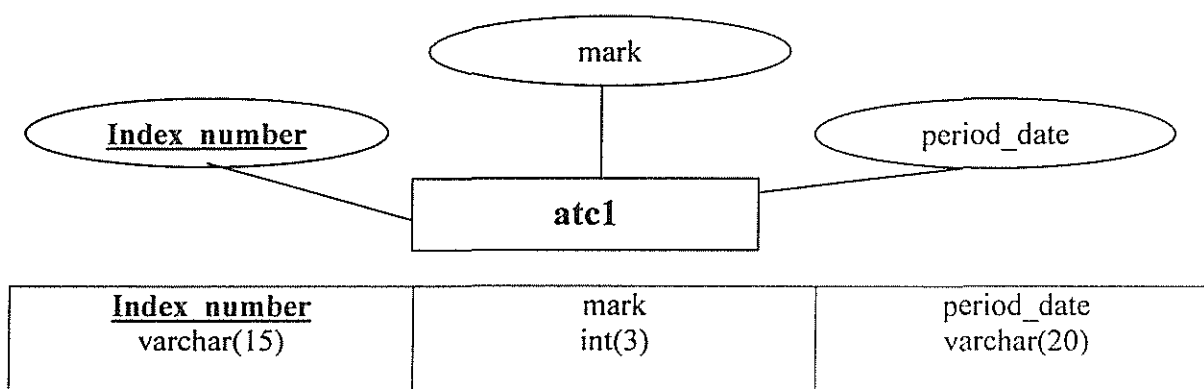
ATC(U) tables

ATC as a course consists of 12 course units. Each course unit has a particular table which stores the students' index number who sat for that course unit, the mark obtained by the student and the date of sitting of the paper.

Each course unit has a table assigned to it.

Table	Course unit
atc1	Principles of Accounting I
atc2	Principles of Law I
atc3	Business Math & Statistics
atc4	Commercial Environment
atc5	Principles of Accounting II
atc6	Principles of Law II
atc7	Economics
atc8	Information Systems
atc9	Introduction to Mgt Accounting
atc10	Principles of Taxation
atc11	Business Management
atc12	Introduction to Fin.Reporting

Considering one of the tables for example atc1 which records information concerning principles of accounting one;



The figure above shows the design of all ATC(U) tables

The names of the course units are not stored in the database but they are displayed in the HTML code on the website. That nomenclature of the tables was used in order to allow easy manipulation of the tables in the PHP code.

Refer to appendix 1 for the design of other ATC(U) tables.

The CPA(U) tables

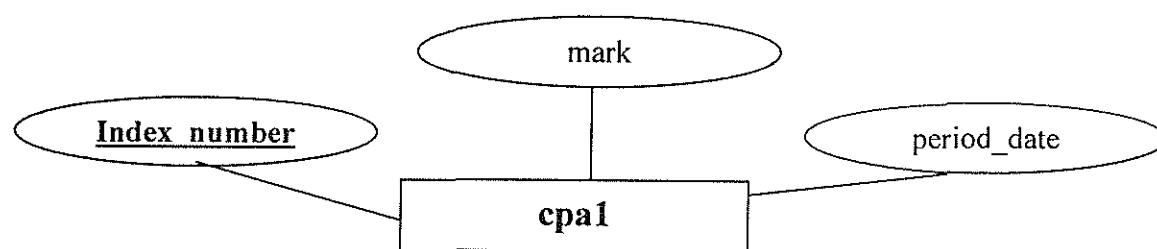
ATC as a course consists of 19 course units. Each course unit has a particular table which stores the students' index number who sat for that course unit, the mark obtained by the student and the date of sitting of the paper.

Each course unit has a table assigned to it as shown in the table below.

Table	Course unit
cpa1	Business Accounting
cpa2	Economic Environment
cpa3	Business Law
cpa4	Business Management
cpa5	Information Technology
cpa6	Financial Accounting
cpa7	Introduction to Management Accounting
cpa8	Principles of Taxation
cpa9	Financial Analysis
cpa10	Audit Theory
cpa11	Advanced Financial Accounting
cpa12	Mgt Decision and Control
cpa13	Taxation
cpa14	Finance
cpa15	Auditing
cpa16	Financial Reporting
cpa17	Business Policy
cpa18	Corporate Financial management
cpa19	Integration of Knowledge

The design of all these CPA(U) tables is synonymous.

Giving an example of cpa1 table, the design is as shown below.



<u>Index number</u> varchar(15)	mark int(3)	period_date varchar(20)
------------------------------------	----------------	----------------------------

The figure above shows the design of all CPA(U) tables

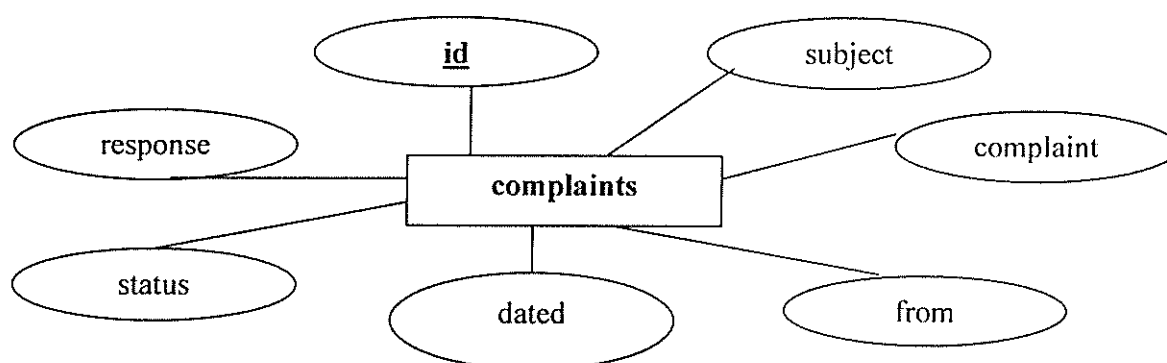
The names of the course units are not stored in the database but they are displayed in the HTML code on the website. That nomenclature of the tables was used in order to allow easy manipulation of the tables in the PHP code.

Refer to appendix 2 for the design of other CPA(U) tables.

The complaints table

This table consists of visitors complaints. It records all the complaints sent by the students. It also consists of the senders' username, the date on which the complaint was sent, the subject of the complaint and the id of the complaint.

The detailed design of the complaints table is as shown below.



<u>id</u>	response	status	complaint	from	dated	subject
int(6)	text	int(2)	text	varchar(30)	date	text

The figure above shows the design of complaints tables

2.0. Interface design

A log in form was designed. The log in form will be used to capture the username and password of the user.

The design of the login form is as shown below.

Member login	
User Name	<input type="text"/>
Password	<input type="password"/>
<input type="button" value="Log in"/>	

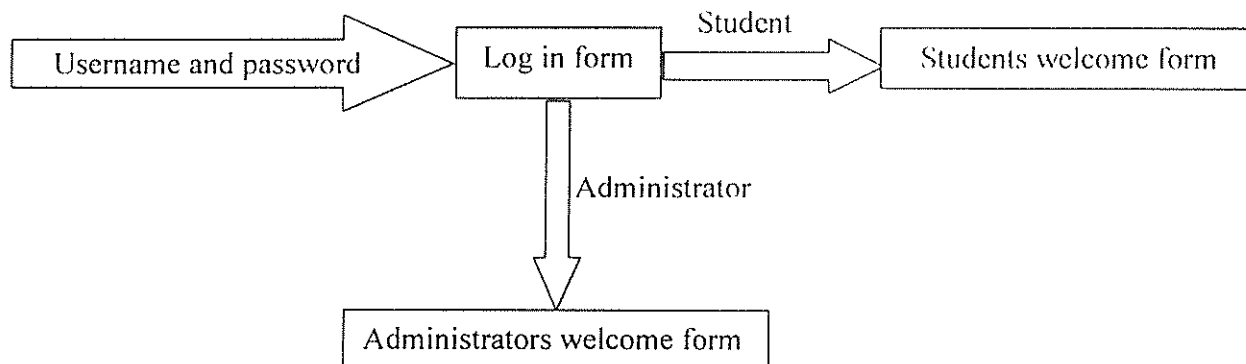
A webpage was designed to facilitate administrative privileges. It will only be accessed by the administrator.

The design of the administrators welcome page is as shown below.

<input type="button" value="Log out"/>	
Click on any choice to perform any administrative action	
Administrator ID	<input type="text"/>
Administrator Name	<input type="text"/>
Contacts	<input type="text"/>
<div><input type="button" value="Add user"/> <input type="button" value="Delete User"/> <input type="button" value="View Records"/> <input type="button" value="View Complaints"/> <input type="button" value="Edit Records"/></div>	

The administrators form will act as a base for accessing the other forms. These include the adduser, delete user, view records, view complaints, and edit records forms depending on the button clicked by the administrator. The design of the latter forms is illustrated in appendix 3.

The figure below shows the flow of accessing the welcome pages.



CHAPTER FIVE

IMPLEMENTATION

1.0 Creating the database

MySQL was used to design the database from which the website contents are extracted from.

PhpMyAdmin was used to assist in the creation of the database. The database was given a name- icpau_database.

PhpMyAdmin interface is as shown in the diagram below.

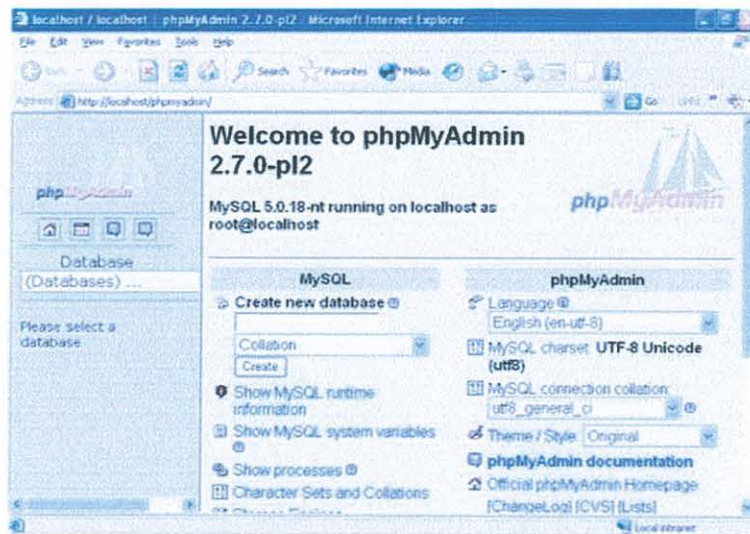


Figure showing how to create a database using PHPmyadmin

The database could also be created using the MySQL console using the code shown below;

```
CREATE DATABASE `icpau_database`;
```

2.0. Creation of tables

Table creation in PhpMyAdmin is as follows.

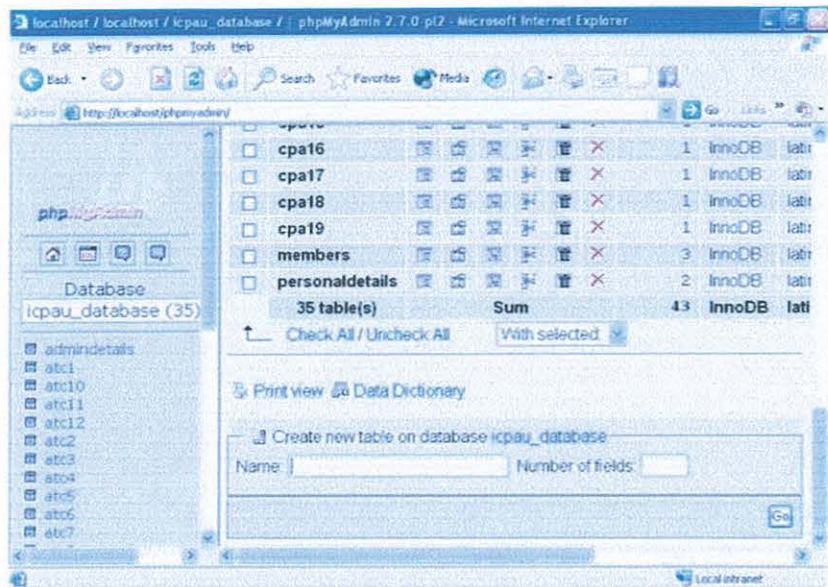
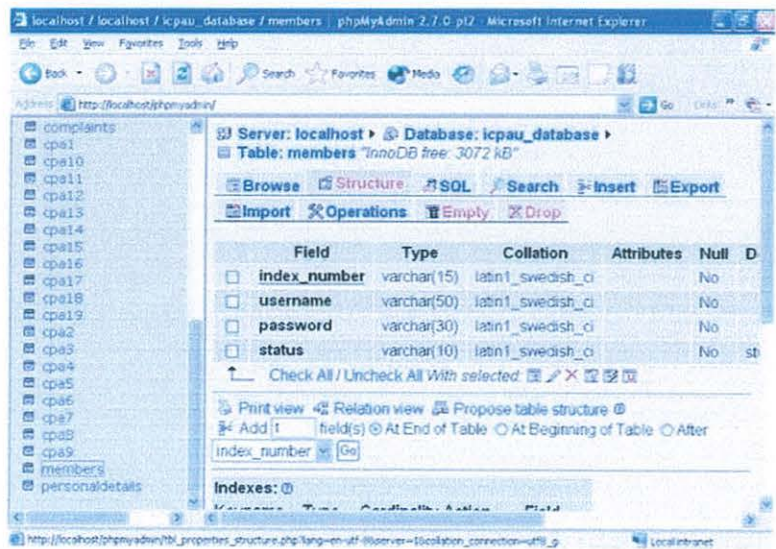


Figure showing how a table is created using PHPmyadmin

A table name is provided including the number of fields contained in the table.

An example of a table called members that was created is as shown below. The data type can be customized to match the intended data usage.



A table in PHPmyadmin

The MySQL console was some times used to implement the table creation. This was done by typing the following code at the console. For example creating the table “members” using the MySQL console, the following code can be used.

```
CREATE TABLE `members` (  
  `index_number` varchar(15) NOT NULL,  
  `username` varchar(20) NOT NULL,  
  `password` varchar(30) NOT NULL,  
  `status` varchar(20) NOT NULL default 'student',  
  PRIMARY KEY (`index_number`)  
);
```

The field properties such as the field name, data type, length, null and attributes were edited as shown in the diagram below.

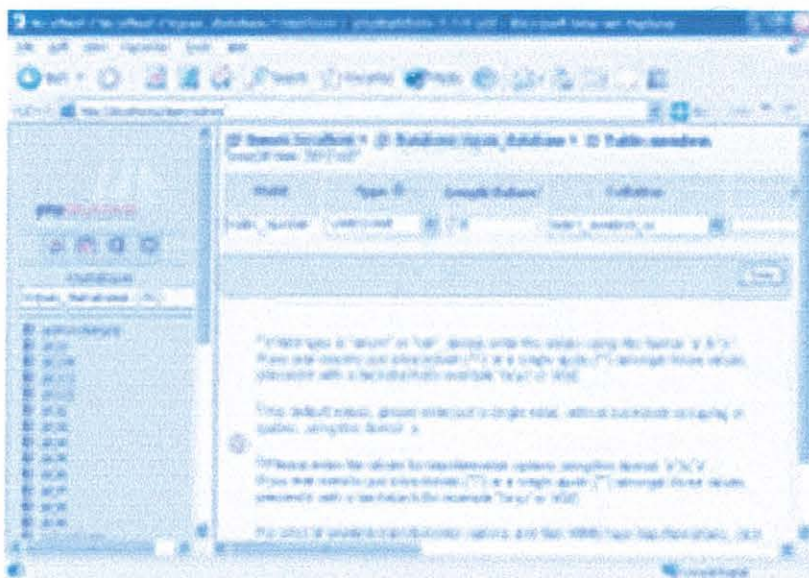


Figure showing how field properties are altered in PHPmyadmin

3.0. Database integrity

This can be defined as the correctness of the data entered or stored in the database. This can be in terms of the accepted data type for a field, the range of acceptable values and whether

or not a value is required. The correctness and accuracy of the data in the database is implemented both at database level and at interface level.

4.0. Integrity at table level

This was implemented by specifying the kind of data types for each field in all the tables of the data base. The width of the data to be entered was also specified at design level.

Those fields that are actually required for the system to function properly where specified as being NOT NULL. This therefore ensures that only correct values are entered in the tables which will avoid database corruption.

5.0. Interface level integrity.

Integrity at form creation

This was implemented in the forms. The kind of data required by the database was filtered by some validations on the forms. For example the length of the index number should not exceed 15 characters. This was implemented using the following code below.

```
.....  
<input type="text" name="index_number" maxlength="15">
```

Integrity on submitting the forms

Interface integrity was also implemented using javascript in the HTML code. The javascript code checks for fulfillment of certain conditions before forms are submitted to the web browser. For example the following code was used make sure empty forms are not submitted to the database.

```
.....  
<script language="JavaScript" type="text/JavaScript">  
function check() {  
    var i,j,k,m;  
    i=document.adding_user.user_name.value;  
    j=document.adding_user.index_number.value;  
    k=document.adding_user.password.value;
```

```

m=document.adding_user.name.value;
if ( i=="")
{
    alert("The user name of the person is required!");
    //history.back();
    adding_user.user_name.focus();
    adding_user.user_name.select();
    return false
}
else if (j=="")
{
    alert("The index number is required!");
    //history.back();
    adding_user.index_number.focus();
    adding_user.index_number.select();
    return false
}

if (k=="")
{
    alert("Please fill in the password text!!");
    //history.back();
    adding_user.password.focus();
    adding_user.password.select();
    return false
}

if (m=="")
{
    alert("Enter the name of the person");
    //history.back();
    adding_user.name.focus();

```

```

        adding_user.name.select();
        return false
    }

    else
    {
        adding_user.action ="register.php":
        adding_user.submit();
    }
}

```

It is very important to note that the server is totally ignorant of this browser-user interaction.

Integrity was also implemented in the PHP code.

For example PHP code was used to find out the existence of a username and a password. An appropriate message is returned on submission of an incorrect username and password. The following PHP code was used to test the validity of the visitor.

```

<?php
if($_POST[user_name]==" " || $_POST[pass]==" ") // blank text boxes
{
    header("Location: index.php?determinant=incomplete"); // return a variable
    incomplete to index.php where an appropriate message is echoed
}

else
{
    //connecting to db
    $host="localhost";
    $username="root";
    $password="";
    $dbname="icpau_database";

```

```
$dbconnect=mysql_connect($host,$username,$password) or die("Unable to connect to database");
```

```
mysql_select_db($dbname,$dbconnect);
```

```
if($result=mysql_query("select username, password from members where  
username='$_POST[user_name]' and password='$_POST[pass]'"))
```

```
{  
    $row=mysql_fetch_array($result);  
}
```

```
if($row[username]==$_POST[user_name] &&  
$row[password]==$_POST[pass])
```

```
{  
    $newuser=mysql_query("SELECT index_number from members where  
username='$_POST[user_name]'");
```

```
$newrow=mysql_fetch_array($newuser);  
setcookie("mapsession",$newrow[index_number],time()+36000);
```

```
$resultstatus=mysql_query("select status from members where  
username='$_POST[user_name]'");
```

```
$rowstatus=mysql_fetch_array($resultstatus);
```

```
if($rowstatus[status]=="student")
```

```
    header("Location: students/index.php");
```

```
    else
```

```
        header("Location: admin/index.php");
```

```
}
```

```
else
```

```
{
```

```
    header("Location: index.php?determinant=incorrect");
```

```
}
```

```
}
```

?>

The code below works hand in hand with the one above. The code below prints an appropriate message depending on the results returned by the code above.

```
<?php
    if($_GET[determinant]=="incomplete")
        echo "<font color=red>Please enter all fields</font>";
    if($_GET[determinant]=="incorrect")
        echo "<font color=red>Incorrect username or password<br> The username and password
are case sensitive</font>";
    if(isset($_POST[signin]))
        echo "<font color=red><center>You have successfully logged out</font>";
?>
```

More operations were implemented using the code in appendix 5.

6.0 Interface Implementation

The interface was designed in form of web pages. Buttons were included on the interface to ease the navigation on the website and control access to the system. PHP code was used to determine what to display depending on the user request.

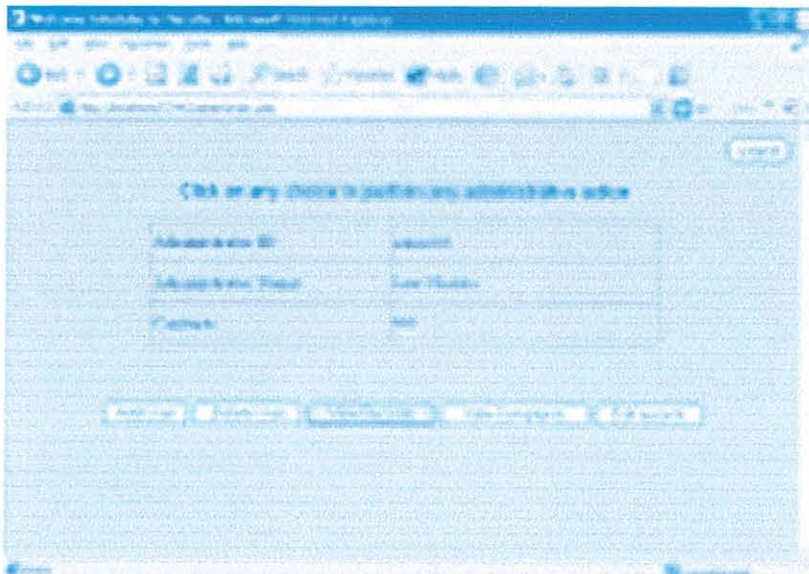


Figure showing the various navigation buttons on the webpage

For example the above diagram shows a webpage with a number of buttons. PHP code displays an appropriate page depending on the button clicked by the administrator.

The following explains further the above explanation.

```
<?php
```

```
if($_POST[deleteuser])  
{  
.....
```

```
} //end of delete user if
```

```
if($_POST[viewstudent])  
{  
.....
```

```
} //end of view student if
```

```
if($_POST[viewcomplaints])  
{  
.....
```

```
}
```

```
if($_POST[editinfo]) //edit the students records  
{  
.....
```

```
}
```

```
?>
```

Refer to appendix 4 for the detailed script which processes the above web page.

This was implemented using macromedia Dreamweaver. Dynamic pages were created into which the forms that connect to the database where included.

Creating a dynamic page using Dream Weaver is as shown in the diagram below.

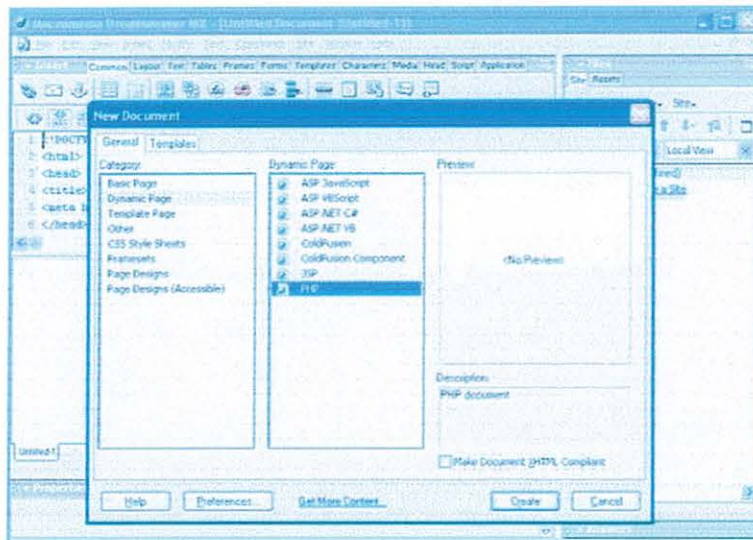


Figure shows how to create a PHP dynamic page

A login form was created and included into the index page.

An example of a form that was created in Dreamweaver is as shown below.

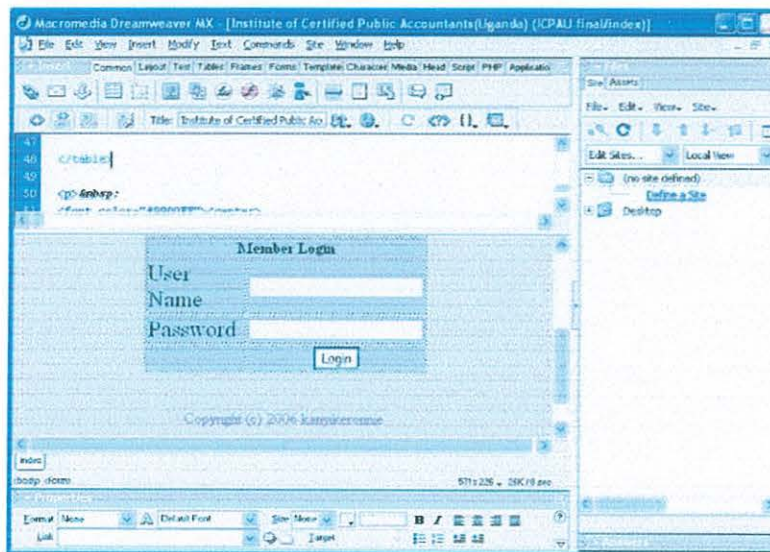


Figure showing a form creation process in Dreamweaver

CHAPTER SIX

EVALUATION, RECOMMENDATION AND CONCLUSION

1.0. Evaluation

The system was required to automate the examination results distribution process of the institute of Certified Public Accountants of Uganda.

Some problems were encountered during the process of the project. These include:

- Regular power interruptions which didn't enable the completion of particular tasks.
- Limited resources especially book materials for referencing.

However the obtained system meets approximately 80% of the purpose for which it was expected to fulfill.

The system has been tested and still contains a few identified loop holes. However it is OK for the system to be put on test in a working environment for a period of at least one month in order to identify the hidden loop holes so that they can as well be eliminated.

2.0. Recommendation

Provided a trusted server is used together with the necessary precautions taken, am positive that this system will drastically solve the problems specified in the problem statement. However this system could later be upgraded to meet other upcoming needs of the institute. This included registration online and payment of registration fees online.

Merits of the System

Easy to upgrade

Modifications to the system only need to be done on an independent computer and uploaded again.

Accessibility.

Since the system is on the World Wide Web, it has a wide coverage. A visitor just needs to log in to the system from where ever he or she is so long as the computer is connected to the internet.

3.0. Conclusion

Originally all the tasks of the institute specified in the problem statement where being done manually.

With the fast advancement in today's technology, we need to keep up to date with the modern technology. Better and efficient ways have to be improvised in order to simplify work. This in the long run will improve on performance since a bulk of tasks can be accomplished with in a short time.

REFERENCES

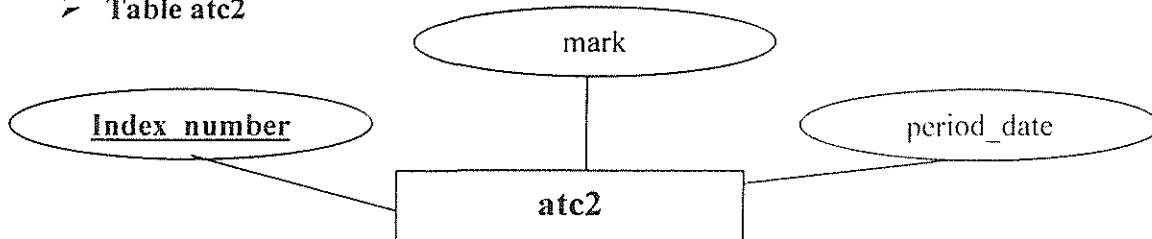
- [1].Management Information Systems, 4th 'Edition, James A. O'Brien, pages 9, 14 and 304.
- [2].Introduction to systems analysis and design, 4th edition, page 2,15, Igor Hawryszkiewicz
- [3].Access 2002 Bible by Cary N. Prague and Michael R. Irwin, page 1-20.
- [4].Wilkepedia online encyclopedia as per 30th June 2006.
- [5].SAMS Teach your self PHP4 in 24 Hours by Matt Zandstra page 1-350.
- [6].Web Application Development with PHP 4.0 by: Tobias Ratschiller and Till Gerken
- [7].Migrating from Microsoft Access to MySQL by Paul DuBois, NuSphere Corporation
- [8].Louis Rosenfeld and Peter Morville, Cambridge: Oreilly, 1998 “Information Architecture for the World Wide Web”.
- [9].Create dynamic sites with PHP & MySQL by: Md. Ashraful Anam

APPENDICES

Appendix 1

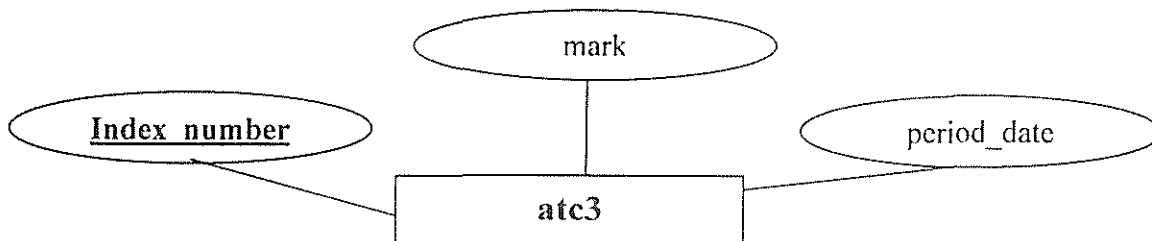
Below is a description of other ATC(U) tables.

➤ Table atc2



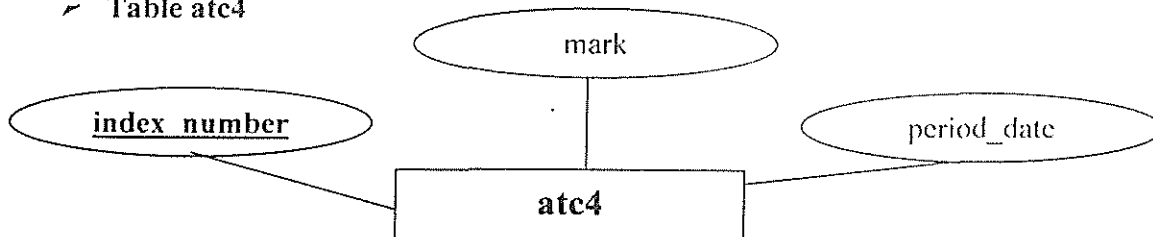
<u>Index number</u> varchar(15)	mark int(3)	period_date varchar(20)
------------------------------------	----------------	----------------------------

➤ Table atc3



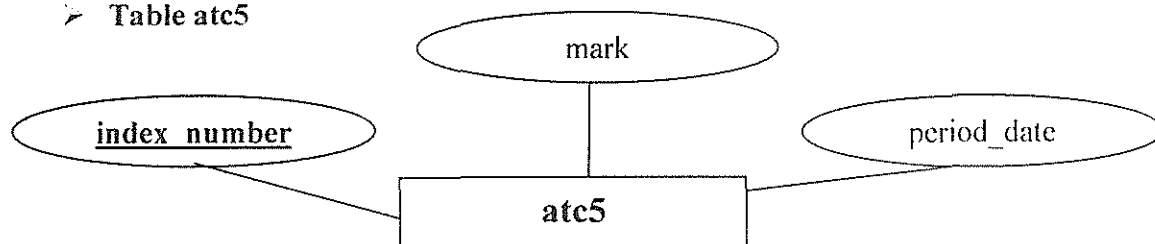
<u>Index number</u> varchar(15)	mark int(3)	period_date varchar(20)
------------------------------------	----------------	----------------------------

➤ Table atc4



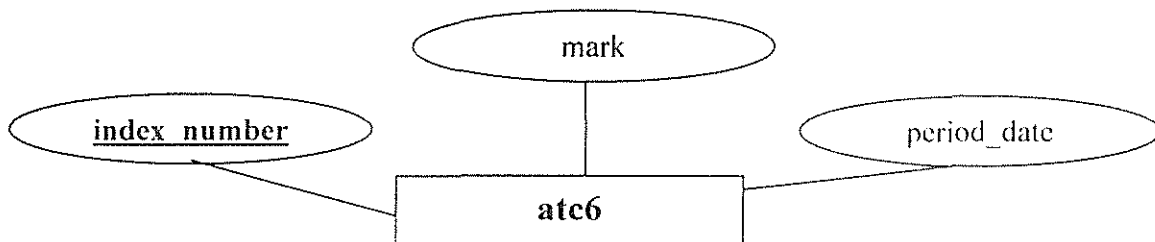
<u>index number</u> varchar(15)	mark int(3)	period_date varchar(20)
------------------------------------	----------------	----------------------------

➤ Table atc5



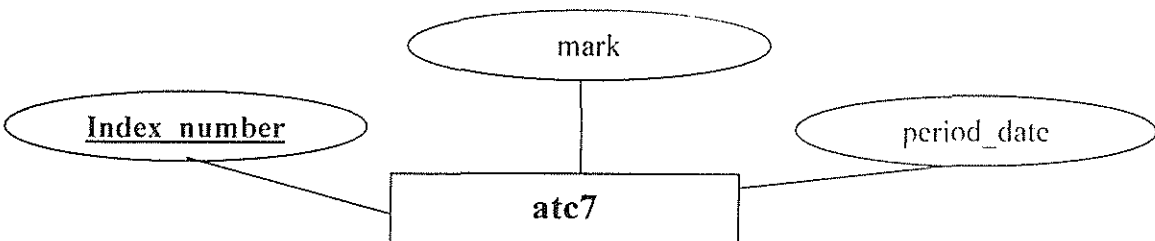
<u>index_number</u> varchar(15)	mark int(3)	period_date varchar(20)
------------------------------------	----------------	----------------------------

➤ Table atc6



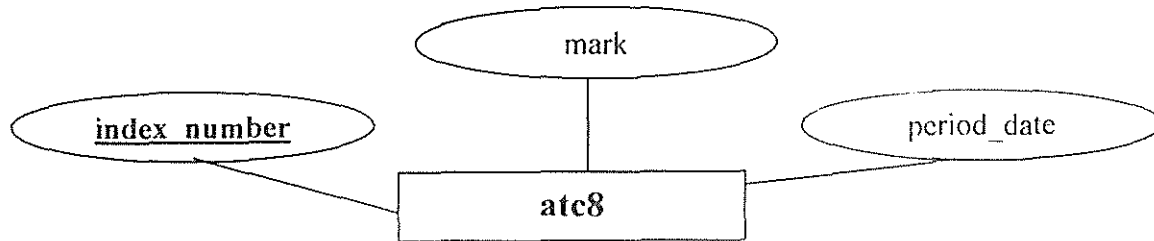
<u>index_number</u> varchar(15)	mark int(3)	period_date varchar(20)
------------------------------------	----------------	----------------------------

➤ Table atc7



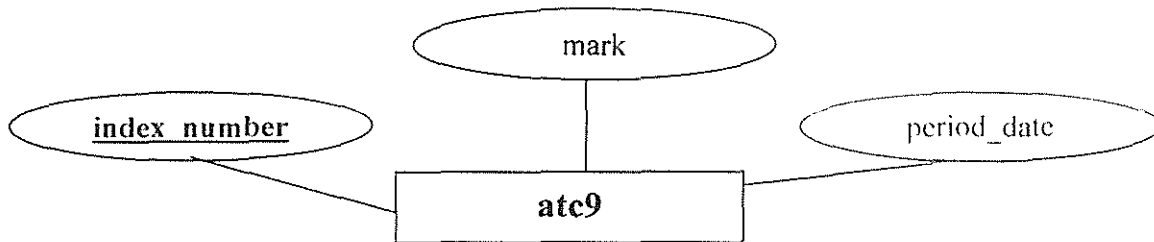
<u>index_number</u> varchar(15)	mark int(3)	period_date varchar(20)
------------------------------------	----------------	----------------------------

➤ Table atc8



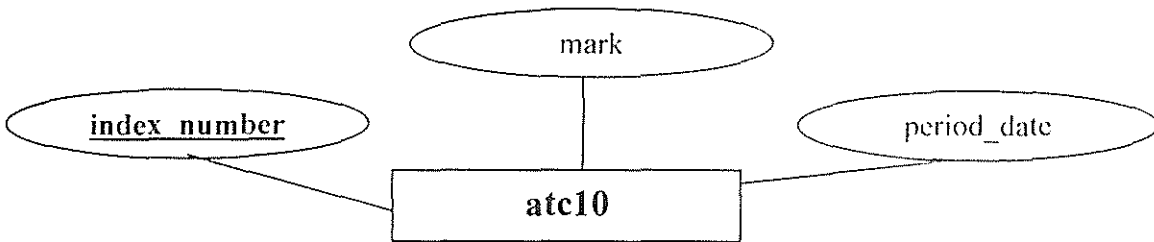
<u>Index number</u> varchar(15)	mark int(3)	period_date varchar(20)
------------------------------------	----------------	----------------------------

➤ Table atc9



<u>index number</u> varchar(15)	mark int(3)	period_date varchar(20)
------------------------------------	----------------	----------------------------

➤ Table atc10

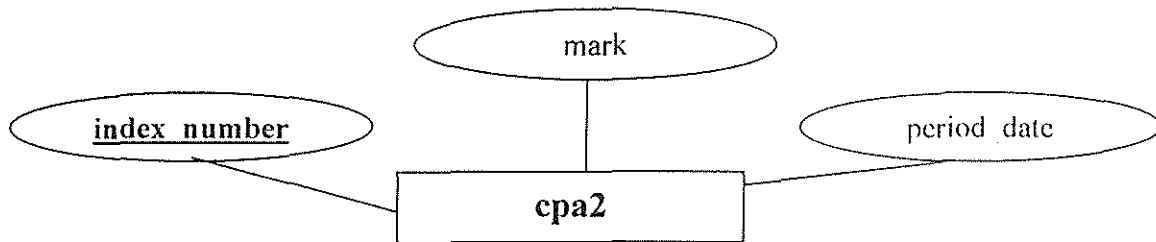


<u>index number</u> varchar(15)	mark int(3)	period_date varchar(20)
------------------------------------	----------------	----------------------------

Appendix 2

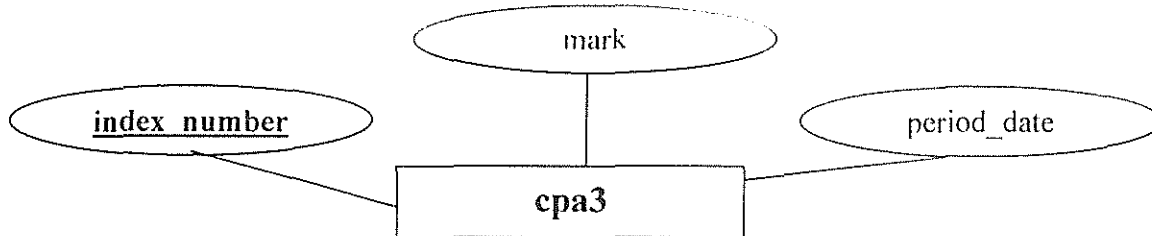
The design of other CPA(U) tables

➤ Table cpa2



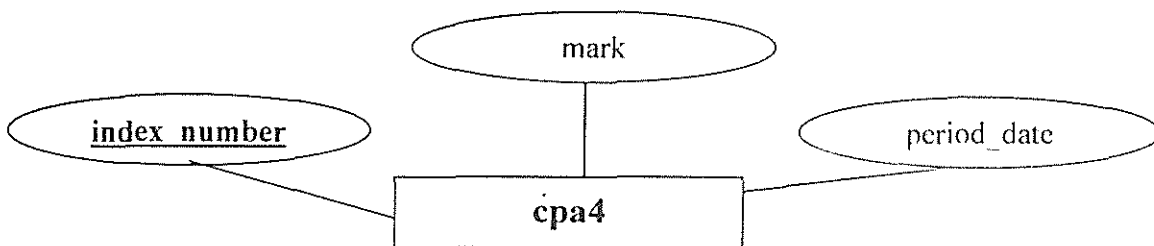
<u>index_number</u> varchar(15)	mark int(3)	period_date varchar(20)
------------------------------------	----------------	----------------------------

➤ Table cpa3



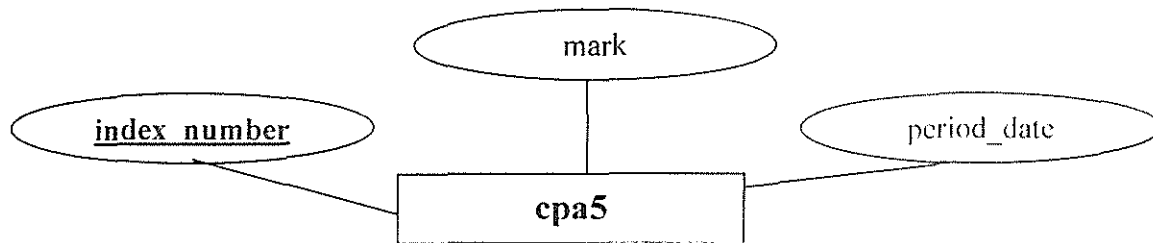
<u>index_number</u> varchar(15)	mark int(3)	period_date varchar(20)
------------------------------------	----------------	----------------------------

➤ Table cpa4



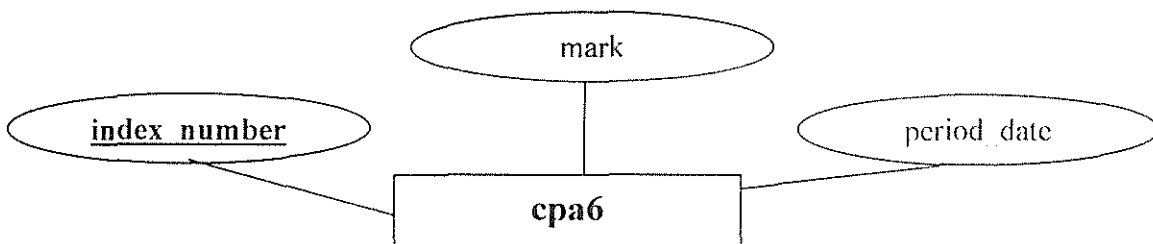
<u>index_number</u> varchar(15)	mark int(3)	period_date varchar(20)
------------------------------------	----------------	----------------------------

➤ Table cpa5



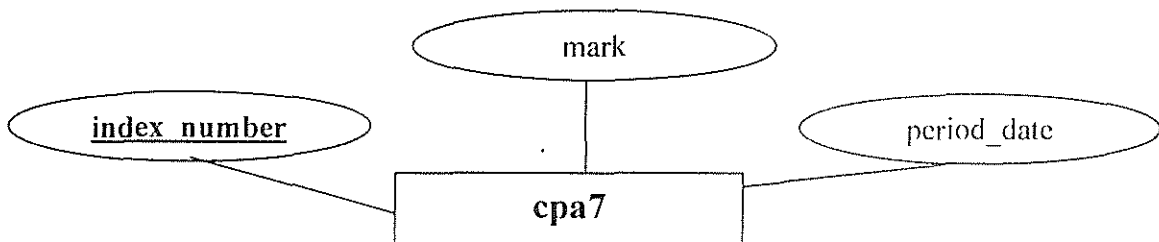
<u>index_number</u> varchar(15)	mark int(3)	period_date varchar(20)
------------------------------------	----------------	----------------------------

➤ Table cpa6



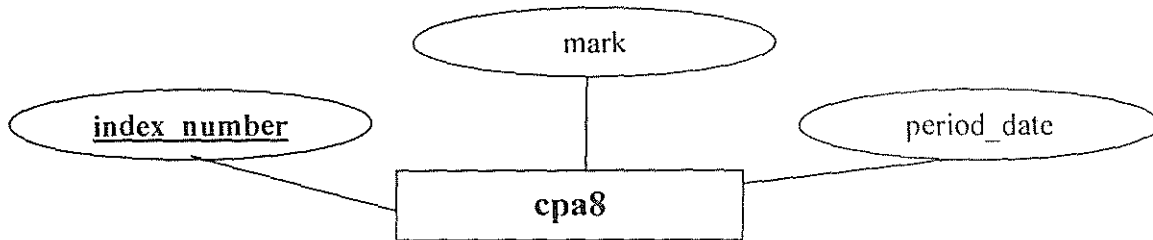
<u>index_number</u> varchar(15)	mark int(3)	period_date varchar(20)
------------------------------------	----------------	----------------------------

➤ Table cpa7



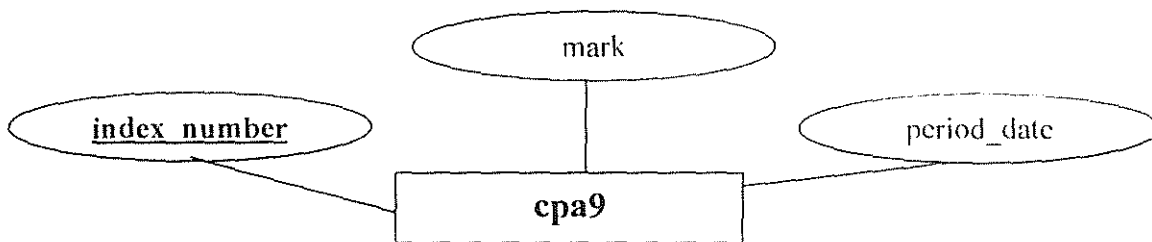
<u>index_number</u> varchar(15)	mark int(3)	period_date varchar(20)
------------------------------------	----------------	----------------------------

➤ Table cpa8



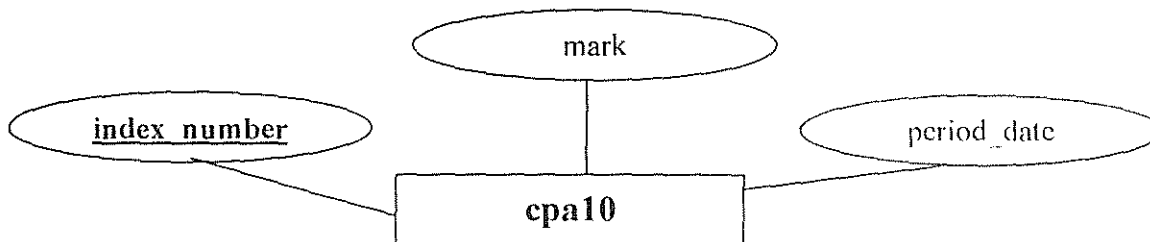
<u>index_number</u> varchar(15)	mark int(3)	period_date varchar(20)
------------------------------------	----------------	----------------------------

➤ Table cpa9



<u>index_number</u> varchar(15)	mark int(3)	period_date varchar(20)
------------------------------------	----------------	----------------------------

➤ Table cpa10



<u>index_number</u> varchar(15)	mark int(3)	period_date varchar(20)
------------------------------------	----------------	----------------------------

Appendix 3

➤ Form used to add a new user:

User name:	<input type="text"/>
Index number:	<input type="text"/>
Password:	<input type="text"/>
Name:	<input type="text"/>
Address:	<input type="text"/>
Tel.no:	<input type="text"/>
Institute:	<input type="text"/>
Sponsor:	<input type="text"/>
Email:	<input type="text"/>
Course:	<input type="text"/>
<div><input type="button" value="Create user"/> <input type="button" value="Reset"/></div>	

➤ Form used to delete a user:

Select the index number of the person to be deleted

Index number	<input type="text"/>
<div><input type="button" value="Delete"/></div>	

➤ Form which links to edit records:

Select the index number of the person to be deleted

Index number	<input type="text"/>
<input type="button" value="Edit"/>	

➤ Form used to edit personal details:

Index number:	<input type="text"/>
Name:	<input type="text"/>
Address:	<input type="text"/>
Telephone number:	<input type="text"/>
Email address	<input type="text"/>
Course:	<input type="text"/>
Level:	<input type="text"/>
Institute:	<input type="text"/>
Sponsor:	<input type="text"/>
<input type="button" value="Save"/> <input type="button" value="Reset"/>	

➤ Page to display complaints from students

No	From	Subject	Message	Date reported

The rows of the table will automatically be generated as more complaints are sent in from the students.

Showing records 1 to 10 of NEXT

➤ Page to display ATC(U) student's results will be as shown below

Paper No	Paper	Grade	Sitting Period
1	Principles of Accounting I		
2	Principles of Law I		
3	Business Math & Statistics		
4	Commercial Environment		
5	Principles of Accounting II		
6	Principles of Law II		
7	Economics		
8	Information Systems		
9	Introduction to Mgt Accounting		
10	Principles of Taxation		
11	Business Management		
12	Introduction to Fin.Reporting		

The grade column will display either a pass or a fail depending on the mark attained by the student. 50% and above in any subject is a pass. Below 50% is a fail. A subject will automatically be assigned pending if the student has not yet sat for the paper.

➤ Page to display CPA(U) student's results will be as shown below

Paper No	Paper	Grade	Sitting period
1	Business Accounting		
2	Economic Environment		
3	Business Law		
4	Business Mgt		
5	Information Technology		
6	Financial Accounting		
7	Introduction to Mgt Accounting		
8	Principles of Taxation		
9	Financial Analysis		
10	Audit Theory		
11	Advanced Financial Accounting		
12	Mgt Decision & Control		
13	Taxation		
14	Finance		
15	Auditing		
16	Financial Reporting		
17	Business Policy		
18	Corporate Financial Mgt		
19	Integration of Knowledge		

The grade column will display either a pass or a fail depending on the mark attained by the student. 50% and above in any subject is a pass. Below 50% is a fail. A subject will automatically be assigned pending if the student has not yet sat for the paper.

Appendix 4

Script which processes the administrators welcome page.

```
<?php
if($_COOKIE[mapsession])    //checking the existence of a cookie
{
    //connecting to db
    $host="localhost";
    $username="root";
    $password="accounts";
    $dbname="icpau_database";
    $dbconnect=mysql_connect($host,$username,$password) or die("Unable to connect to
database");
    mysql_select_db($dbname,$dbconnect);

//ADDING NEW USER
if($_POST[adduser]) // this adds new user
{
    ?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
    <title>Adding a new user</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<script language="JavaScript" type="text/JavaScript">
// function which ensures that blank textboxes are not submitted
function check() {
    var i,j,k,m;
    i=document.adding_user.user_name.value;
    j=document.adding_user.index_number.value;
    k=document.adding_user.password.value;
    m=document.adding_user.name.value;
```

```

if ( i=="")
{
    alert("The user name of the person is required!");
    //history.back();
    adding_user.user_name.focus();
    adding_user.user_name.select();
    return false
}
else if (j=="")
{
    alert("The index number is required!");
    //history.back();
    adding_user.index_number.focus();
    adding_user.index_number.select();
    return false
}

if (k=="")
{
    alert("Please fill in the password text!!");
    //history.back();
    adding_user.password.focus();
    adding_user.password.select();
    return false
}

if (m=="")
{
    alert("Enter the name of the person");
    //history.back();
    adding_user.name.focus();
    adding_user.name.select();

```

```

        return false
    }

    else
    {
        adding_user.action ="register.php";
        adding_user.submit();
    }
}

</SCRIPT>
</head>
<body bgcolor="#CCCCCC">
<form name="adding_user" method="POST">
    <div align="center">
        <p><font size="+1" face="Georgia, Times New Roman, Times, serif">Fill in the
textboxes below , to add a new user to the
        database</font></p>
    </div>
    <table width="54%" border="1" align="center">
        <tr>
            <td width="49%" bordercolor="#FFFFFF" bgcolor="#CCCCCC">
                <p><strong>Username: </strong></p></td>
            <td width="51%"><input type="text" name="user_name"></td>
        </tr>
        <tr>
            <td bgcolor="#CCCCCC"><strong>Index Number: </strong></td>
            <td><input type="text" name="index_number" maxlength="15"></td>
        </tr>
        <tr>
            <td bgcolor="#CCCCCC"><strong>Password:</strong></td>
            <td><input type="text" name="password"></td>
        </tr>
    </table>

```

```

<tr>
  <td bgcolor="#CCCCCC"><strong>Name:</strong></td>
  <td><input type="text" name="name"></td>
</tr>
<tr>
  <td bgcolor="#CCCCCC"><strong>Address:</strong></td>
  <td><input type="text" name="address"></td>
</tr>
<tr>
  <td bgcolor="#CCCCCC"><strong>Tel. No:</strong></td>
  <td><input type="text" name="tel_number"></td>
</tr>
<tr>
  <td bgcolor="#CCCCCC"><strong>Institute:</strong></td>
  <td><input type="text" name="institute"></td>
</tr>
<tr>
  <td bgcolor="#CCCCCC"><strong>Sponsor:</strong></td>
  <td><select name="sponsor">
    <option value="self">Self</option>
    <option value="Government">Government</option>
  </select></td>
</tr>
<tr>
  <td bgcolor="#CCCCCC"><strong>Email:</strong></td>
  <td><input type="text" name="e_mail"></td>
</tr>
<tr>
  <td bgcolor="#CCCCCC"><strong>Course:</strong></td>
  <td><select name="course">
    <option value="ATC(U)">ATC(U)</option>

```



```

<td>Index Number</td>
<td><select name="index_number">
    <?php
        $result=mysql_query("select `index_number` from `members` where
`status`='student'");

        while($row=mysql_fetch_array($result))
        {
            ?>
                <option value="<?php echo $row[index_number]; ?>"><?php echo
$row[index_number]; ?></option>
            <?php
        } ?>
    </select>
</td>
</tr>
<tr>
<?php
    //the name of the delete button is todelete
    ?>
    <td colspan="2"><input type="submit" name="todelete" value="delete">
    </td>
</tr>
</table>
</form>
</body>
</html>

<?php
}
//end of delete user if

```

```

if($_POST[viewstudent]) // viewing records of a student
{
?>

<html><head><title>View student</title></head>

<body bgcolor="#CCCCCC">

    <table border="0" align="center" width="75%"><tr><td>

        <font face="Arial, Helvetica, sans-serif" color="#0033CC" size="+1"><center>Select the
index number of the person whose records are to be viewed

        </center></font></td></tr></table><br>

        <form method="POST" action="view_student.php">

            <table width="75%" border="1" align="center">

                <tr>

                    <td>Index Number</td>

                    <td><select name="index_number">

                        <?php

                            $result=mysql_query("select `index_number` from `members` where
`status`='student'");

                            while($row=mysql_fetch_array($result))
                            {
                                ?>

                                    <option value="<?php echo $row[index_number]; ?>"><?php echo
$row[index_number]; ?></option>

                                    <?php

                                        } ?>

                                </select>

                                </td>

                            </tr>

                            <tr>

                                <td colspan="2"><input name="to_view" type="submit" value="View Records">

```

```

        </td>
    </tr>
</table>
</form>
</body>
</html>
<?php
} //end of view student if
if($_POST[viewcomplaints]) //viewing complaints from a student
{
    header("Location: complaints.php");
}
if($_POST[editinfo]) //forewords you to a page for editing students records.
{
    ?>
    <html><head><title>Edit User</title></head>
    <body bgcolor="#CCCCCC">
    <form method="POST" action="edit_user.php">
    <table border="0" align="center" width="75%"><tr><td>
    <font face="Arial, Helvetica, sans-serif" color="#0033CC" size="+1"><center>Select the
index number of the person whose records are to be edited
    </center></font></td></tr></table><br>
    <table width="75%" border="1" align="center">
    <tr>
    <td>Index Number</td>
    <td><select name="index_number">
    <?php
        $result=mysql_query("select `index_number` from `members` where
`status`='student'");

        while($row=mysql_fetch_array($result))

```

```

        {
            ?>
                <option value="<?php echo $row[index_number]; ?>"><?php echo
$row[index_number]; ?></option>
            <?php
        } ?>
    </select>
    </td>
</tr>
<tr>
    <td colspan="2"><input name="toedit" type="submit" value="Edit Records">
    </td>
</tr>
</table>
</form>
</body>
</html>
<?php
} //end of edit info
}
?>

```

Appendix 5

Script which adds an entry to the database.

```
<?php
//connecting to db
$host="localhost";
$username="root";
$password="accounts";
$dbname="icpau_database";
$dbconnect=mysql_connect($host,$username,$password) or die("Unable to connect to
database");
mysql_select_db($dbname,$dbconnect);

if(mysql_query("INSERT INTO `members` ( `index_number` , `username` , `password` ,
`status` )
VALUES ('$_POST[index_number]', '$_POST[user_name]', '$_POST[password]',
'student')"))
{
?>
<html> <head>
<title> Successfully registered</title>
</head>
<body bgcolor="#CCCCCC">

<p align="center"><font color="#CC0000" size="+1" face="Arial, Helvetica, sans-
serif">The following Information has been entered
into the database</font></p>
<table width="60%" border="0" align="center">
<tr>
<td width="37%" bordercolor="#000000" bgcolor="#9999CC"> User name </td>
<td width="63%" bordercolor="#99FF00" bgcolor="#CC99CC"><?php echo
$_POST[user_name];?></td>
</tr>
<tr>
<td bordercolor="#000000" bgcolor="#9999CC">Index number</td>
<td bordercolor="#99FF00" bgcolor="#CC99CC"><?php echo
$_POST[index_number];?></td>
</tr>
<tr>
<td bordercolor="#000000" bgcolor="#9999CC">Pass word</td>
<td bordercolor="#99FF00" bgcolor="#CC99CC"><?php echo
$_POST[password];?></td>
</tr>
<tr>
<td bordercolor="#000000" bgcolor="#9999CC">Name</td>
```

```

    <td bordercolor="#99FF00" bgcolor="#CC99CC"><?php echo $_POST[name];?></td>
</tr>
<tr>
    <td bordercolor="#000000" bgcolor="#9999CC">Address</td>
    <td bordercolor="#99FF00" bgcolor="#CC99CC"><?php echo $_POST[address];?></td>
</tr>
<tr>
    <td bordercolor="#000000" bgcolor="#9999CC">Telephone number</td>
    <td bordercolor="#99FF00" bgcolor="#CC99CC"><?php echo
$_POST[tel_number];?></td>
</tr>
<tr>
    <td bordercolor="#999999" bgcolor="#9999CC">Institute</td>
    <td bordercolor="#99FF00" bgcolor="#CC99CC"><?php echo $_POST[institute];?></td>
</tr>
<tr>
    <td bordercolor="#000000" bgcolor="#9999CC">Sponsor</td>
    <td bordercolor="#99FF00" bgcolor="#CC99CC"><?php echo $_POST[sponsor];?></td>
</tr>
<tr>
    <td bordercolor="#000000" bgcolor="#9999CC">Email</td>
    <td bordercolor="#99FF00" bgcolor="#CC99CC"><?php echo $_POST[e_mail];?></td>
</tr>
<tr>
    <td bordercolor="#000000" bgcolor="#9999CC">Course</td>
    <td bordercolor="#99FF00" bgcolor="#CC99CC"><?php echo $_POST[course];?></td>
</tr>
</table>
<?php
}

```

```

if(mysql_query("INSERT INTO `personaldetails` ( `index_number` , `name` , `address` ,
`tel_number`,`name_institute`,`sponsor`,
`email`,`course`)
VALUES ('$_POST[index_number]', '$_POST[name]', '$_POST[address]',
'$_POST[tel_number]', '$_POST[institute]', '$_POST[sponsor]',
'$_POST[e_mail]', '$_POST[course]')"))
{
    if($_POST[course]="ATC(U)")
    {
        // echo "<br> the course is ATC";
        // updating all the ATC course tables
        for($i=1;$i<=12;$i++)
        {
            $tablename="atc";
            $value=$i;

```

```

        settype($value, "string");
        $tablename.= $value;
        mysql_query("INSERT INTO ` $tablename ` ( `index_number` , `mark`)
VALUES (`$_POST[index_number]`, '-1')");
    }
}

if($_POST[course]="CPA(U)")
{
    // echo "<br> the course is ATC";
    // updating all the ATC course tables
    for($i=1;$i<=19;$i++)
    {
        $tablename="cpa";
        $value=$i;
        settype($value, "string");
        $tablename.= $value;
        mysql_query("INSERT INTO ` $tablename ` ( `index_number` , `mark`) ;
VALUES (`$_POST[index_number]`, '-1')");
    }
}
?>

```

Script used to update personal details in the database.

```

<?php
//connecting to db
$host="localhost";
$username="root";
$password="accounts";
$dbname="icpau_database";
$dbconnect=mysql_connect($host,$username,$password) or die("Unable to connect to
database");
mysql_select_db($dbname,$dbconnect);
//updating the database
mysql_query("UPDATE `personaldetails` SET `name` = `$_POST[naming]`,
`address` = `$_POST[address]`, `tel_number` = `$_POST[tel_number]`, `email` =
`$_POST[e_mail]`, `course` = `$_POST[course]`,
`institute` = `$_POST[institute]`, `sponsor` = `$_POST[sponsor]`
WHERE `index_number`=`$_POST[index_number]`");
?>

```


Script used to view records in a form.

```
<?php
if($_COOKIE[mapsession])
{

//connecting to db
$host="localhost";
$username="root";
$password="accounts";
$dbname="icpau_database";
$dbconnect=mysql_connect($host,$username,$password) or die("Unable to connect to
database");
mysql_select_db($dbname,$dbconnect);
$result=mysql_query("select username from members where
index_number=$_POST[index_number]");
$row=mysql_fetch_array($result);
?>
<html>
<head>
<meta http-equiv="pragma" content="no-cache" />
<title>You are viewing <?php echo $row[username]; ?>'s page</title>
</head>
<body bgcolor="#CCCCCC">
<?php //reading from the personalinfor table

$personalinfor = mysql_query("select
index_number,name,address,tel_number,email,course,level,name_institute,sponsor from
personaldetails where index_number='$_POST[index_number]'");
$row=mysql_fetch_array($personalinfor);
?>
<form name="signout" method="post" action="../index.php">
<div align="right">
<input type="submit" name="signout" value="Logout">
</div>
</form>
<table width="62%" border="0" align="center" bordercolor="#00FFFF"
bgcolor="#CCCC99">
<tr bgcolor="#CC9999">
<td width="49%" bordercolor="#FFCCCC">Index number</td>
<td width="51%"><?php echo $row[index_number]; ?> </td>
<?php //display the index number of the logged in person ?>
</tr>
<tr bgcolor="#CC9999">
<td bordercolor="#FFCCCC">Name</td>
<td><?php echo $row[name]; ?></td>
```

```

</tr>
<tr bgcolor="#CC9999">
  <td bordercolor="#FFCCCC">Address</td>
  <td><?php echo $row[address]; ?></td>
</tr>
<tr bgcolor="#CC9999">
  <td bordercolor="#FFCCCC">Telephone number</td>
  <td><?php echo $row[tel_number]; ?></td>
</tr>
<tr bgcolor="#CC9999">
  <td bordercolor="#FFCCCC">Email Address</td>
  <td><?php echo $row[email]; ?></td>
</tr>
<tr bgcolor="#CC9999">
  <td bordercolor="#FFCCCC">Course</td>
  <td><?php echo $row[course]; ?></td>
</tr>
<tr bgcolor="#CC9999">
  <td bordercolor="#FFCCCC">Level</td>
  <td><?php echo $row[level]; ?></td>
</tr>
<tr bgcolor="#CC9999">
  <td bordercolor="#FFCCCC">Name of training institute</td>
  <td><?php echo $row[name_institute]; ?></td>
</tr>
<tr bgcolor="#CC9999">
  <td bordercolor="#FFCCCC">Sponsor</td>
  <td><?php echo $row[sponsor]; ?></td>
</tr>
</table>
<form method="post" action="results.php">
  <div align="center">
    <input name="index_number" type="hidden" value="<?php echo
$_POST[index_number]; ?>">
    <input type="submit" name="Submit" value="check out your results" align="absmiddle">
  </div>
</form>
</body>
</html>
<?php
}
?>

```

Script used to delete from the database

```
<?php
//connecting to db
$host="localhost";
$username="root";
$password="accounts";
$dbname="icpau_database";
$dbconnect=mysql_connect($host,$username,$password) or die("Unable to connect to
database");
mysql_select_db($dbname,$dbconnect);
$result=mysql_query("select `course` from `personaldetails` where
index_number=$_POST[index_number]");
$row=mysql_fetch_array($result);
//echo $_POST[todelete];
if(mysql_query("DELETE FROM `members` WHERE `index_number`
=$_POST[index_number]"))
{
if ($row[course]="ATC(U)")
{
for($i=1;$i<=12;$i++)
{
$tablename="atc";
$value=$i;
settype($value, "string");
$tablename.=$value;
mysql_query("DELETE FROM `$tablename` WHERE
`index_number`=$_POST[index_number]");
}
}

if ($row[course]="CPA(U)")
{
for($i=1;$i<=19;$i++)
{
$tablename="cpa";
$value=$i;
settype($value, "string");
$tablename.=$value;
mysql_query("DELETE FROM `$tablename` WHERE
`index_number`=$_POST[index_number]");
}
}
echo "<br>". "Succesfully managed to delete from the database";
}
```

```

if(mysql_query("DELETE FROM `personaldetails` WHERE `index_number`
=$_POST[index_number]"))
{
    echo "successfully managed to delete from the database";
}
?>

```

Script for updating students marks

```

<?php
//connecting to db
$host="localhost";
$username="root";
$password="accounts";
$dbname="icpau_database";
$dbconnect=mysql_connect($host,$username,$password) or die("Unable to connect to
database");
mysql_select_db($dbname,$dbconnect);
echo $_POST[textfield1]."this is one". "<br>";
echo $_POST[textfield2]." <br>";
    echo $_POST[textfield3]."this is 3";
// updating all the ATC course tables
for($i=1;$i<=12;$i++)
{
    $tablename="atc";
    $textbox="textfield";
    $value=$i;
    settype($value, "string");
    $tablename.=$value;
    $textbox.=$value;
    mysql_query("UPDATE ` $tablename` SET `mark`='$_POST[$textbox]'
WHERE index_number=$_POST[TRACKING]");
    //echo "I HAVE SUCCEFULLY UPDATED THE DATABASE";
}
?>

```